

무선 센서 네트워크에서 효율적인 집계 질의 처리

Efficient Processing of Aggregate Queries in Wireless Sensor Networks

김 정 준* 신 인 수** 이 기 영*** 한 기 준****
 Joung Joon Kim In Su Shin Ki Young Lee Ki Joon Han

요 약 최근 무선 센서 네트워크에서 센서로부터 원하는 데이터를 가져오는 네트워크 내 집계 질의 처리 기법에 대한 연구가 활발히 진행되고 있다. 기존의 대표적인 네트워크 내 집계 질의 처리 기법들은 집계 질의 처리를 위해 라우팅 알고리즘과 데이터 구조를 제안하고 있다. 그러나 이러한 기법들은 센서 노드들의 에너지 소모가 크고, 질의 처리 결과 정확도가 떨어지고, 또한 질의 처리 시간이 오래 걸리는 문제점들을 가지고 있다. 본 논문에서는 이러한 기존 집계 질의 처리 기법들의 문제점을 해결하고 무선 센서 네트워크에서 보다 효율적인 집계 질의 처리를 위해 BPA(Bucket-based Parallel Aggregation)를 제시하였다. BPA는 질의 영역을 센서 노드 분포에 따라 쿼드 트리로 구성하여 집계 질의를 병렬로 처리하고, 각 센서 노드로 하여금 데이터를 이중 전송하게 함으로써 전송 오류로 인한 데이터 손실을 줄인다. 또한, BPA는 집계 질의 처리시 버킷 기반의 데이터 구조를 이용하고 이러한 버킷 데이터 구조를 버킷내 데이터 개수에 따라 적응적으로 분할 및 합병한다. 특히 버킷내 데이터 크기를 줄이기 위해 데이터를 압축하고 데이터 전송 횟수를 줄이기 위해 필터링을 수행한다. 마지막으로 센서 데이터를 이용한 다양한 실험을 통해 본 논문에서 제안하는 BPA의 우수성을 입증하였다.

키워드 : 무선 센서 네트워크, 집계 질의, 라우팅, 데이터 구조

Abstract Recently as efficient processing of aggregate queries for fetching desired data from sensors has been recognized as a crucial part, in-network aggregate query processing techniques are studied intensively in wireless sensor networks. Existing representative in-network aggregate query processing techniques propose routing algorithms and data structures for processing aggregate queries. However, these aggregate query processing techniques have problems such as high energy consumption in sensor nodes, low accuracy of query processing results, and long query processing time. In order to solve these problems and to enhance the efficiency of aggregate query processing in wireless sensor networks, this paper proposes Bucket-based Parallel Aggregation(BPA). BPA divides a query region into several cells according to the distribution of sensor nodes and builds a Quad-tree, and then processes aggregate queries in parallel for each cell region according to routing. And it sends data in duplicate by removing redundant data, which, in turn, enhances the accuracy of query processing results. Also, BPA uses a bucket-based data structure in aggregate query processing, and divides and conquers the bucket data structure adaptively according to the number of data in the bucket. In addition, BPA compresses data in order to reduce the size of data in the bucket and performs data transmission filtering when each sensor node sends data. Finally, in this paper, we prove its superiority through various experiments using sensor data.

Keywords : Wireless Sensor Networks, Aggregate Query, Routing, Data Structure

† 본 연구는 국토해양부 첨단도시기술개발사업 - 지능형국토정보기술혁신 사업과제의 연구비지원(10국토정보J71)에 의해 수행되었습니다.

* 건국대학교 컴퓨터공학과 강의교수, jjkim9@db.konkuk.ac.kr

** 건국대학교 컴퓨터공학과 박사과정, isshin@db.konkuk.ac.kr

*** 을지대학교 의료IT마케팅학과 교수, kylee@eulji.ac.kr(교신저자)

**** 건국대학교 컴퓨터공학과 교수, kjhan@db.konkuk.ac.kr

1. 서론

최근 온도 센서, 습도 센서, 압력 센서 등과 같이 다양한 데이터를 센싱하는 센서 기술과 무선 통신 기술 등이 급속히 발전함에 따라 군사, 의료, 기상, 환경, 교통, 가정, 회사 등과 같은 다양한 응용 분야에서 무선 센서 네트워크 기술을 활용하기 위한 연구가 활발히 진행되고 있다[2, 10].

특히, 센서로부터 센싱된 데이터에 대해 집계 결과를 구하는 집계 질의(Aggregate Query) 처리가 중요한 연구 분야로 인식되고 있다. 집계 질의는 무선 센서 네트워크에서 전체 또는 특정 영역에 대해 MAX, MIN, SUM, AVG, COUNT, MEDIAN, HISTOGRAM 등을 수행하는 질의이다.

기존의 중앙 집계 질의 처리 기법은 센서 노드들의 에너지 소모가 많은 문제점이 있기 때문에 최근에는 센서 노드들의 에너지 소모를 줄이기 위해 센서 노드들에서 센싱된 데이터에 대해 집계 질의를 처리한 후 그 결과를 서버로 보내는 네트워크 내 집계 질의 처리 기법에 대한 연구가 활발히 진행되고 있다[3]. 대표적인 네트워크 내 집계 질의 처리 기법으로는 라우팅 알고리즘에 중점을 둔 TAG(Tiny AGgregation)와 IWQE(Itinerary-based Window Query Execution)가 있고, 데이터 구조에 중점을 둔 q-digest(quantile-digest)와 SMC(Secure Median Computation)가 있다. TAG는 계층형 라우팅을 이용한 집계 질의 처리 기법[5]이고, IWQE는 Itinerary 라우팅을 이용한 집계 질의 처리 기법이다[9]. 이렇게 TAG와 IWQE는 효율적인 집계 질의 처리를 위해 라우팅 알고리즘을 제시하였으나 센서 노드들의 에너지 소모가 크고, 질의 처리 결과 정확도가 떨어지고, 또한 질의 처리 시간이 오래 걸리는 문제점들을 가지고 있다. 또한, TAG와 IWQE는 MEDIAN과 HISTOGRAM 집계 연산을 고려하지 못한 단점을 가지고 있다.

q-digest는 MEDIAN과 HISTOGRAM 집계 연산을 위해 트리 데이터 구조를 이용하는 근사 집계 질의 처리 기법[7]이고, SMC는 MEDIAN과 HISTOGRAM 집계 연산을 위해 버킷 데이터 구조를 이용하는 근사 집계 질의 처리 기법이다[8]. 이렇게 q-digest와 SMC는 효율적인 집계 질의 처리를 위해 데이터 구조를 제시하였으나 센서 노드들의 에너지 소모가 크고, 질의 처리 결과 정확도가 떨어지

고, 질의 처리 시간이 오래 걸리는 문제점들을 가지고 있다. 또한, q-digest와 SMC는 Composite 집계 질의 유형(동시에 2개 이상의 집계 질의를 수행)을 고려하지 않은 단점을 가지고 있다.

본 논문에서는 이러한 기존 집계 질의 처리 기법의 문제점을 해결하고 무선 센서 네트워크에서 보다 효율적인 집계 질의 처리를 위해 버킷 기반의 병렬 집계 질의 처리 기법인 BPA(Bucket-based Parallel Aggregation)를 제시하였다. BPA는 질의 영역 내의 센서 노드들의 정보를 수집하여 질의 영역을 센서 노드 분포에 따라 여러 개의 셀(Cell)로 분할하여 쿼드 트리(Quad-Tree)로 구성하고, 쿼드 트리 노드의 각 셀 영역을 Itinerary 라우팅에 따라 집계 질의를 병렬로 처리한다. BPA는 집계 질의에 대해 병렬로 처리하기 때문에 센서 노드들의 에너지 소모가 적고 질의 영역이 크거나 센서 노드들의 개수가 많아도 질의 처리 시간이 빠르다.

그리고 BPA는 질의 영역내 센서 노드 중 집계 질의 처리에 참여하지 못하는 Missing 노드 발생을 최대한 줄이고 Missing 노드 발생시에는 질의 시작 센서 노드와 가장 가까운 센서 노드에게 데이터를 전송한다. 또한, BPA의 센서 노드들은 데이터를 이중 전송함으로써 전송 오류로 인한 데이터 손실을 줄인다. 이렇게 BPA는 Missing 노드를 최소화하고 중복된 데이터를 제거하여 데이터를 이중 전송하기 때문에 질의 처리 결과 정확도를 높일 수 있다.

BPA는 MEDIAN, HISTOGRAM 집계 연산을 위해 버킷 기반의 데이터 구조를 이용한다. 버킷 데이터 구조는 Composite 집계 질의 유형을 고려하여 수집된 데이터의 최소값, 최대값, 버킷내 평균값, 버킷내 데이터 개수를 저장한다. 그리고 버킷내 데이터 크기를 줄이기 위해 가변 길이 비트 압축 코딩 기법(Variable Bit Compression Coding Technique)[4, 11]을 사용하여 데이터를 압축한다. 이렇게 BPA는 버킷 데이터 구조와 가변 길이 비트 압축 코딩 기법을 이용하기 때문에 센서 노드들의 에너지 소모를 줄일 수 있다.

BPA는 버킷 데이터 구조를 버킷내 데이터 개수에 따라 적응적으로 분할 및 합병하므로 데이터의 분포가 편중되더라도 질의 처리 결과 정확도를 높일 수 있다. 또한, BPA는 각 센서 노드에 필터링 허용 범위를 설정하여 데이터 전송시 필터링 허용 범위를 벗어난 경우에만 데이터를 전송하는 데이터 전송 필

터링을 수행하기 때문에 센서 노드들의 에너지 소모를 줄일 수 있다.

2. 관련 연구

2.1 TAG(Tiny AGgregation)

TAG는 집계 질의 처리를 위해 계층형 라우팅을 이용한 네트워크내 집계 질의 처리 기법이다. 즉, TAG는 네트워크 내 집계 질의 처리를 위해 무선 센서 네트워크 전체 영역에 대해서 계층형 라우팅을 구성한다. 그림 1은 TAG의 계층형 라우팅 구조를 보여준다.

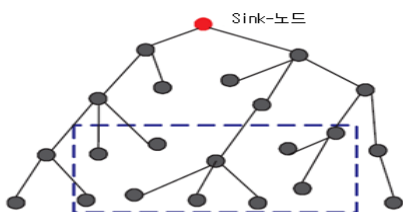


그림 1. TAG의 계층형 라우팅 구조

그림 1에서 보는 바와 같이 TAG는 모든 센서 노드들간에 부모 센서 노드와 자식 센서 노드 관계를 설정하여 계층형 라우팅을 구성한다. 질의 영역내 자식 센서 노드들은 센싱된 데이터를 부모 센서 노드에게 전송하고, 부모 센서 노드는 중간 집계 질의 결과를 다시 상위 부모 센서 노드에게 계속해서 전송한다. 그리고 Sink-노드는 최종 집계 질의 처리 결과를 서버에게 반환한다.

TAG는 무선 센서 네트워크 내부에서 집계 질의를 처리하여 중앙 처리 방법 보다 전체적으로 센서 노드들의 에너지 소모를 줄였다[9]. 그래서 TAG는 질의 영역이 크거나 센서 노드들이 많은 경우에는 효율적이다. 그러나 질의 영역에 포함되지 않는 센서 노드들의 에너지 소모가 많이 발생하고, 라우팅 유지로 인한 센서 노드들의 에너지 소모가 큰 문제점을 가지고 있다[6]. 그리고, TAG는 MEDIAN, HISTOGRAM 집계 연산을 고려하지 못한 단점을 가지고 있다.

2.2 WQE(Itinerary-based Window Query Execution)

IWQE는 집계 질의 처리를 위해 Itinerary 라우팅을 이용한 네트워크 내 집계 질의 처리 기법이다.

IWQE는 무선 센서 네트워크 전체 영역에 대해 라우팅을 미리 구성하지 않고 사용자 질의가 주어질 시기에 관심 지역에 대해서만 일시적으로 라우팅을 구성하여 집계 질의를 처리한다. 그림 2는 IWQE의 Itinerary 라우팅 구조를 보여준다.

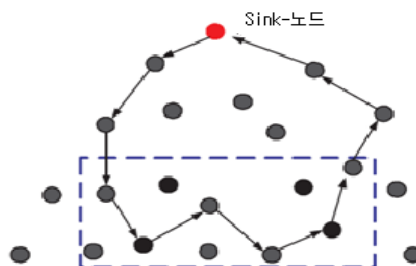


그림 2. IWQE의 Itinerary 라우팅 구조

그림 2에서와 같이 IWQE는 Itinerary 라우팅을 이용하여 질의 영역 내 센서 노드들의 센싱 데이터에 대해 집계 질의를 처리하고, Sink-노드가 최종 집계 질의 결과를 서버에게 반환한다.

IWQE에서는 질의 영역에 포함되지 않는 센서 노드들의 불필요한 에너지 소모가 없고, 라우팅 유지 비용이 없으며, 또한 질의 영역이 작거나 센서 노드들이 적은 경우 효율적이다. 그러나 Missing 노드 발생으로 인해 질의 결과의 정확도가 떨어지고, 데이터 전송 오류를 줄이기 위한 브로드 캐스팅 전송으로 인해 중복 데이터가 발생하기 때문에 질의 처리 결과 정확도가 저하되고 센서 노드들의 에너지 소모가 커진다. 그리고 센서 노드들이 많거나 질의 영역이 큰 경우 질의 처리 시간이 오래 걸리는 문제점을 가지고 있다. 또한, IWQE는 MEDIAN과 HISTOGRAM 집계 연산을 고려하지 못한 단점을 가지고 있다.

2.3 q-digest(quantile-digest)

q-digest는 MEDIAN과 HISTOGRAM 집계 연산 처리를 위해 트리 데이터 구조를 이용한 근사 집계 질의 처리 기법이다. q-digest의 트리 데이터 구조는 다음과 같은 특성을 가지고 있다. 루트 노드는 $[1, \sigma]$ 범위 값을 가지고, 자식 노드는 $[1, \sigma/2]$ 와 $[\sigma/2+1, \sigma]$ 범위 값을 가진다. 그리고 각 노드는 범위 안에 포함되는 센싱 데이터의 개수를 저장한다. 그림 3은 q-digest의 트리 데이터 구조와 예를 보여준다.

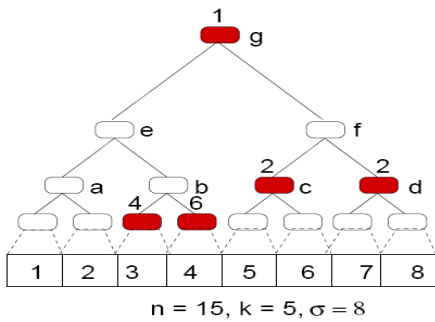


그림 3. q-digest의 트리 데이터 구조

그림 3에서 보는 바와 같이 전체 센싱 데이터 범위 σ 는 1~8, 데이터 개수 n 은 15, 그리고 압축률 k 는 5로 가정한다. 따라서 루트 노드 g 는 1~8의 범위 값을 가지고, 자식 노드 e 와 f 는 각각 1~4, 5~8의 범위 값을 가진다. 그리고 자식 노드 a, b, c, d 는 각각 1~2, 3~4, 5~6, 7~8의 범위 값을 가진다. 또한 각 노드에는 해당 범위에 속하는 센싱된 데이터의 개수가 저장된다. 특히 q-digest의 각 노드는 자신의 노드내 데이터 개수에 부모 노드와 이웃 노드의 데이터 개수를 더한 값이 압축률 n/k 보다 작을 경우 데이터를 압축한다.

q-digest는 트리 데이터 구조를 이용하여 집계 질의를 처리함으로써 MEDIAN과 HISTOGRAM 집계 연산시 데이터 전송량을 크게 줄여 좋은 성능을 보인다. 그러나 트리 데이터 구조 구성을 위한 추가 정보로 인해 센서 노드들의 에너지 소모가 크고, 데이터 압축으로 인해 집계 질의 결과 정확도가 떨어지며, 노드 범위값이 고정되어있기 때문에 센싱된 데이터가 편중될 경우 집계 질의 처리 결과의 정확도가 떨어지는 문제점을 가지고 있다[1]. 그리고, q-digest는 Composite 집계 질의 유형을 고려하지 않은 단점을 가지고 있다.

2.4 SMC(Secure Median Computation)

SMC는 MEDIAN과 HISTOGRAM 집계 연산을 위해 버킷 데이터 구조를 이용한 근사 집계 질의 처리 기법이다[8]. SMC의 버킷 데이터 구조는 다음과 같은 특성을 가지고 있다. SMC는 센싱된 데이터를 저장하기 위해 b 개의 버킷을 구성하는데, 각 버킷 B_i 는 $q_i \sim q_{i+1}$ 범위값과 센싱 데이터의 개수를 저장한다. 특히 SMC는 특정 집계값이 저장되어 있는 버킷을 분할하거나 병합함으로써 다음 질의 때 집계 질

의 처리 결과 정확도를 높인다. 그림 4는 SMC의 버킷 데이터 구조와 예를 보여준다.

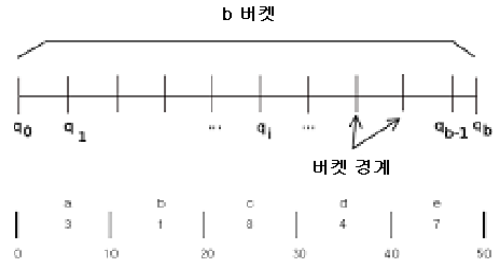


그림 4. SMC의 버킷 데이터 구조

그림 4에서 보는 바와 같이 버킷 데이터 구조의 전체 데이터 범위는 0~50 범위값을 가지며 5개의 버킷으로 구성되어 있다. 그리고 버킷 a 는 0~10, b 는 11~20, c 는 21~30, d 는 31~40, e 는 41~50의 범위값을 가지며, 각 버킷에는 3개, 1개, 8개, 4개, 7개의 센싱 데이터 개수를 저장하고 있다.

SMC는 버킷 데이터 구조를 이용하여 집계 질의를 처리함으로써 MEDIAN과 HISTOGRAM 집계 연산시 데이터 전송량을 크게 줄여 좋은 성능을 보인다. 그러나 고정된 버킷 데이터 구조 전송으로 인해 센서 노드들의 에너지 소모가 크고, 버킷 데이터 구조 병합으로 인해 집계 질의 처리 결과 정확도가 떨어지는 문제점을 가지고 있다[1]. 또한, SMC는 Composite 집계 질의 유형에 대해서 고려하지 않은 단점을 갖고 있다.

3. BPA(Bucket-based Parallel Aggregation)

3.1 라우팅

BPA는 센서 노드들의 에너지 소모를 줄이고 질의 처리 시간을 빠르게 하기 위해 계층형 라우팅을 구성하고 센서 노드들의 정보를 수집한다. 그리고 수집된 센서 노드들의 정보를 이용하여 질의 영역을 센서 노드 분포에 따라 여러 개의 셀로 분할하여 쿼드 트리로 구성하고, 쿼드 트리의 각 셀 영역을 Itinerary 라우팅을 따라 집계 질의를 병렬로 처리한다. 그림 5는 계층형 라우팅 구조와 센서 노드 정보 수집을 위한 MBR 구성 예를 보여준다.

그림 5에서 보는 바와 같이 계층형 라우팅을 구성하기 위해 질의 영역의 중심점에 가장 가까운 센서

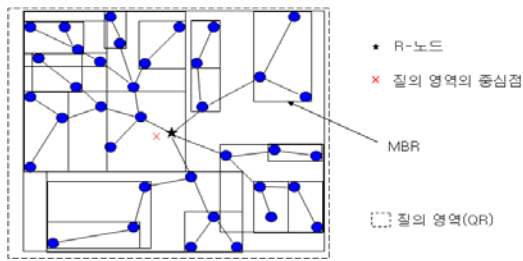


그림 5. 계층형 라우팅 구조와 MBR 구성 예

노드를 검색하여 계층형 라우팅의 R-노드(Root 노드)로 선정한다. 그리고 R-노드부터 시작하여 자식 노드를 갖고 있는 센서 노드들은 자신과 자식 센서 노드들을 포함하는 최소 경계 사각형 MBR을 구성하고 MBR 영역내 센서 노드들의 정보를 수집하여 부모 센서 노드에게 전송한다.

BPA는 이렇게 수집된 센서 노드들의 정보를 이용해 질의 영역을 여러 개의 셀로 분할하고 쿼드 트리(Quad-Tree)를 구성한다. 그림 6은 쿼드 트리 구성 예를 보여준다.

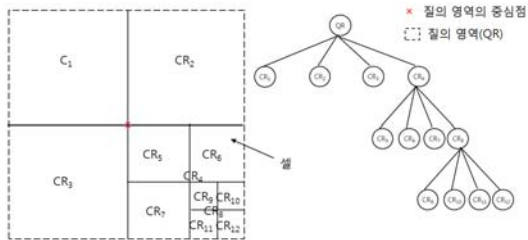


그림 6. 쿼드 트리 구성 예

그림 6에서 보는 바와 같이 질의 영역 QR은 CR1, CR2, CR3, CR4로 분할되고, CR4는 다시 CR5, CR6, CR7, CR8로 분할되며, CR8은 다시 CR9, CR10, CR11, CR12로 분할된다. 특히, 질의 영역은 셀내 센서 노드 개수가 셀내 최대 센서 노드 개수 임계치보다 작아질 때까지 분할된다.

BPA는 각각의 쿼드 트리 셀내에서 대표 센서 노드인 C-노드를 선정하고, 쿼드 트리 각 셀 영역에서 집계 질의를 병렬로 처리한다. 각 셀 영역의 집계 질의 처리 결과는 부모 노드 셀의 대표 센서 노드로 연속적으로 전달된다. 그림 7은 각 셀의 대표 센서 노드가 집계 질의 처리 결과를 부모 노드 셀의 대표 센서 노드에게 연속적으로 전달하는 과정을 보여준다.

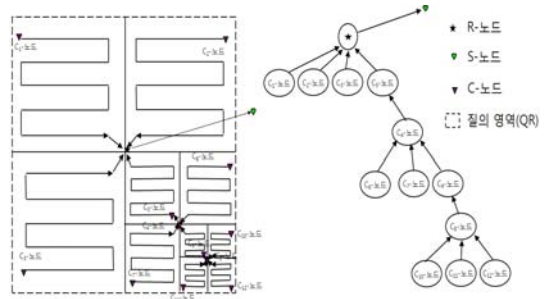


그림 7. 집계 질의 처리 결과 전달 과정

그림 7에서 보는 바와 같이 C₁₀-노드, C₁₁-노드, C₁₂-노드의 집계 질의 처리 결과는 C₈-노드로 전달되고, C₈-노드의 집계 질의 처리 결과는 C₉-노드로 전달된다. 그리고 C₆-노드, C₇-노드, C₉-노드의 집계 질의 처리 결과는 C₄-노드에게 전달되고, C₄-노드의 집계 질의 처리 결과는 C₅-노드에게 전달된다. 마지막으로 C₁-노드, C₂-노드, C₃-노드, C₅-노드의 집계 질의 처리 결과는 R-노드로 전달되고, R-노드의 집계 질의 처리 결과는 질의 시작 센서 노드인 S-노드에게 반환된다.

BPA는 쿼드 트리 셀내의 집계 질의 처리를 위해 Itinerary 라우팅을 이용한다. 그림 8은 쿼드 트리 셀내의 라우팅 과정 예를 보여준다.

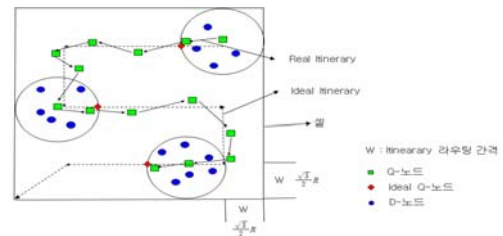


그림 8. 쿼드 트리 셀내 라우팅 과정 예

그림 8에서 보는 바와 같이 각 셀내에서 질의 전송 센서 노드인 Q-노드는 Ideal Itinerary 라우팅에 따라 통신 범위내에 있는 데이터 전송 센서 노드인 D-노드들의 데이터를 수집하여 집계 질의를 처리한 후 다음 Q-노드에게 집계 질의 처리 결과를 전송한다. 이때 실제 Q-노드의 라우팅 경로는 Real Itinerary 라우팅이며 각 Itinerary 라우팅 간격 W는 센서 노드들의 통신 범위 R을 이용하여 $\frac{\sqrt{3}}{2}R$ 로 설정된다.

BPA는 Itinerary 라우팅 과정에서 집계 질의 처리에 참여하지 못하는 Missing 노드를 최대한 줄이기 위해 Q-노드의 통신 범위내 있는 D-노드 중 Ideal Q-노드에 가장 가까운 센서 노드를 다음 Q-노드로 선정한다. Ideal Q-노드란 실제로 존재하지는 않지만 최적의 라우팅을 위해 가상으로 설정된 센서 노드로써 Ideal Itinerary 라우팅 선상과 Q-노드의 통신 범위와의 교차점을 의미한다.

만약, Missing 노드가 발생하면 다음과 같이 처리된다. 즉, 질의 주기안에 Q-노드나 D-노드로 선정되지 않아 Missing 노드로 인식된 센서 노드는 S-노드와 가장 가까운 센서 노드에게 자신의 데이터를 전송한다.

BPA의 D-노드와 Q-노드는 네트워크 전송 오류로 인한 집계 질의 처리 결과의 오류를 줄이기 위해서 센서 노드간에 이중 데이터 전송을 수행한다. 그림 9는 BPA의 D-노드 이중 데이터 전송 과정 예를 보여준다.

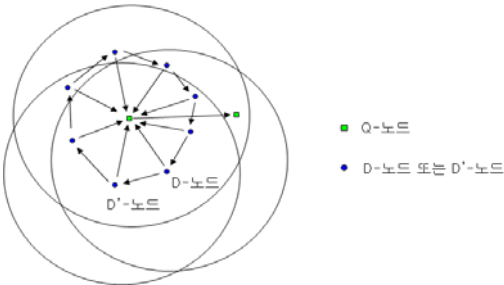


그림 9. D-노드의 이중 데이터 전송 과정 예

그림 9에서와 같이 D-노드는 센싱된 데이터를 Q-노드와 이웃 다른 D'-노드에게 전송한다. 즉, D-노드는 Q-노드와 D-노드의 이웃 센서 노드 중 통신 범위내 Q-노드를 포함하고 있고 Right-hand rule을 만족하는 D-노드인 D'-노드에게 데이터를 이중 전송한다. 그러면 D'-노드는 D-노드의 데이터와 자신이 센싱한 데이터를 Q-노드에게 전송한다.

그림 10은 BPA의 Q-노드 이중 데이터 전송 과정 예를 보여준다.

그림 10에서 보는 바와 같이 Q-노드는 D-노드로부터 수집된 데이터를 다음 Q-노드와 다른 이웃 Q'-노드에게 전송한다. 즉, Q-노드는 다음 Q-노드와 Q-노드의 이웃 센서 노드 중 통신 범위내 다음 Q-노드를 포함하고 있고 다음 Q-노드에 가장 가까

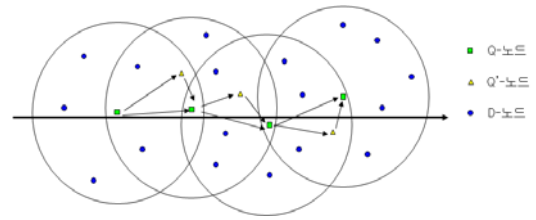


그림 10. Q-노드의 이중 전송 과정 예

운 Q'-노드에게 데이터를 이중 전송한다. 그리고 Q'-노드는 Q-노드의 데이터와 자신이 센싱한 데이터를 다음 Q-노드에게 전송한다.

BPA에서는 센서 노드가 데이터를 전송할 때 자신의 아이디를 추가하여 이중으로 데이터를 전송하고, 데이터를 전송받은 센서 노드는 중복된 아이디의 데이터를 제거하여 전송함으로써 데이터 중복 문제를 해결한다.

3.2 데이터 구조

BPA는 MEDIAN이나 HISTOGRAM과 같은 집계 질의 처리시 센서 노드들의 에너지 소모를 줄이기 위해 버킷 기반의 데이터 구조를 이용하고 가변 길이 비트 압축 코딩 기법을 사용한다.

D-노드의 데이터 저장 구조는 센서 노드 ID를 나타내는 ID와 센싱 데이터를 나타내는 Value로 구성된다. 그리고 Q-노드의 데이터 저장 구조는 센서 노드 ID를 나타내는 ID, 수집된 데이터의 최소값과 최대값을 나타내는 MinValue, MaxValue, 버킷내 평균값과 데이터 개수를 나타내는 AValue, Count_i로 구성된다. 그림 11은 Q-노드의 버킷 데이터 구조를 보여준다.

그림 11에서 보는 바와 같이 전체 버킷 리스트는 BLSize(BLMax~BLMin) 크기를 가지며, 각 버킷의 초기 크기는 버킷 한 개의 최대 크기 BMax를 가진다. 그리고 버킷 번호 i별로 버킷내 데이터들의 평균값 AValue_i와 버킷내 데이터 개수 Count_i가 버킷 리스트에 저장된다. (식 1)은 AValue를 구하는 식이다.

$$AValue_i = \frac{\sum_{k=1}^{Count_i} Value_i^k}{Count_i} \tag{식 1}$$

식 1에서 AValue_i는 버킷 i에 속하는 데이터들의 평균값이고, Count_i는 버킷 i에 속하는 데이터들의 개수이며, Value_i^k는 버킷 i에 속하는 k번째 데이터

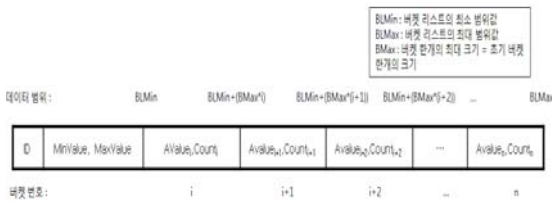


그림 11. Q-노드의 버킷 데이터 구조

를 나타낸다.

BPA는 데이터 압축을 위해 가변 길이 비트 압축 코딩 기법을 이용한다. 즉, BPA는 센서 노드 데이터 전송시 데이터 전송량을 줄이기 위해 D-노드와 Q-노드의 데이터 중 가장 큰 크기를 차지하는 D-노드의 Value와 Q-노드의 MinValue, MaxValue, AValue를 압축하여 저장한다.

또한 BPA는 질의 처리 결과 정확도를 높이기 위해 버킷 데이터 구조를 버킷내 데이터 개수에 따라 적응적으로 분할 및 합병한다.

그림 12와 그림 13은 버킷 리스트 크기 BSize가 60(MinValue = 0, MaxValue = 60), 버킷 한 개의 최대 크기 BMax가 10, 버킷 한 개의 최소 크기 BMin가 2, 버킷내 최대 데이터 개수 BMaxC가 5인 경우 버킷 분할과 버킷 병합 예를 보여준다.

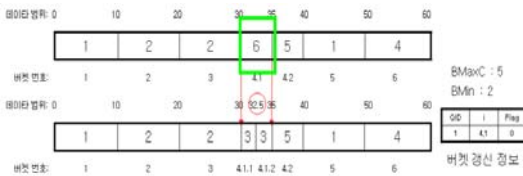


그림 12. 버킷 분할 예

그림 12에서 보는 바와 같이 버킷 4.1이 버킷내 최대 데이터 개수 BMaxC = 5를 초과하면서 버킷 한 개의 최소 크기 BMin = 2 이상이기 때문에 버킷 4.1.1과 4.1.2로 분할된다. 이러한 버킷 분할은 버킷 크기가 BMin 이상일 때까지 분할되기 때문에 버킷 4.1.1.1과 4.1.1.2는 더 이상 분할되지 않는다.

그림 13에서 보는 바와 같이 두 버킷 4.1.1과 4.1.2는 데이터 개수의 합이 버킷내 최대 데이터 개수 BMaxC = 5를 미달하면서 버킷 한개의 최대 크기 BMax = 10 이하이기 때문에 버킷 4.1로 병합된다.

BPA는 센서 노드들의 에너지 소모를 줄이기 위해 각 센서 노드에 필터링 허용 범위를 설정하고, 데



그림 13. 버킷 병합 예

이타 전송시 필터링 허용 범위를 벗어난 경우에만 데이터를 전송한다.

D-노드는 필터링 허용 범위 DF를 할당받는데, D-노드의 DF는 (식 2)에서와 같이 전체 필터링 허용 범위 TF와 질의 영역내 센서 노드 개수 TSC를 이용하여 계산된다. D-노드는 DF를 벗어난 경우에만 데이터를 전송한다.

$$DF = \frac{TF}{TSC} \tag{식 2}$$

Q-노드는 초기 필터링 허용 범위 IQF를 할당받고 집계 질의 처리 과정에서 필터링 허용 범위 QF를 재설정한다. 그리고 Q-노드는 버킷 리스트의 합이 필터링 허용 범위 QF를 벗어난 경우에만 데이터를 전송한다.

Q-노드의 IQF는 (식 3)에서와 같이 전체 필터링 허용 범위 TF와 질의 영역내 센서 노드 개수 TSC를 이용하여 구해진다.

$$IQF = \frac{TF}{TSC} \tag{식 3}$$

Q-노드는 집계 질의 처리 과정에서 Q-노드의 IQF, D-노드의 DF, Q-노드가 통신 범위내에서 데이터를 전송한 D-노드 개수 DSC, 이전 Q-노드의 QF' 등을 이용하여 QF를 재설정한다. 이전 Q-노드가 있으면서 이전 Q-노드가 데이터를 전송한 경우에는 Q-노드의 IQF에 데이터를 전송한 D-노드들의 DF와 이전 Q-노드의 QF'를 더한 값으로 QF를 재설정한다. 이는 (식 4)와 같이 나타낼 수 있다.

$$QF = IQF + (DF \times DSC) + QF' \tag{식 4}$$

이전 Q-노드가 있으면서 이전 Q-노드가 데이터를 전송하지 않은 경우와 이전 Q-노드가 없는 경우에는 Q-노드의 IQF에 데이터를 전송한 D-노드들의 DF를 더한 값으로 QF를 재설정한다. 이는 (식 5)와 같이 나타낼 수 있다.

$$QF = IQF + (DF \times DSC) \quad (\text{식 } 5)$$

그림 14와 그림 15는 센서 노드들의 데이터 전송 필터링 예를 보여준다.

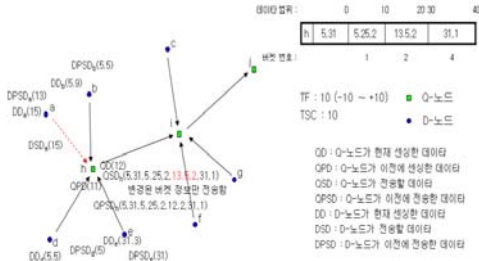


그림 14. 데이터 전송 필터링 예 1

그림 14에서 보는 바와 같이 D-노드들의 DF가 1이므로 D-노드 b, d, e는 데이터를 전송하지 않고, D-노드 a는 센싱 데이터가 필터링 허용 범위 DF_a 를 벗어났기 때문에 Q-노드 h에게 $DSD_a(15)$ 를 전송한다. 그리고 Q-노드 h의 QF_h 는 이전 Q-노드가 없기 때문에 (식 5) 공식을 이용하여 구하면 $1+(1 \times 1)=2$ 가 된다. Q-노드 h의 $QPSD_h$ 의 버킷 리스트의 합은 65.5이고 QSD_h 의 버킷 리스트의 합은 68.5이므로 필터링 허용 범위 QF_h 를 벗어났기 때문에 Q-노드 h는 Q-노드 i에게 $QSD_h(5, 21, 5, 25, 2, 13, 5, 2, 31, 1)$ 를 전송한다.

그림 15에서 보는 바와 같이 D-노드들의 DF가 1이므로 D-노드 g는 데이터를 전송하지 않고, D-노드 c와 f는 센싱 데이터가 필터링 허용 범위 DF_c 와 DF_f 를 벗어났기 때문에 Q-노드 i에게 $DSD_c(36.5)$ 와 $DSD_f(37)$ 를 전송한다. 그리고 Q-노드 i의 QF_i 는 이전 Q-노드가 있으면서 이전 Q-노드가 데이터를 전송하였기 때문에 (식 4)를 이용하여 구하면 $1+(1 \times 2)+2=5$ 이다. Q-노드 i의 $QPSD_i$ 의 버킷 리스트의 합

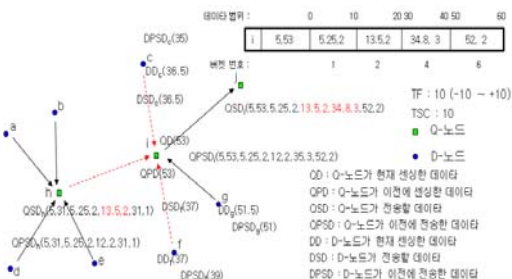


그림 15. 데이터 전송 필터링 예 2

은 243.5이고 QSD_i 의 버킷 리스트의 합은 245.9이므로 필터링 허용 범위 QF_i 를 벗어나지 않았기 때문에 Q-노드 i는 Q-노드 j에게 데이터를 전송하지 않는다.

3.3 알고리즘

BPA 알고리즘은 Itinerary 라우팅을 생성하는 과정과 생성된 Itinerary 라우팅에 따라 집계 질의를 전송하고 집계 질의 결과를 반환하는 과정으로 구성된다. 그림 16은 전체 BPA 알고리즘을 보여준다.

```

함수 BPA_Process(QUERY q, SENSOR_NODE nodelist[])
입력 q : 질의 정보, nodelist[] : 센서 노드들의 정보
출력 BUCKET_LIST bkl : 버킷 리스트 정보

1 : SENSOR_NODE mode = RNode_Search(q, nodelist) // R-노드 선정
2 : HROUTE hr = HRoute_Build(mode, nodelist) // 계층형 라우팅 구성
3 : HR_SENSOR_NODE_INFO sni = SensorInfo_Collection(hr, MBR()) // 센서 노드
4 : 정보 수집
5 : QT_NODE qn = QT_Build(sni, nodelist) // 쿼드 트리 구성
6 : IROUTE ir = QT_CNode_Itinerary(qn, nodelist) // 대표 센서 노드 선정 및 Itinerary
7 : 라우팅 설정
8 : if(QNodes() || DNodes())
9 : {
10 :   QNode_Query_Send(q, ir) // Q-노드의 질의 전송
11 :   If (!QNode_Filtering())
12 :   {
13 :     Bucket_Build(Bucket_Update()) // 버킷 데이터 구조 생성 및 갱신
14 :     Compress_Value() // 데이터 압축
15 :     BUCKET_LIST bkl = QNode_Duplicate_Result_Send(ir) // Q-
16 :       노드의 질의 결과 이중 전송
17 :     return bkl;
18 :   }
19 :   If (!DNode_Filtering())
20 :   {
21 :     Bucket_Build(Bucket_Update()) // 버킷 데이터 구조 생성 및
22 :       갱신
23 :     Compress_Value() // 데이터 압축
24 :     BUCKET_LIST bkl = DNode_Duplicate_Result_Send(ir) // D-
25 :       노드의 질의 결과 이중 전송
26 :     return bkl;
27 :   }
28 : }
29 : else
30 : {
31 :   MissingNode_Send_Result() // Missing 노드 처리
32 : }
    
```

그림 16. BPA 알고리즘

그림 16에서 보는 바와 같이 BPA 알고리즘은 먼 저 질의 영역의 중심점에 가장 가까운 R-노드를 선정하고, 선정된 R-노드를 기준으로 계층형 라우팅을 구성한 후 센서 노드들의 정보를 수집한다. 그리고 수집된 센서 노드들의 정보를 기반으로 쿼드 트리를 구성하고, 쿼드 트리 각 셀내 대표 센서 노드 C-노드를 선정하여 Itinerary 라우팅을 생성한다.

또한 BPA 알고리즘은 각 C-노드를 시작으로 Itinerary 라우팅에 따라 집계 질의를 처리한다. 즉, Q-노드는 Itinerary 라우팅에 따라 집계 질의를 전송하고, D-노드는 집계 질의 결과를 반환한다. 이때 Q-노드와 D-노드는 필터링 여부를 판단하여 필터링이 안된 경우에는 변경된 버킷 정보만을 압축하여 다음 센서 노드에게 전송한다. 만약, 집계 질의 처리 과정에서 Q-노드나 D-노드로 선정되지 않은 Missing 센서 노드들은 질의 시작 센서 노드인 S-노드와 가장 가까운 센서 노드에게 질의 결과를 전송한다.

4. 성능 평가

4.1 성능 평가 환경

성능 평가에 사용된 시스템의 하드웨어 사양은 Intel Core 2.4GHz CPU, 2GB RAM, 300GB HDD이며 운영체제로는 Windows XP를 사용하였다. 그리고 시뮬레이션을 위해 MFC(Microsoft Foundation Class Library)를 사용하였으며, 성능 평가에서는 표 1에서 보여주는 바와 같이 13가지 파라미터를 설정하였다.

특히 BPA와 TAG-q-digest, TAG-SMC, IWQE-q-digest, IWQE-SMC에 대해서 MEDIAN 질의의 성능 평가를 BPA, q-digest, SMC의 데이터 구조를 최적화하여 질의 처리 결과 정확도가 95% 이상 유지되도록 하였다.

4.2 성능 평가 결과

표 1. 성능 평가를 위한 파라미터

파라미터	값
센서 네트워크 크기	1,650 * 1,650 m ²
센서 노드의 통신 범위	30 m 내
센서 노드의 총 전송 횟수	10,000,000 번
센서 노드의 총 에너지량	10,000 J
1회 데이터 전송시 에너지 소비량	1 mJ
1회 데이터 전송 크기	30 Byte
1회 데이터 전송 시간	0.12 ms
데이터 전송시 오류 발생 주기	10% 예) 전송 10회 마다 한번씩
센서 노드의 센싱 범위	온도 70°C 예) 70°C(-10°C ~ +60°C)
센싱 데이터의 편차 범위	온도 0.5°C 예) 0.5°C(-0.5°C ~ +0.5°C)
센서 노드 개수	2만개, 4만개, 6만개, 8만개, 10만개
연속 질의 횟수	200회, 400회, 600회, 800회, 1,000회
질의 영역 크기	330*330 m ² , 660*660 m ² , 990*990 m ² , 1,320*1,320 m ² , 1,650*1,650 m ²

그림 17은 센서 노드 개수에 따라 BPA와 TAG, IWQE에 대해서 AVG 질의의 처리 결과 정확도, 에너지 소모, 질의 처리 시간의 성능 평가 결과를 보여준다.

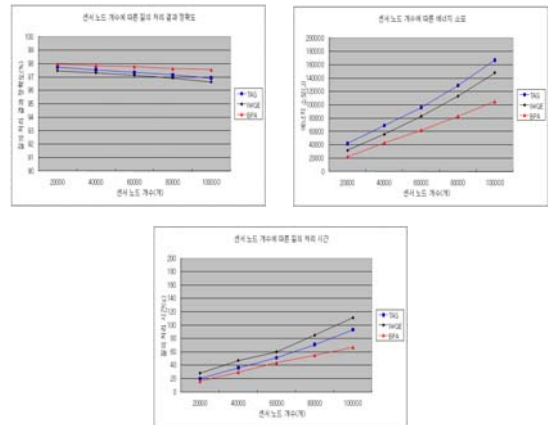


그림 17. 센서 노드 개수에 따른 AVG 집계 연산 성능 평가

그림 17에서 보는 바와 같이 질의 처리 결과 정확도는 BPA가 IWQE 보다 평균 29%, TAG 보다 평균 18% 성능이 향상되었고, 에너지 소모는 BPA가 TAG 보다 평균 59%, IWQE 보다 평균 37% 성능이 향상되었다. 그리고 질의 처리 결과 시간은 BPA가 IWQE 보다 평균 57%, TAG 보다 평균 28% 성능이 향상되었다.

그림 18은 센서 노드 개수에 따라 BPA와TAG-q-digest, TAG-SMC, IWQE-q-digest, IWQE-SMC에 대해서 MEDIAN 질의의 에너지 소모와 질의 처리 시간의 성능 평가 결과를 보여준다.

그림 18에서 보는 바와 같이 에너지 소모는 BPA가 TAG+q-digest 보다 평균 101%, TAG+SMC 보다 평균 88%, IWQE+q-digest 보다 평균 66%,

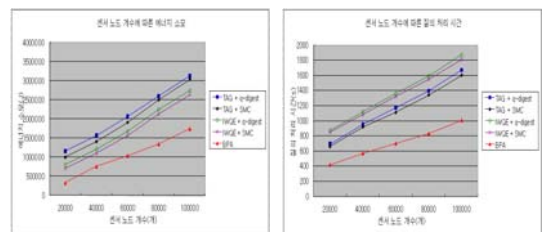


그림 18. 센서 노드 개수에 따른 MEDIAN 집계 연산 성능 평가

IWQE+SMC 보다 평균 55% 성능이 향상되었다. 그리고 질의 처리 시간은 BPA가 IWQE+q-digest 보다 평균 93%, IWQE+SMC 보다 평균 87%, TAG+q-digest 보다 평균 66%, TAG+SMC 보다 평균 59% 성능이 향상되었다.

그림 19는 연속 질의 횟수에 따라 BPA와 TAG, IWQE에 대해서 AVG 질의의 처리 결과 정확도, 에너지 소모, 질의 처리 시간의 성능 평가 결과를 보여준다.

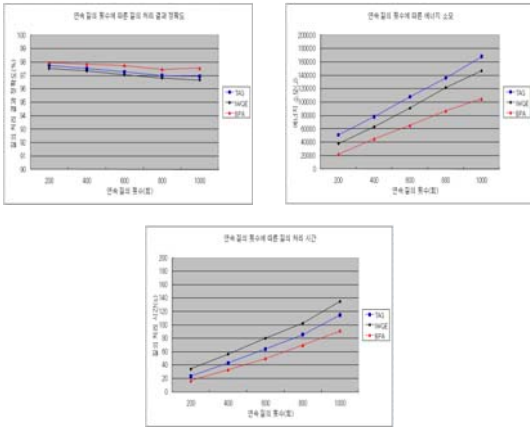


그림 19. 연속 질의 횟수에 따른 AVG 집계 연산 성능 평가

그림 19에서 보는 바와 같이 질의 처리 결과 정확도는 BPA가 IWQE 보다 평균 28%, TAG 보다 평균 18% 성능이 향상되었고, 에너지 소모는 BPA가 TAG 보다 평균 66%, IWQE 보다 평균 42% 성능이 향상되었다. 그리고 질의 처리 시간은 BPA가 IWQE 보다 평균 56%, TAG 보다 평균 26% 성능이 향상되었다.

그림 20은 연속 질의 횟수에 따라 BPA와 TAG+q-digest, TAG+SMC, IWQE+q-digest, IWQE+SMC에 대해서 MEDIAN 질의의 에너지 소모와 질

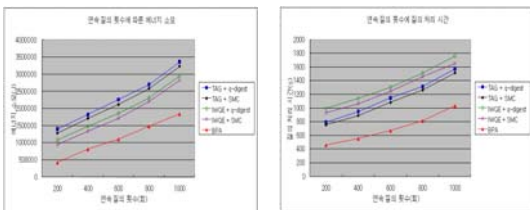


그림 20. 연속 질의 횟수에 따른 MEDIAN 집계 연산 성능 평가

의 처리 시간의 성능 평가 결과를 보여준다.

그림 20에서 보는 바와 같이 에너지 소모는 BPA가 TAG+q-digest 보다 평균 104%, TAG+SMC 보다 평균 93%, IWQE+q-digest 보다 평균 71%, IWQE+SMC 보다 평균 59% 성능이 향상되었다. 그리고 질의 처리 시간은 BPA가 IWQE+q-digest 보다 평균 90%, IWQE+SMC 보다 평균 80%, TAG+q-digest 보다 평균 64%, TAG+SMC 보다 평균 56% 성능이 향상되었다.

그림 21은 질의 영역 크기에 따라 BPA와 TAG, IWQE에 대해서 AVG 질의의 처리 결과 정확도, 에너지 소모, 질의 처리 시간의 성능 평가 결과를 보여준다.

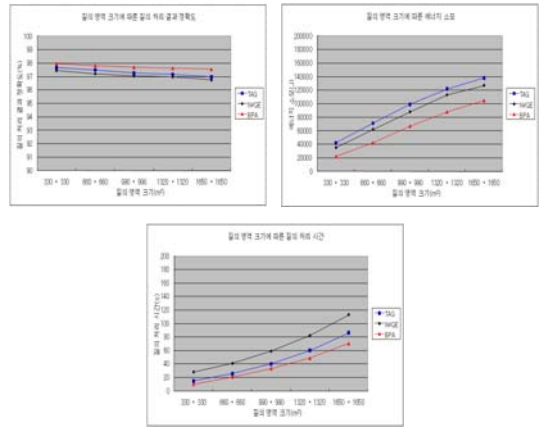


그림 21. 질의 영역 크기에 따른 AVG 집계 연산 성능 평가

그림 21에서 보는 바와 같이 질의 처리 결과 정확도는 BPA가 IWQE 보다 평균 28%, TAG 보다 평균 18% 성능이 향상되었고 에너지 소모는 BPA가 TAG 보다 평균 45%, IWQE 보다 평균 31% 성능이 향상되었다. 그리고 질의 처리 시간은 BPA가 IWQE 보다 평균 75%, TAG 보다 평균 23% 성능이 향상되었다.

그림 22는 질의 영역 크기에 따라 BPA와 TAG+q-digest, TAG+SMC, IWQE+q-digest, IWQE+SMC에 대해서 MEDIAN 질의의 에너지 소모와 질의 처리 시간의 성능 평가 결과를 보여준다.

그림 22에서 보는 바와 같이 에너지 소모는 BPA가 TAG+q-digest 보다 평균 94%, TAG+SMC 보다 평균 84%, IWQE+q-digest 보다 평균 68%,

IWQE+SMC 보다 평균 56% 성능이 향상되었다. 그리고 질의 처리 시간은 BPA가 IWQE+q-digest 보다 평균 100%, IWQE+SMC 보다 평균 90%, TAG+q-digest 보다 평균 66%, TAG+SMC 보다 평균 54% 성능이 향상되었다.

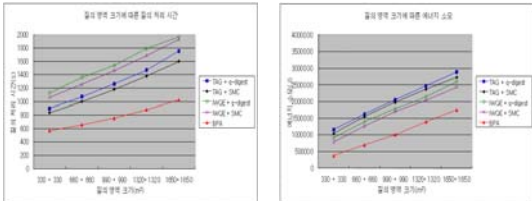


그림 22. 질의 영역 크기에 따른 MEDIAN 집계 연산 성능 평가

4.3 성능 평가 분석

센서 노드 개수, 연속 질의 횟수, 질의 영역 크기에 따른 AVG 질의의 성능 평가에서는 BPA가 TAG, IWQE 보다 항상 처리 결과 정확도, 에너지 소모, 질의 처리 시간에서 우수한 성능을 보여 주었다. 이것은 BPA가 질의 수행시 TAG의 문제점인 질의 영역에 포함되지 않는 센서 노드의 데이터 전송 문제를 해결하고, IWQE의 단점인 라우팅 과정에서 발생할 수 있는 Missing 노드의 발생을 줄였고, 또한 질의 영역을 분할하여 집계 질의를 병렬로 처리하기 때문인 것으로 판단된다.

그리고 센서 노드 개수, 연속 질의 횟수, 질의 영역 크기에 따른 MEDIAN 질의의 성능 평가에서는 BPA가 TAG-q-digest, TAG-SMC, IWQE-q-digest, IWQE-SMC 보다 항상 에너지 소모와 질의 처리 시간에서 우수한 성능 결과를 보여 주었다. 이것은 BPA가 q-digest와 SMC와 같이 범위값이 고정된 데이터 구조를 사용하지 않고 적응적으로 데이터 구조를 갱신하고, 매번 집계된 모든 데이터를 전송하지 않고 변경된 버킷 정보만을 전송하고, 또한 전송 데이터에 대해 압축 및 필터링을 수행하기 때문인 것으로 사료된다.

특히, AVG, MEDIAN 질의 성능 평가 결과에서 센서 노드 개수와 연속 질의 횟수가 많을수록 그리고 질의 영역 크기가 클수록 BPA가 기존 관련 기법들 보다 더욱 우수한 성능 결과를 주었다.

5. 결론

본 논문에서는 무선 센서 네트워크에서 보다 효율적인 집계 질의 처리를 위해 버킷 기반의 병렬 집계 질의 처리 기법인 BPA를 제시하였다. BPA는 센서 노드들의 에너지 소모를 줄이고 질의 처리 시간을 빠르게 하기 위해 질의 영역을 쿼드 트리로 구성하고 쿼드 트리 노드의 각 셀 영역을 Itinerary 라우팅에 따라 집계 질의를 병렬로 처리한다. 그리고 질의 처리 결과 정확도를 높이기 위해 Missing 노드 발생을 최대한 줄이고, 센서 노드들은 데이터를 이중 전송함으로써 전송 오류로 인한 데이터 손실도 줄인다.

BPA는 MEDIAN과 HISTOGRAM 집계 질의 처리시 센서 노드들의 에너지 소모를 줄이기 위해 버킷 기반의 데이터 구조를 이용하고 가변 길이 비트 압축 코딩 기법을 사용한다. 특히, BPA는 질의 처리 결과 정확도를 높이기 위해 버킷 데이터 구조를 버킷내 데이터 개수에 따라 적응적으로 분할 및 합병한다. 또한 BPA는 센서 노드들의 에너지 소모를 줄이기 위해 데이터 전송시 필터링 허용 범위를 벗어난 경우에만 데이터를 전송한다. 마지막으로 센서 데이터를 이용한 다양한 실험을 통해 본 논문에서 제안하는 집계 질의 처리 기법인 BPA의 우수성을 입증하였다.

참고 문헌

[1] J. Li, F. Considine, G. Kollios and J. Byers, 2004, "Approximate Aggregation Techniques for Sensor Databases," Proc. of the ICDE, pp.449 - 460.

[2] D. Culler and D. Estrin, 2004, "Overview of Sensor Networks," Journal of the IEEE Computer Magazine, Vol.37 No.8, pp.41 - 49.

[3] E. Fasolo, M. Rossi, J. Widmer and M. Zorzi, 2007, "In-network Aggregation Techniques for Wireless Sensor Networks: a Survey," Journal of the IEEE Wireless Communications, Vol.14 No.2, pp.77 - 87.

[4] J. J. Kim, H. K. Kang, D. S. Hong and K. J. Han, 2007, "An Efficient Compression Technique for a Multi-dimensional Index in Main Memory," Proc. of the Int. Conf. on Visual

Information Systems, pp.336-346.

- [5] S. Madden, M. Franklin, J. Hellerstein and W. Hong, 2002, "TAG: A Tiny Aggregation Service for Ad-hoc Sensor Networks," Proc. of the Symposium on Operating System Design and Implementation, pp.131 - 146.
- [6] S. Motegi, K. Yoshihara and H. Horiuchi, 2006, "DAG Based In-Network Aggregation for Sensor Network Monitoring," Proc. of the IEEE SAINT, pp.292 - 299.
- [7] D. Pendarakis, N. Shrivastava, L. Zhen and R. Ambrosio, 2007, "Information Aggregation and Optimized Actuation in Sensor Networks," Proc. of the IEEE Int. Conf. on Computer Communications, pp.2386 - 2390.
- [8] S. Roy, M. Conti, S. Setia and S. Jajodia, 2009, "Secure Median Computation in Wireless Sensor Networks," Journal of the Ad Hoc Networks, Vol.7 No.8, pp.1448 - 1462.
- [9] Y. Xu, W. Lee, J. Xu and G. Mitchell, 2006, "Processing Window Queries in Wireless Sensor Networks," Proc. of the IEEE Int. Conf. on Data Engineering, pp.270 - 280.
- [10] 강홍구, 김정준, 한기준, 2007, "데이터 중심 센서 네트워크에서 에너지 효율성을 고려한 비균등 네트워크 분할 기법," 한국공간정보시스템학회 논문지, 제9권 제3호, pp.35 - 50.
- [11] 김정준, 심희정, 강홍구, 이기영, 한기준, 2009, "플래시 메모리 기반 효율적인 공간 인덱스," 한국공간정보시스템학회 논문지, 제11권 제2호, pp. 133-142.

논문접수 : 2011.05.09
 수정일 : 2011.06.24
 심사완료 : 2011.06.27



김 정 준

2003년 건국대학교 컴퓨터공학과(공학사)
 2005년 건국대학교 대학원 컴퓨터공학과(공학석사)
 2010년 건국대학교 대학원 컴퓨터공학

과(공학박사)

2010년~현재 건국대학교 컴퓨터공학과 강의교수
 관심분야는 GIS, LBS, USN, Geo Semantic Web



신 인 수

2006년 건국대학교 컴퓨터공학 공학사
 2008년 건국대학교 대학원 공학석사
 2008년~현재 건국대학교 대학원 컴퓨터공학과 박사과정

관심분야는 공간 데이터베이스, GIS



이 기 영

1984년 숭실대학교 전자계산학과(공학사)
 1988년 건국대학교 대학원 컴퓨터공학과(공학석사)

2005년 건국대학교 대학원 컴퓨터공학과(공학박사)

1996년~1998년 한국컴퓨터정보학회 이사 및 서울동부지회장

1991년~현재 을지대학교 의료IT마케팅학과 교수
 관심분야는 GIS, LBS, USN, 텔레매틱스



한 기 준

1979년 서울대학교 수학교육학과(이학사)
 1981년 한국과학기술원(KAIST) 전산학과(공학석사)

1985년 한국과학기술원(KAIST) 전산학과(공학박사)

1985년~현재 건국대학교 컴퓨터공학과 교수

1990년 Stanford 대학 전산학과 Visiting Scholar

2000년~2002년 한국정보과학회 데이터베이스 연구회 운영위원장

2004년~2006년 한국공간정보시스템학회 회장

2004년~2008년 한국정보시스템감리사협회 회장

관심분야는 공간 데이터베이스, GIS, LBS, 텔레매틱스, 정보시스템 감리