센서 네트워크에서 연속 스카이라인 질의 처리를 위한 상향식 필터링 투플 선정 방법

(A Bottom up Filtering Tuple Selection Method for Continuous Skyline Query Processing in Sensor Networks)

선 진 호 † 정 진 완 ††
(JinHo Sun) (Chin-Wan Chung)

요 약 스카이라인 질의 처리는 센서 네트워크 응용에서 다차원 데이터를 효과적으로 활용할 수 있어서 그 역할이 중요하다. 센서 네트워크는 배터리 제약 사항을 가지고 있기 때문에, 센서 네트워크에서의 스카이라인에 관한 연구는 에너지 소비를 최소화 하는데 그 목표를 두고 있다. 이를 위해 기존연구에서 필터링 기법이 제안되었다. 하지만 기존 필터링 기법은 일회성 질의에 초점을 맞추고 있고, 상위 노드의 정보만을 활용하기 때문에 그 성능의 한계가 있다. 본 논문에서는 연속스카이라인 질의 처리를 위한 상향식 필터링 투플 선정 방법을 제안한다. 하위노드에서 생성된 이전 스카이라인 정보를 각 센서노드에 저장하고, 필터링 투플 선정에 활용함으로써 불필요한 데이터 통신을 감소시킬 수 있다. 이와 더불어 추가 필터링 투플을 선택할 때 사용될 수 있는 SFT(Support Filtering Tuple)방법을 제안한다. 센서 데이터의 경우, 이전 센싱된 데이터와 현재 데이터 간의 시간 관계성(temporal correlation)의 특징을 갖고 있다. SFT 방법은 저장된 과거 데이터를 기반으로 현재데이터를 예측하여 추가 필터링 투플을 선정하여 필터링 성능을 향상시킨다. 실험 결과를 통해, 제안하는 방법들이 기존 방법에 비해 데이터 감소율과 총 통신량 측면에서 효율적임을 보여준다.

키워드 : 센서 네트워크 스카이라인, 연속 질의, 에너지 효율

Abstract Skyline Query processing is important to wireless sensor applications in order to process multi-dimensional data efficiently. Most skyline researches about sensor network focus on minimizing the energy consumption due to the battery powered constraints. In order to reduce energy consumption, Filtering Method is proposed. Most existing researches have assumed a snapshot skyline query processing and do not consider continuous queries and use data generated in ancestor node. In this paper, we propose an energy efficient method called Bottom up filtering tuple selection for continuous skyline query processing. Past skyline data generated in child nodes are stored in each sensor node and is used when choosing filtering tuple. We also extend the algorithms, called Support filtering tuple(SFT) that is used when we choose the additional filtering tuple. There is a temporal correlation between previous sensing data and recent sensing data. Thus, Based on past data, we estimate current data. By considering this point, we reduce the unnecessary communication cost. The experimental results show that our method outperforms the existing methods in terms of both data reduction rate(DRR) and total communication cost.

Key words : Sensor network, Skyline, Continuous query, Energy efficiency

· 본 연구는 방위사업청과 국방과학연구소의 지원으로 수행되었습니다.

† 학생회원 : 한국과학기술원 전산학전공
jhsun@islab.kaist.ac.kr
†† 종신회원 : 한국과학기술원 전산학전공 교수
chungcw@kaist.edu

논문접수 : 2009년 4월 17일
심사완료 : 2009년 5월 12일

정보과학회논문지: 데이타베이스 제36권 제4호(2009.8)

1. 서 론

미래의 유비 쿼터스 환경에서 센서 네트워크는 다양한 센서 정보를 수집, 처리하는 중요한 역할을 수행할 것으로 보인다. 그 응용으로는 기상, 해류, 지진 조사 등의 환경 모니터링에서부터 동식물의 생태계 조사, u-city, 보건관리 등에 이르기까지 광범위하다고 할 수 있다. 이로 인해 최근 Wireless sensor networks(이하 WSN)로부터 효율적으로 데이터를 수집하기 위한 많은 알고리즘 및 시스템이 개발되어왔다.

기본적으로 센서네트워크는 매우 작은 센서 노드를 활용하기 때문에 배터리 사용한도, 메모리 사용량의 제약, 대역폭의 한계, 센서로부터 발생한 데이터의 불확실성 등과 같은 제약사항을 갖고 있다. 센서 노드는 배터리에 의해 작동하므로, 에너지 소비량이 가장 고려되어야 할 사항이고, 에너지 소비의 가장 큰 비중을 차지하고 있는 통신량을 줄이는 것이 가장 큰 목표이다.

이를 위한 대표적인 방법으로 네트워크 내 처리(in-network processing)방법이 존재한다[1-4]. 하지만 기존 연구에서는 MAX, MIN, AVG, SUM, COUNT 등과 같은 단순한 형태의 병합(aggregation) 처리에 초점을 맞추어 연구되어왔다. 그러나, 센서 네트워크를 보다 효과적으로 사용하기 위해서는 조인, 스카이라인 등과 같은 복잡한 형태의 질의에 대한 연구가 필수적이라 할 수 있다.

스카이라인 질의는, 관계형 데이터베이스에서 투플의 집합 T가 주어졌을 때 다른 어느 투플에 대해서도 지배(dominate)되지 않는 값들의 집합을 반환하게 된다. 즉, 집합 T에 두 개의 투플 t_i와 t_j가 존재할 때, 투플 t_i가 모든 차원에 대해서 투플 t_j 보다 나쁘지 않고, 적어도 한 차원에 대해서 투플 t_j 보다 낫다면, 투플 t_i가 투플 t_j를 지배한다고 한다.

이러한 스카이라인 질의는 센서 네트워크에서 상황파악 및 감시의 목적으로 널리 활용될 수 있다. 예를 들어, "대기 오염이 가장 심한 지역을 알려 주시오"와 같은 질의가 존재 할 수 있다. 측정값이 CO와 SO2라 가정할 때, 이러한 다차원 측정값을 기반으로 스카이라인을 처리함으로써 질의 처리가 가능하다[5,6]. 그리고 온도, 습도 및 기압 등을 감지하는 센서를 대상으로 온도가 높고, 습도 및 기압이 낮은 지역에 대한 스카이라인을 적용함으로써 "화재 발생 가능성이 높은 지역을 파악하라" 라는 질의의 수행이 가능하다. 위와 같은 다양한 형태의 스카이라인 질의를 통해, 사용자는 보다 효과적으로 의사 결정을 내릴 수 있게 된다.

센서 네트워크 환경에서는 지속적으로 데이터가 수집되게 되는데, 센서로부터 수집된 무한한 데이터 스트림에 대해 스카이라인 질의를 처리하는 것은 무의미할 뿐만 아니라 불가능하다고 할 수 있다. 그러므로 스카이라인 질의의 대상이 되는 데이터를 슬라이딩 윈도우를 기반으로 한정 시킬 필요가 있다. 또한, 지속적인 환경 감시를 위해서 연속적으로 스카이라인 질의를 처리할 필요가 있다.

기존 스카이라인 질의처리 방법으로는 필터링 방법이 있다[5-7]. 모든 스카이라인 투플을 베이스스테이션으로 전송하는 중앙집중식 질의 처리 방법에서는 스카이라인에 포함되지 않는 불필요한 데이터를 전송하게 된다. 이를 해결하기 위해 필터링 투플을 선정하여 전송함으로써 총 통신량을 감소시키는 방식이 필터링 방법이다. 하지만 기존 필터링 방법은 센서네트워크 기본 환경을 고려하지 않았다. 즉, 상위 노드의 투플로서 하위 노드의 투플을 지배하여 불필요한 전송을 방지한다는 점에서 지역관계성(spatial correlation)을 고려하였지만, 상황판단 및 감시에 있어 연속 스카이라인을 처리할 때 t시간에 감지된 데이터와 t+1 시간에 감지된 데이터 간의 유사성 즉 시간관계성(temporal correlation)이 존재한다는 특성을 고려하지 않았다.

본 논문에서는, 윈도우기반의 연속스카이라인 질의를 처리함에 있어, 시간관계성을 고려한 상향식 필터링 투플 선정 방법을 제안한다. 이와 더불어, 기존에 제안된 필터링 투플을 선정하는 방법을 개선하여 시간관계성을 고려한 추가 필터링 투플을 선정하는 SFT(Support Filtering Tuple) 방법을 제안한다.

본 논문의 공헌은 다음과 같다.

- **상향식 필터링 투플 선정 방법** 센서 네트워크의 기본 특성인 시간관계성을 활용하기 위해, 스카이라인이 베이스스테이션으로 모아지는 과정에서 중간 노드에 이를 임시 저장하여 향후 필터링 투플 선정 시 활용하는 방법을 제안하였다.
- **SFT(Support Filtering Tuple)** 기존에 제안된 필터링 투플 선정 방법의 성능을 향상시키기 위해, 각 노드들은 자식 노드로부터 받았던 이전 스카이라인 정보를 활용하여 기존 하나의 필터링 투플을 전송하는 것과 더불어 추가 필터링 투플을 선정하는 방법을 제안하였다.
- **제안한 방법의 성능 평가** 인위적 랜덤 데이터와 실제 데이터를 이용하여 속성의 차수를 변화시키면서 다양한 실험을 수행함으로써, 본 논문에서 제안한 방법의 효율성을 평가하였다.

본 논문의 구성은 다음과 같다. 먼저 2장에서 기존에 제안된 스카이라인 처리 방법에 대해 살펴보고, 3장에서는 본 논문에서 제안하는 센서네트워크에서의 상향식 필터링 투플 선정을 활용한 스카이라인 처리 방법과 추

가 필터링 투플의 선정에 있어서 성능향상을 위한 SFT(Support filtering tuple) 방법을 소개한다. 4장에서는 제안한 방법에 대해 실험을 통해 우수성을 평가한다. 최종적으로 5장에서 결론을 맺는다.

2. 관련 연구

스카이라인에 관한 연구는 데이터베이스 분야에서 과거부터 현재까지 활발히 진행되어 왔다. Borzonyi[8]은 Block Nested Loop(BNL)와 Divide-and-Conquer 알고리즘을 통한 스카이라인 처리를 제안하였다. Chomicki[9]은 BNL의 변형인 Sort-Filter-Skyline 알고리즘 제안하였다. 하지만 이러한 다양한 스카이라인 방법은 센서 네트워크 환경에 바로 적용할 수 없는 중앙집중식 질의 처리 방법들이다. 즉 모든 데이터를 베이스스테이션으로 전송한 후 스카이라인 처리를 수행하게 되는데, 이는 많은 불필요한 데이터를 전송하게 됨으로써 센서 네트워크의 에너지 효율성을 떨어뜨리게 된다.

이러한 문제점을 해결하기 위해 제안된 방법이 필터링 방법이다. Manet 환경에서의 스카이라인 처리에 관한 연구논문[5]에서는 필터링 방법과 필터링 투플 선정 기준을 제안하였다. 필터링 방법이란 지배영역에 기반하여 특정 투플을 필터링 투플로 선정하여 전송함으로써 불필요한 데이터의 전송을 방지하는 방법이다. 네트워크 내 처리 기법으로 스카이라인을 처리할 때 최종스카이라인에 포함되지 않는 많은 투플들이 여러 노드 간에 통신하게 되어 센서네트워크의 수명이 감소되게 된다. 필터링 방법에서는 상위 노드에서 해당 로컬 스카이라인을 처리한 후, 하위 노드의 로컬 스카이라인의 많은 투플을 지배(dominating)할 가능성이 큰 투플을 하위 노드로 전송한다. 즉 필터링 투플로써 최종스카이라인에 포함되지 않는 하위 노드의 투플들을 미리 제거하여 상위 노드로 전송하지 않게 함으로써 총 통신량을 감소시킨다. 또한, 필터링 투플 선정기준으로 VDR(Volume of Dominating Region)이라는 개념을 도입하였다. 필터링 투플은 되도록 많은 하위 노드의 투플을 지배하여 상위 노드로 전송되지 않도록 하는 것에 그 목적이 있다. VDR은 데이터의 각 속성의 곱으로 지배영역의 크기를 결정한다. 지배영역의 크기가 클수록 더 많은 투플들이 해당 지배영역 안에 포함될 가능성이 높기 때문에 가장 큰 지배영역을 갖는 투플을 필터링 투플로 선정한다.

하지만, 이는 데이터의 차수가 높을 경우 필터링 성능이 좋지 않다. [6]에서는 이러한 VDR의 성능을 향상시키기 위해 MFT(Min Filtering Tuple) 방법을 제안하였다. 각 속성의 값에 역수를 취하여 모두 더한 후, 가장 작은 값을 가진 투플을 필터링 투플로 선정한다. MFT방법은 속성의 차수가 4이상인 데이터에 대하여 VDR에 비해 필터링 성능의 우수성을 보였다.

[7]은 필터링 방법에 기반한 분산환경에서의 스카이라인 병렬 처리에 관한 연구를 수행하였다. 이 때 추가적인 필터링 투플을 전송하는 방법을 제안하였다. 이는 단일 필터링 투플을 전송할 때와 비교하여 더 많은 투플을 필터링할 수 있어 통신량을 감소시킬 수 있었다. 여러 필터링 투플 후보 중 추가적인 필터링 투플을 선정할 때 휴리스틱 방법인 Maximal Sum방법과 Maximal distance 방법을 소개하였다. Maximal Sum방법에서는 필터링 투플 후보들의 VDR의 합을 구하여 가장 큰 합을 가진 투플의 조합을 기반으로 추가 필터링 투플을 선정한다. Maximal distance 방법에서는 필터링 투플들 간의 유클리디안 거리가 가장 큰 투플의 조합을 기반으로 추가 필터링 투플을 선정한다. 기존 필터링 투플과 비슷한 값을 가진 투플을 추가 필터링 투플로 전송할 때, 기존 필터링 투플과 많은 지배영역이 중복되어 필터링 성능 향상에 도움이 되지 않는다는 점에 초점을 맞춘 방법이다.

하지만 [5-7] 방법들은 일회성 스카이라인 질의 처리에 초점을 맞추었기 때문에, 센서 네트워크의 상황판단 및 감시를 위한 연속스카이라인 처리에 그대로 적용하기 적절치 않다. 즉, 이전 데이터 측정값과 현재 데이터의 측정값과의 시간관계성이 존재하는 특성을 반영하지 못한다. 또한 [7]에서의 추가 필터링 투플 선정방법은 필터링 투플 후보들의 모든 조합을 고려하여 계산량이 많아지기 때문에 처리능력의 제약이 있는 센서네트워크 환경에 그대로 적용하는 것이 불가능하다.

기존 제안된 필터링 투플 선정 방법들은 모든 데이터가 균등하게 분포하였을 때를 가정한다. 하지만, 실제 센서네트워크 데이터는 균등하게 분포하지 않기 때문에 기존 방법을 그대로 적용할 경우 성능이 좋지 않다. 본 논문에서는 [5,6]에서 제안한 필터링 기법의 단점을 보완하고, [7]의 방법에 착안하여 센서네트워크 상황에 적합한 추가 필터링 투플 선정방법을 제안한다.

3. 상향식 필터링 투플 선정 방법을 활용한 연속 스카이라인 처리

3.1 문제 정의

본 논문에서는 m개의 센서 노드 $S = \{S_1, S_2, \cdots, S_m\}$가 존재하는 센서네트워크 환경을 가정한다. 센서 노드의 통신은 방송(broadcasting)방식으로 전송하며, 동일한 크기의 메시지는 동일한 에너지를 소비하게 된다.

센서 노드 S_i는 n개의 센서를 갖고 있어 매 센싱 주기마다 $P_i = \{p_1, p_2, \cdots p_n\}$의 n차원 데이터를 생성하게 되고, S_i에서 t 시간에 생성된 데이터를 $P_{i,t}$로 나타낸다.

각 센서 노드는 w개의 윈도우를 갖고 있다고 가정할

때, t-w+1시간에 센싱된 데이터부터 가장 최신 데이터인 t시간에 센싱된 데이터까지가 윈도우를 구성하게 된다. 즉 Si의 경우 {$P_{i,t-w+1}$, $P_{i,t-w+2}$, … $P_{i,t}$}의 데이터를 윈도우에 보관하게 되고, 이를 기반으로 스카이라인 질의를 처리하게 된다.

이를 기반으로 연속스카이라인을 그림 1처럼 처리하게 된다.

그림 1에서는 윈도우의 크기가 10이고, 스카이라인 처리는 매 2 시간간격마다 수행하게 됨을 나타낸다.

센서 노드 S_i에서의 연속스카이라인 처리의 진행과정은 다음과 같다.

T10(time 10)에서는 time 1부터 time 10까지 센싱된 데이터 즉, {$P_{i,1}$, $P_{i,2}$, … $P_{i,10}$}의 투플들을 기준으로 스카이라인을 처리한다. 그 후 T12(time 12)가 되었을 때는 두 개의 새로운 데이터 $P_{i,11}$과 $P_{i,12}$가 새롭게 센싱되어 추가 되어 있고, $P_{i,1}$과 $P_{i,2}$는 윈도우 보관한계를 벗어나기 때문에 삭제된다. T12에서는 {$P_{i,3}$, $P_{i,4}$, … $P_{i,12}$}의 투플들을 대상으로 스카이라인을 처리하게 된다.

이러한 형태로 연속적인 스카이라인을 처리할 때, 중복된 데이터 영역이 존재하기 때문에, 이전 스카이라인 처리에서 스카이라인으로 지정된 투플들을 활용한다면, 보다 효율적인 처리가 가능하다.

즉, T10에서 처리된 스카이라인과 T12에서 처리된 스카이라인은 일부 중복된 데이터를 대상으로 처리된다. 또한 센서 데이터의 경우 시간이 흐름에 따라 급격하게 변하지 않고 전체적으로 원만한 변화를 이루기 때문에, T10에서의 스카이라인과 T12에서의 스카이라인은 큰 차이가 없게 된다.

베이스 스테이션이 스카이라인 질의 처리를 위해 모든 센서 노드에게 스카이라인 질의를 전송하면, 각 센서 노드 S_i는 자신의 메모리인 윈도우에 저장된 데이터를 대상으로 스카이라인(LSK_i)를 구하고, 그 결과를 베이스 스테이션으로 멀티홉을 거쳐 전송하게 된다. 이러한 과정을 그림 1에서처럼 T10, T12, T14, T16 시점에 연속적으로 처리함으로써 스카이라인을 처리하게 된다.

표 1 본 논문에서 사용된 기호

기호	의미
S_i	i 번째 센서 노드
p_j	센서 노드 내 j번째 센서가 센싱한 데이터
$P_{i,t}$	t 시간에 i번째 센서가 센싱한 튜플
$LSK_{i,t}$	t 시간에 i번째 센서 노드의 스카이라인 결과
$LSK2_{i,t}$	t 시간에 i번째 센서 노드에서 병합된 스카이라인 결과
$a_{i,t,k}$	$LSK2_{i,t}$ 내 존재하는 k번째 튜플
tp_i	i번째 센서 노드의 필터링 튜플

표 1은 본 논문에서 사용된 기호이다.

3.2 상향식 필터링 투플 선정 방법

기존 연구인 MANET[5]에서는 필터링 방법을 제안하였다. [5]에서의 "필터링 방법의 애드혹 네트워크에의 적용"은 센서네트워크에서의 네트워크 내 처리 방법(In-Network Processing)에 그대로 대응될 수 있는 방법이다. 이 방법은 필터링 투플 전송 단계와 스카이라인 결과 수집단계로 나눌 수 있다.

해당 센서 노드 S_i는 t시간의 로컬 스카이라인 $LSK_{i,t}$를 구한 후, $LSK_{i,t}$ 중 한 투플을 필터링 투플tp_i로 선정하고 자식노드인 S_j에게 전송한다. 자식노드 S_j 또한 $LSK_{j,t}$를 구하게 되고, $LSK_{j,t}$ 와 tp_i로부터 필터링 투플 tp_j를 선정한다. 이 과정이 센서 네트워크 트리의 모든 노드에 반복하게 되는 필터링 투플 전송 단계이다. 이 과정이 종료되면, 해당 노드S_i는 그의 자식노드 S_j로부터 스카이라인을 전송 받게 된다. 이를 자신의 로컬 스카이라인 $LSK_{i,t}$과 병합한 후 지배되는 투플을 제거하고 부모 노드S_k에게 전송할 $LSK2_{i,t}$를 생성한다. 부모 노드 S_k 또한 이러한 과정을 반복하여 베이스스테이션으로 스카이라인 결과를 전송한다. 이 과정을 스카이라인 결과 수집 단계라 한다.

본 논문에서 스카이라인 처리의 기본은 MANET에서 제안한 방법을 그대로 활용한다. MANET에서 제안한 방법은 기본적으로 상위노드에서 하위노드로 전송된 정

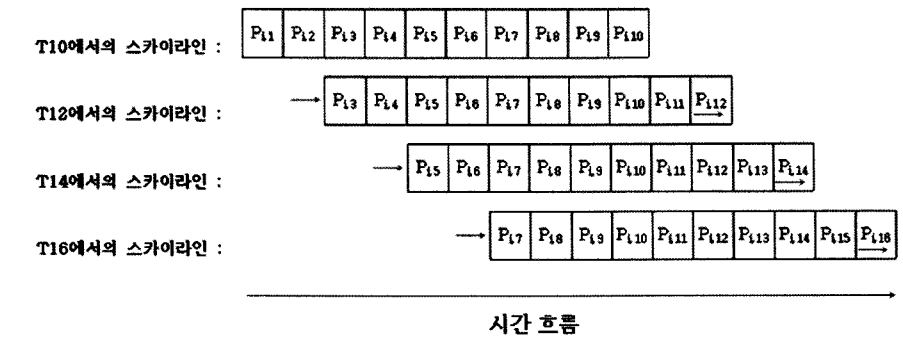

그림 1 연속스카이라인 처리

보를 기반으로 필터링 투플을 선정하는 하향식 필터링 투플 선정 방법이었다. 본 논문에서는 상위 노드의 정보를 활용하는 하향식 필터링 투플 선정 방법에 추가하여, 하위 노드의 정보 또한 활용하는 상향식 필터링 투플 선정 방법을 제안한다.

3.2.1 연속스카이라인 처리에서 필터링 투플 선정 단계

본 논문에서 제안한 상향식 필터링 투플 선정방법은 다음과 같다.

첫 스카이라인 질의는 기존 MANET에서 제안한 방법으로 처리하게 된다. 단, 본 논문에서는 각각의 센서노드가 자식노드로부터 전송된 스카이라인을 해당 노드에 CHILD_SKYLINE이라는 버퍼에 저장하는 방식을 취한다.

$a_{i,t,1}$	$a_{i,t,1}$ 의 유효시간	$a_{i,t,2}$	$a_{i,t,2}$ 의 유효시간	...	...	$a_{i,t,k}$	$a_{i,t,k}$ 의 유효시간

그림 2 전송할 스카이라인 형식

즉, 전송할 스카이라인에는 그림 2와 같이 유효시간을 포함하여 전송한다. 유효시간이란 해당 투플이 스카이라인에 포함될 수 있는 남은 시간을 의미한다. 예를 들면, 윈도우 내에서 유효시간이 7인 투플의 경우, 앞으로 7번의 센싱시간 안에 수행되는 스카이라인의 처리 대상 투플로 지정 될 수 있고 다른 투플을 지배할 수 있다. 즉, 앞으로 7번의 센싱시간 동안 해당 투플이 유효하다고 할 수 있는데, 이러한 남은 시간을 유효시간이라 한다.

두 번째 이 후의 스카이라인 질의부터는 다음 방식을 따른다.

부모 노드 S_k는 t시간에 자신의 스카이라인 $LSK_{k,t}$를 구하고 필터링 투플 선정 단계를 수행한다. 이 때, 이전 스카이라인 처리 시 자식노드 S_i로부터 전송 받았던 정보가 담긴CHILD_SKYLINE 버퍼를 고려한다. t시간에 스카이라인의 대상이 되는 투플들은 t+1시간에 스카이라인의 대상이 되는 투플들과 일부 중복된다. 그러므로 t시간에 스카이라인 처리 시, 이전에 수행되었던 스카이라인 정보를 활용할 수 있게 된다. 그림 1에서 설명했듯이 해당 시간의 스카이라인 윈도우 일부에 대해, CHILD_SKYLINE에는 자식 노드들의 스카이라인 정보가 담겨 있다. CHILD_SKYLINE에 포함된 정보를 이용하면, t시간에 스카이라인을 처리할 때는 이전에 수행된 스카이라인의 기간 이 후에 센싱된 투플을 추가적으로 고려하면 된다. 그림 1의 예를 보자. T12에 스카이라인을 구할 때 T2에서T10까지 센싱된 투플에 대해서는 CHILD_SKYLINE에 포함된 정보를 활용하여 필터링 하게 된다.

새롭게 스카이라인 처리 대상으로 추가된 투플들에 대해서도 시간관계성의 성질에 따라, CHILD_SKYLINE의 정보를 통해 보다 많은 불필요한 스카이라인 투플을 제거 가능하게 된다. 그러므로 필터링 투플 tp_i를 선정할 때, $LSK_{i,t}$와 CHILD_SKYLINE을 병합한 스카이라인을 대상으로 VDR(Volume of Dominating Region) 또는 MFT 방법을 이용하여 필터링 투플을 선정한다.

즉, 각 센서노드가 자신을 포함한 모든 자식 노드들의 이전 스카이라인에 포함되었던 투플을 활용하여 필터링 투플을 선정하게 된다. 그 후 이 투플로 인해 지배될 수 있는 새롭게 센싱된 투플이 존재하게 될 때 이를 전송하지 않을 수 있게 되어 총 통신량을 감소시킬 수 있게 된다.

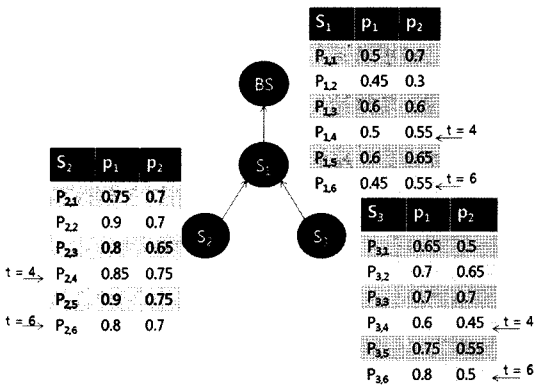

그림 3 상향식 필터링 투플 선정 방법

그림 3은 센서 노드 3개로 이루어진 센서네트워크 구성이다. 각 노드의 윈도우크기는 4이고, 더 큰 데이터 값이 작은 데이터 값을 지배한다고 가정한다. 또한 p_1과 p_2는 센서 데이터의 속성으로서, 각 투플의 차수는 2라 가정한다. t가 4일 때, S_1의 로컬 스카이라인 대상이 되는 투플은 $\{P_{1,1}, P_{1,2}, P_{1,3}, P_{1,4}\}$이고, S_i의 로컬 스카이라인$LSK_{1,4}$는 $\{P_{1,1}, P_{1,3}\}$이다. VDR기법으로 필터링 투플을 선정한다면, $p_{1,1}$은 0.5*0.7이고 $P_{1,3}$은 0.6*0.6이므로 $P_{1,3}$이 필터링 투플이 된다. 이를 S_2와 S_3로 전송한다. S_2에서 $LSK_{2,4}$는 $\{P_{2,2}, P_{2,4}\}$이고, S_3에서 $LSK_{3,4}$는 $\{P_{3,3}\}$이다. 이 때 S_1으로부터 전송된 필터링 투플 $P_{1,3}$은 어느 투플도 필터링 하지 못한다. 그러므로 S_1으로 로컬 스카이라인 $\{P_{2,2}, P_{2,4}\}$, $\{P_{3,3}\}$가 수집된다. 이 때 S_1은 자신의 CHILD_SKYLINE버퍼에 자식 노드들로부터 받은 $\{P_{2,2}, P_{2,4}, P_{3,3}\}$을 저장한다. 그리고 S_2, S_3로부터 받은 스카이라인을 병합할 때를 보면, $\{P_{2,2}, P_{2,4}\}$에 의해 $\{P_{1,1}, P_{1,3}\}$과 $\{P_{3,3}\}$이 지배되므로 $\{P_{2,2}, P_{2,4}\}$만을 베이스스테이션으로 전송한다. 이러한 기반 하에, t가 6일 때 연속 스카이라인을 처리한다. 윈도우가 4이므로 각 센서에서는 t=2부터 t=6의 데이터를 대상으로 스카이라인을

처리한다. S_1에서는 {$P_{1,3}$, $P_{1,4}$, $P_{1,5}$, $P_{1,6}$} 중 {$P_{1,5}$}가, S_2에서는 {$P_{2,5}$}가, S_3에서는 {$P_{3,3}$, $P_{3,5}$, $P_{3,6}$}가 로컬 스카이라인으로 처리된다. S_1에서 필터링 투플을 선정시 CHILD_SKYLINE의 데이터도 고려하게 된다. 그러므로 S_1에서는 VDR이 가장 큰 {$P_{2,4}$}를 필터링 투플로 선정하여 자식노드인 S_2와 S_3에게 전송하게 된다. 만약, 필터링 투플 선정 시 CHILD_SKYLINE의 정보를 활용하지 않는다면, {$P_{1,5}$}가 필터링 투플로 선정되어 S_3로 전송되게 되고, S_1으로 전송해야 할 S_3의 로컬 스카이라인인 {$P_{3,3}$, $P_{3,5}$, $P_{3,6}$}을 필터링하지 못한다. 하지만, 본 논문에서 제안한 방법을 이용하면 {$P_{2,4}$}={0.85, 0.75}가 필터링 투플로 전송되어, {$P_{3,3}$, $P_{3,5}$, $P_{3,6}$}을 지배한다. 그러므로 S_3는 아무런 데이터를 전송하지 않게 되어 통신량을 감소시킬 수 있다. S_2에서는 S_1으로부터 {$P_{2,4}$}를 필터링 투플로 전송받게 된다. 하지만, {$P_{2,4}$}는 S_2의 로컬 스카이라인인 {$P_{2,5}$}를 지배하지 못하므로, S_2는 {$P_{2,5}$}를 S_1으로 전송하게 된다. S_1은 이를 병합하여 최종 스카이라인 결과를 베이스스테이션으로 전송한다.

3.2.2 연속스카이라인 처리에서 스카이라인 결과 수집 단계

스카이라인 결과 수집단계에서는, 스카이라인 데이터를 베이스 스테이션으로 전송한다. t타임에 자식 노드 S_j는 자신의 스카이라인 $LSK_{j,t}$를 해당 노드 S_i에게 전송하고, 해당 노드는 자식 노드로부터 전송받은 $LSK_{j,t}$를 CHILD_SKYLINE 메모리 버퍼에 이를 등록한다. 이때, t-1타임에 전송받은 $LSK_{j,t-1}$의 데이터가 CHILD_SKYLINE에 등록되어 있게 되는데, 다음과 같은 기준으로 CHILD_SKYLINE 메모리 버퍼를 갱신하게 된다.

P_j = 새롭게 전송받은 $LSK_{i,t}$에 포함된 투플
C_i = CHILD_SKYLINE 내 투플
If C_i.유효시간 < P_j.유효시간 & P_j dominate C_i
Then Remove C_i from CHILD_SKYLINE

즉, 새롭게 전송받은 스카이라인 데이터가 CHILD_SKYLINE의 투플을 지배하고 유효시간이 더 길다면, CHILD_SKYLINE의 투플은 더 이상 스카이라인에 포함될 수 없으므로 제거하게 된다. P_j가 C_i를 지배하고 유효기간이 더 짧은 경우엔, C_i를 제거할 수 없다. C_i의 유효기간이 더 길기 때문에 P_j의 유효기간이 만료된 후에도 C_i 는 스카이라인에 포함될 수 있기 때문이다. 만약 제거하였을 경우, C_i가 스카이라인에 포함될 가능성이 존재함에도 불구하고 P_j에 의해 이미 제거 되었으므로 C_i가 스카이라인에 포함되지 못하게 된다.

CHILD_SKYLINE 메모리 버퍼를 갱신한 후, 해당 노드 S_i는 베이스 스테이션으로 전송할 스카이라인 $LSK2_{i,t}$를 생성해야 한다. 해당 노드는 자신이 센싱한 투플들을 대상으로 $LSK_{i,t}$를 생성한 후, 자식노드의 스카이라인과 병합하여 스카이라인 $LSK2_{i,t}$을 생성함으로써 센서네트워크 내 처리를 수행한다.

이러한 형태로 각 센서 노드에서 스카이라인을 처리한 후, 새로운 센싱 주기가 도래할 때 마다 CHILD_SKYLINE에 포함된 투플들의 유효시간을 -1씩 감소시킴으로써 올바른 스카이라인 처리를 보장하게 된다. 만약 유효시간이 0이 된다면, 더 이상 스카이라인 처리 대상이 되는 윈도우에 포함되지 않게 되므로 해당 투플을 CHILD_SKYLINE에서 제거하게 된다.

3.3 SFT(Support Filtering Tuple)

MANET[5]에서 제안된 필터링 방법은 항상 하나의 필터링 투플을 선정하여 필터링을 수행하게 된다. 그러나 필터링 투플을 추가로 선정함으로써 자식 노드들로부터 더 많은 스카이라인 투플을 제거할 수 있다.

[7]에서는 이러한 문제점을 보완하기 위해 여러 필터링 투플을 선정하는 방법을 제안하였다. 하지만 이는 센서 노드에서 계산하기에는 너무 복잡한 형태를 띄고 있다.

이 장에서는 센서 데이터의 시간관계성과 멀티홉을 거쳐 통신하는 센서 네트워크의 특성을 고려함으로써, VDR과 MFT필터링 투플 선정 기준을 개선한 SFT를 제안한다.

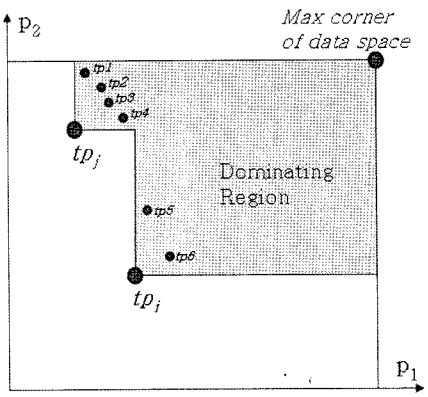

그림 4 여러 SFT 후보

VDR의 경우 추가적인 필터링 투플 선정시 큰 지배영역을 차지하는 투플 순서대로 우선순위가 정해진다. 센서 네트워크에서 연속스카이라인 질의는 시간관계성을 갖고 있기 때문에 이전 스카이라인 처리에서 전송된 투플들과 비슷한 투플들이 지속적으로 스카이라인으로 처리되어 전송될 가능성이 크다. 그러므로, 특정 센서 노드의 CHILD_SKYLINE에 저장된 투플 정보를 이용한다면 필터링 성능을 더욱 향상시킬 수 있다.

SFT는 추가적인 필터링 투플을 선정 할 때, 기본적으로 센서 데이터 분포에 대한 예측에 기반을 두고 있다. 그림 4의 경우를 보자. {tp1, tp2, tp3, tp4, tp5, tp6}는 자식노드로부터 전송 받은 CHILD_SKYLINE에 존재하는 투플이고, tp_i와 tp_j은 추가 필터링 투플의 후보들이다. VDR방법을 활용했을 때 VDR(tp_i) > VDR(tp_j)이므로 tp_i가 SFT로 선정되게 된다. 하지만, tp_j는 {tp1, tp2, tp3, tp4}, 총 4개의 자식노드로부터 전송되는 투플을 지배하게 되는 반면, tp_i는 {tp5, tp6}, 총 2개의 투플을 지배하게 된다. 물론, CHILD_SKYLINE에 포함된 투플들은 새롭게 센싱되는 데이터와 상관이 없지만, 이전 정보(history)를 기반으로 앞으로 센싱될 데이터의 분포를 미리 예측할 수는 있는 것이다.

CHILD_SKYLINE에 포함된 투플들은 각각 다른 센서 노드로부터 전송된 스카이라인임에 주목하자. 자신으로부터 6홉이 떨어진 자식노드로부터 전송받은 투플도 존재할 수 있고, 바로 한 홉 떨어진 자식노드로부터 전송받은 투플도 존재할 수 있다. 이 경우, 한 홉 떨어진 자식노드의 투플을 필터링하는 것보다 6홉 떨어진 자식노드의 투플을 필터링하는 것이 총 5번의 통신량을 감소시킬 수 있으리라고 예측 가능하다. 즉, CHILD_SKYLINE 내 투플들이 해당 노드로부터 몇 홉이나 떨어져 있는지에 따라 가중치를 두어 다음과 같이 SFT를 선정한다.

$\sum i$의 가중 치, $i \in tp_i$에 의해 지배되는 튜플의 집합 (1)

각각의 SFT 후보들 중 그들이 지배하는 투플들의 가중치의 합에 따라, 그 합이 가장 큰 투플을 SFT로 선정한다.

예를 들어, 그림 4에서 {tp1, tp2, tp3, tp4, tp5, tp6}의 가중치가 {6,5,4,7,8,3}이라고 했을 때, tp_i의 가중치 합은 8+3=11이고, tp_j의 가중치 합은 6+5+4+7=22이므로, tp_j가 SFT로 선정된다. tp_i를 SFT로 선정했을 때 11의 전송비용을 절약할 수 있다고 예측할 수 있는 반면, tp_j는 필터링을 통해 총 22의 전송비용을 절약할 수 있다고 예측하는 것이다.

가중치가 {4,2,1,3,5,7}일 경우에는 tp_i의 가중치 합은 5+7=12이고, tp_j의 가중치 합은 4+2+1+3=10이기 때문에 tp_i가 더 많은 투플을 필터링 할 것으로 예측됨에도 불구하고, tp_i가 전체적으로 더 많은 통신량을 감소시킬 수 있으리라 예측할 수 있기 때문에 tp_i를 SFT로 선정한다.

즉, VDR을 이용하여 추가적인 필터링 투플을 선정하는 것보다, SFT방법을 활용하는 것이 전체적인 통신비용 감소에 효과적일 수 있다.

그림 5는 SFT방법을 적용한 상향식 필터링 투플 선정 알고리즘을 나타낸다. 해당 노드는 자신의 로컬 스카

```
Algorithm  Bottom Up Filtering tuple Method with Support Filtering Tuple in Si
01:  Receive Fitlering Tuple tpk from parent node Sk
02:  Get local skyline LSKi,t of Si
     // filter(a, b) : return tuples in b not to be dominated by tuples in a
03:  candidate = filter(tpk, LSKi,t)
04:  add tpk to candidate
05:  temp = filter(candidate, CHILD_SKYLINE)
06:  add temp to candidate
     //find filtering tuple tpi by VDR or MFT method
07:  if (VDR method is used)
08:      tpi = find a tuple that has maximum Π pj of tuple in candidate
09:  else if (MFT method is used)
10:      tpi = find a tuple that has maximum Σ pj of tuple in candidate
11:  delete tpi from candidate
12:  for each tuple c in candidate //find the support filtering tuple
13:      for each tuple Ci in CHILD_SKYLINE
14:          if (c dominate Ci)
                 //weight = depth level of Si - depth level of node which generated Ci
15:              valuec + = Ci's weight
16:          end if
17:      end for
18:  end for
19:  SFT = find a tuple which has maximum value
20:  send tpi and SFT to child node
```

그림 5 SFT방법을 적용한 상향식 필터링 투플 선정 알고리즘

```
Algorithm  Collect skyline tuple in Sensor node Si
01:  for each tuple Ci in CHILD_SKYLINE
02:     Ci.VaildTime = Ci.VaildTime - 1
03:     if(Ci.ValidTime == 0)
04:        remove Ci from CHILD_SKYLINE
05:     end if
06:  end for
07:  for each child node Sk of Si
08:     receive local skyline LSKk,t from child node Sk
09:     for each tuple Ci in CHILD_SKYLINE
10:        for each tuple tp in LSKk,t
11:           if (Ci.ValidTime < tp.ValidTime) and (tp dominate Ci)
12:              remove Ci from CHILD_SKYLINE
13:           end if
14:        end for
15:     end for
16:     ADD LSKk,t to temp
17:  end for
18:  temp = Get Skyline for temp
19:  add temp to CHILD_SKYLINE
20:  get local skyline LSKi,t of Si
21:  LSK2i,t = get skyline for temp and LSKi,t
22:  send LSK2i,t to parent node of Si
```

그림 6 스카이라인 투플 수집 단계 알고리즘

이라인을 구한다. 부모 노드 S_k로부터 전송 받은 필터링 투플 tp_k로 로컬 스카이라인을 필터링하여 자신의 필터링 투플이 될 수 있는 후보군(candidate)을 선정한다. tp_k 또한 해당 노드의 필터링 투플이 될 수 있기 때문에 후보군에 포함시킨다. 하위 노드로부터 전송받은 이전 스카이라인 정보(CHILD_SKYLINE) 중 필터링 투플 후보가 될 수 있는 투플을 걸러내고 이들 또한 후보군에 포함시킨다. 이렇게 수집된 후보군을 VDR 또는 MFT 방법으로 가장 큰 값을 갖는 투플을 해당 노드의 필터링 투플 tp_i정한다. tp_i는 필터링 투플로 선정되었으므로 후보군에서 삭제한다. 그 후, 추가 필터링 투플을 선정하게 된다. 후보군에 포함된 각 투플들에 대해 CHILD_SKYLINE에 포함된 투플들을 지배하는 정도를 계산하게 된다. 이 때 거리에 기반한 가중치를 고려한다. 그 후 가장 큰 가중치를 갖는 투플을 SFT로 선정하게 되고, 필터링 투플로 선정된 tp_i와 SFT를 자식노드에 전송한다.

그림 6은 스카이라인 투플 수집 단계 알고리즘을 나타낸다. 먼저 하위 노드로부터 전송받은 이전 스카이라인인 CHILD_SKYLINE의 유효시간을 검사한다. 각 투플과 함께 저장된 유효시간을 감소시키고, 유효시간이 종료된 투플을 CHILD_SKYLINE으로부터 제거한다. 해당 노드는 자식 노드 S_K로부터 자식 노드의 스카이라인을 전송 받고, 유효시간과 지배여부를 기준으로 판단하여 CHILD_SKYLINE의 투플 중 제거할 대상을 찾는다. 자식 노드로부터 받은 로컬 스카이라인들을 병합하여 스카이라인을 계산하고 CHILD_SKYLINE을 갱신한다. 이로써 하위노드로부터 전송 받은 이전 스카이라인 정보를 유지하게 된다. 그 후 부모 노드로 전송할 스카이라인을 계산하기 위해, 자식 노드로부터 전송받은 스카이라인인 temp와 해당 노드의 스카이라인 $LSK_{i,t}$을 병합하여 스카이라인을 구한 후 부모 노드로 전송한다.

4. 실험 및 분석

본 논문에서 제안한 방법의 효율성을 입증하기 위해 시뮬레이션을 통하여 실험을 수행하였다. 시뮬레이션은 JAVA 언어로 구현하였으며, 데이터 도메인 내에서 랜덤하게 데이터를 생성한 랜덤 데이터와 실제 센서 데이터를 실험데이터로 활용하였다.

실험에서 사용한 파라미터는 표 2와 같다.

각 센서는 100개의 윈도우 버퍼로서 센싱한 데이터를 저장하고 매 1초마다 한번 씩 센싱을 수행하면서, 매 20번의 센싱이 수행된 후에 스카이라인 쿼리를 수행하였다.

랜덤데이터에서는 센서 노드 100개로 센서 네트워크를 구성하였고, 데이터 도메인 [0, 1000]의 범위 내에서 랜덤하게 변화를 주었다. 또한 데이터 속성 차수는 2에

표 2 실험데이터에 사용한 파라미터

파라미터	내용
센서 네트워크 노드 수	100 (개)
노드의 윈도우 크기	100 (데이터=100sec)
스카이라인 수행 주기	20 (sec)
센싱 주기	1 (sec)
데이터 도메인	[0, 1000]
데이터 분포	랜덤데이터, 실제데이터
데이터 속성 차수	2, 3, 4, 5

서 5까지 변화를 주었다. 실제데이터의 경우, 인텔 버클리 연구실에서 제공한 54개 센서노드의 데이터[10]를 활용하였고, 데이터 속성 차수는 2와 3으로 설정하여 실험하였다.

상향식 필터링 투플 선정 방법(BUF)은 MANET[5]에서 제안한 필터링 방법을 연속질의 처리로 확장하여 비교대상으로 하였고, 필터링 선정 기준으로는 VDR과 MFT방법을 활용하였다. 본 논문에서 제안한 추가 필터링 투플을 선정하는 방법인 SFT는, MFT를 기준으로 추가 필터링 투플을 선정하는 방법과 비교하였다.

실험은 다음과 같은 여섯 가지 방법에 대해 수행하였다.

1. **MANET**: MANET[5]의 필터링 방법 적용, 필터링 투플 선정 기준은 VDR방법
2. **MFT**: MANET[5]의 필터링 방법 적용, 단 필터링 투플 선정 기준은 MFT[6]방법
3. **MFT+AF(Additional Filtering Tuple)**: 2와 동일, 하나의 필터링 투플 전송과 더불어 MFT를 기반으로 추가 필터링 투플 전송
4. **BUF(Bottom Up Filtering)**: 본 논문에서 제안한 BUF방법 적용, 필터링 투플 선정 기준은 MFT방법
5. **BUF(Bottom Up Filtering)+AF(Additional Filtering Tuple)**: 4와 동일, 추가 필터링 투플 선정 기준은 MFT방법
6. **BUF(Bottom Up Filtering)+SF(Support Filtering Tuple)**: 4와 동일, 추가 필터링 투플 선정 기준은 본 논문에서 제안한 SFT방법

성능 평가 척도로는 Data Reduction Rate(DRR)[3]과 총 통신 데이터량을 사용하였다.

$$DRP = \frac{\sum_{i=1}^{n} |SL_i| - |SL_i'|)}{\sum_{i=1}^{n} |SL_i|} \quad (2)$$

위 수식에서 SL_i는 센서 노드 S_i가 보내야 할 스카이라인 데이터 수이고, SL_i'는 필터링을 수행한 후 실제 통신한 스카이라인 데이터 수로서, DRR을 통해 필터링 후 불필요한 데이터가 얼마나 제거되었는지 확인할 수 있다.

4.1 DRR

그림 7, 8, 9, 10은 랜덤 데이터와 실제 데이터에 대

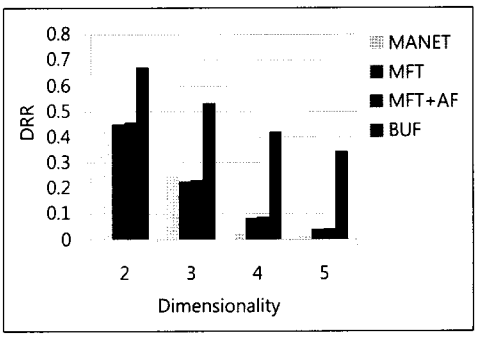

그림 7 랜덤 데이터에 대한 BUF방법의 DRR 결과

한 속성 차수에 따른 DRR의 변화를 보여준다.

그림 7의 랜덤 데이터에 대한 DRR 결과에서 볼 수 있듯이, MANET방법의 경우 MFT방법에 비해 속성의 차수가 3일 때 성능이 우수하지만, 속성의 차수가 4와 5일 때는 그 성능이 급격히 낮아짐을 확인 할 수 있다. 추가 필터링 투플을 전송하는 방법인 MFT+AF는 별다른 성능향상이 없었다.

그러나 BUF방법은 MANET의 방법과 비교할 때, 적게는 속성의 차수가 2일 때 약 0.22정도, 많게는 속성의 차수가 5일 때 0.3정도 DRR이 높게 나타났다. 즉, MANET, MFT 두 방법 모두 본 논문에서 제안한 BUF 방법보다 모든 차수에서 성능이 낮음을 확인할 수 있다.

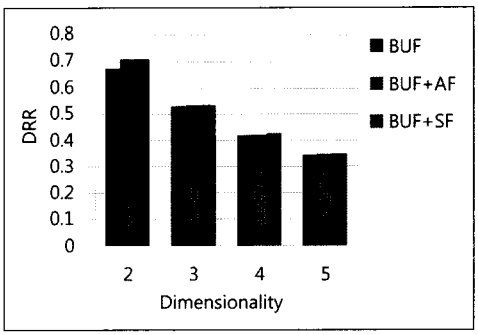

그림 8 랜덤 데이터에 대한 SFT방법의 DRR 결과

그림 8을 보면, 추가 필터링 투플을 전송하는 경우인 BUF+AF, BUF+SF에서 볼 수 있듯이 하나의 필터링 투플을 전송할 때인 BUF보다 DRR이 조금 더 높아진다. 이는 추가 필터링 투플로 인해 더 많은 스카이라인 투플이 필터링 되었기 때문이다.

하지만 랜덤 데이터의 경우, 이전데이터와 현재데이터 간의 상관관계가 전혀 없기 때문에 추가 필터링 투플을 전송하는 경우인 BUF+SF의 결과를 보면 SFT 방법의 적용이 BUF와 비교하여 그다지 성능향상이 없음을 확인할 수 있다.

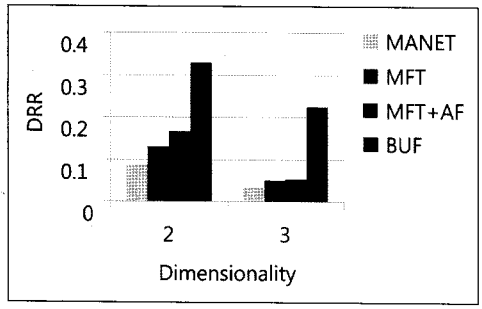

그림 9 실제 데이터에 대한 BUF방법의 DRR 결과

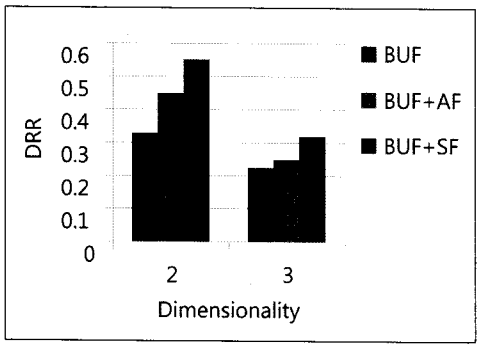

그림 10 실제 데이터에 대한 SFT방법의 DRR 결과

그림 9와 10은 본 논문에서 제안한 BUF방법과 SFT 방법이 랜덤 데이터보다 실제 데이터에서 더 우수한 성능을 나타냄을 보여준다. 랜덤 데이터에서는 SFT의 효과가 미비하였지만, 시간관계성이 반영된 실제 데이터의 경우 SFT의 성능향상이 나타난다. 그림 10에서는 단순히 MFT방법으로 필터링 투플을 추가 선정한 BUF+AF이 하나의 필터링 투플을 전송한 BUF보다, 속성 차수가 2일 때 약 0.12, 3일 때 약 0.02정도 DRR이 더 높게 나타났다. 그러나 SFT방법을 적용한 BUF+SF의 경우 속성 차수가 2일 때 약 0.23, 3일 때 약 0.1정도 높게 나타나 더 우수한 결과를 보였다. 즉, 이전 데이터 값을 기반으로 추가 필터링 투플을 선정하는 예측이 유효하기 때문으로 볼 수 있다.

4.2 통신량

통신량은 필터링 투플 전송 비용과 스카이라인 투플 전송 비용으로 구성된다. 통신 데이터량은 각 차수별로 MANET의 통신량을 1로 보았을 때 다른 실험방법의 통신량의 비율인 PBM(Proportion Based Manet)으로 나타내었다. 예를 들어, PBM이 1.5일 경우 MANET에 비해 통신량이 1.5배 많다는 것을 의미한다.

그림 11의 랜덤 데이터의 경우, DRR에서도 나타나듯이 MFT방법의 경우 MANET방법에 비해 차수가 4와 5일 때는 필터링 성능이 더 높게 나와 통신량이 0.618

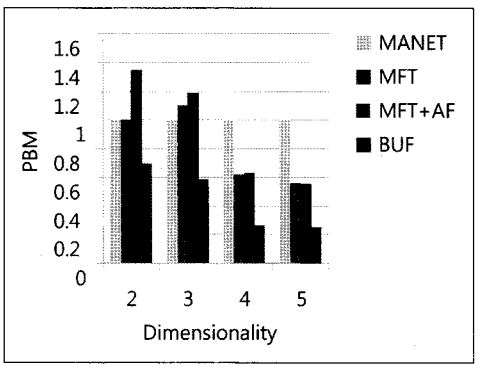

그림 11 랜덤 데이터에 대한 BUF방법의 PBM 결과

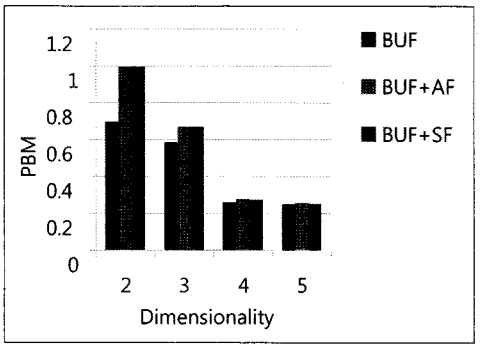

그림 12 랜덤 데이터에 대한 SFT방법의 PBM 결과

배, 0.559배로 더 적게 소비된다. BUF를 활용한 경우 MANET, MFT의 방법들보다 PBM이 전체적으로 낮게 나와 에너지 효율적임을 확인 할 수 있다..

그림 12의 추가 필터링 투플을 전송할 경우인 BUF+AF, BUF+SF를 보자. 차수 2와 3일 때 BUF보다 DRR이 증가하여 전송할 스카이라인 투플 수가 감소했음에도 불구하고, 추가 필터링 투플을 전송하는 비용이 더 많이 들기 때문에 전체적인 통신량은 증가하게 된다. 차수가 2의 경우를 보자. 추가 필터링 투플을 전송하는 경우 단일 필터링 투플 전송과 비교하여 DRR이 거의 동일하거나 조금 향상됨에도 불구하고 PBM은 상당히 증가한다. 이러한 현상은 전체 통신량에서 필터링 투플 전송이 차지하는 비율이 높기 때문에 발생한다. 하지만 차수가 높아질수록 전체 통신량에서 필터링 투플 전송이 차지하는 비중이 상당히 낮아지기 때문에, 추가 필터링 투플을 전송한다고 해도 PEM이 급격하게 변하지 않았다. 즉, 차수 4와 5일때는 추가 필터링 투플을 전송하는 비용과 필터링을 통해 감소되는 비용이 비슷하여 PBM의 변화가 거의 없었다. MFT와 MFT+AF의 결과를 통해서도 이러한 현상을 확인할 수 있다.

그림 13과 14는 실제 데이터에 대한 PBM 결과이다.

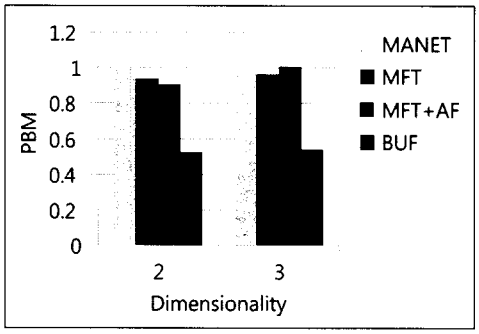

그림 13 실제 데이터에 대한 BUF방법의 PBM 결과

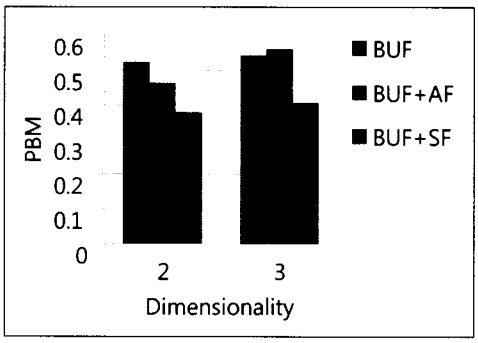

그림 14 실제 데이터에 대한 SFT방법의 PBM 결과

MANET, MFT방법들 보다 본 논문에서 제안한 BUF 방법이 전체적으로 통신량 소비가 낮음을 확인할 수 있다. 그림 14에서 볼 수 있듯이 차수가 2일 때, 추가 필터링 투플을 전송하는 경우인 BUF+AF, BUF+SF방법 모두 BUF에 비해 통신량 측면에서 더 우수하였다. 그러나 차수가 3일 때를 보면 BUF에 비해 BUF+AF의 DRR이 더 높게 나왔지만, 통신량 즉 PBM은 오히려 0.54에서 0.56으로 증가하였다. 필터링으로 인한 통신량 감소보다 추가 필터링 투플의 전송으로 인한 통신량 증가가 더 컸음을 알 수 있다. 반면, BUF+SF의 PBM은 0.40으로 확인되어 더 우수한 성능을 보였다. 즉, 랜덤 데이터에서보다 실제데이터에서 SFT방법을 적용할 때 BUF의 성능이 더욱 개선되었다. SFT방법은 이전데이터를 통해 현재데이터를 예측하는 것에 기반하고 있다. 실제데이터에서는 센서데이터간의 시간관계성의 특성을 잘 반영하고 랜덤데이터는 이러한 특성을 반영하지 않기 때문에, 랜덤데이터보다 실제데이터에서 SFT방법이 더 우수한 성능을 내었다.

5. 결 론

본 연구에서는 센서네트워크 환경에서 연속 스카이라인 질의 처리에 관한 효율적인 방법을 연구하였다. 상향식 필터링 투플 선정 방법은 상위 노드의 정보에 기반한 기존 방법을 보완하기 위해 하위 노드의 과거 스카이라인 정보를 고려하여 필터링 투플을 선정한다. 스카이라인 데이터를 베이스스테이션으로 전송하는 과정에서 하위 노드의 스카이라인 정보를 각 노드의 버퍼에 저장하여, 추후 스카이라인 처리 시 더 큰 지배영역을 갖는 필터링 투플을 선정하는데 활용하게 된다. 이로써 불필요한 스카이라인 데이터 전송을 더욱 감소시킬 수 있다. 이와 더불어 추가 필터링 투플 선정 방법인 SFT 방법을 제안하였다. 하위 노드의 과거 센서 데이터를 기반으로 현재 센서 데이터를 예측하고, 해당 센서 데이터를 생성한 하위 노드와의 거리를 고려하여 추가 필터링 투플을 선정한다. 이를 통해 상향식 필터링 투플 선정방법의 성능을 더욱 향상시켰다.

랜덤 데이터와 실제 데이터를 이용하여 시뮬레이션 실험을 수행하였고, 제안한 방법의 우수성을 확인하였다.

참 고 문 헌

[1] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TAG: a Tiny Aggregation Service for Ad-Hoc Sensor Networks," In *Proc. OSDI*, pp.131-146, 2002.

[2] Y. Yao, J. Gehrke, "Query Processing in Sensor Networks," In *Proc. CIDR*, 2003.

[3] S. Madden, M. J. Franklin, and J. M. Hellerstein, W. Hong, "TinyDB: An Acquisitional Query Processor for Sensor Networks," *ACM Trans. On database Systems*, vol.30, no.1, pp.122-173, 2005.

[4] A. Sharaf, J. Beaver, A. Labrinidis, K. Chrysanthis, "Balancing Energy Efficiency and Quality of Aggregate data in Sensor Networks," *The VLDB Journal*, vol.13, no.4, pp.384-403, 2004.

[5] Z. Huang, C. S. Jensen, H. Lu, and B. C. Ooi, "Skyline Queries Against Mobile Lightweight Devices in MANETs," in *Proc. IEEE ICDE*, p.66, 2006.

[6] Y. Kwon, J. H. Choi, Y. D. Chung, and S. K. Lee, "In-Network Processing for Skyline Queries in Sensor Networks," *IEICE Trans. COMMUN.*, vol.E90-B, no.12, 2007.

[7] B. Cui, H. Lu, Q. Xu, L. Chen, Y. Dai, Y. Zhou, "Parallel Distributed Processing of Constrained Skyline Queries by Filtering," in *Proc. IEEE ICDE*, pp.546-555, 2008.

[8] S. Borzonyi, D. Kossmann, and K. Stocker, "The Skyline Operator," In *Proc. IEEE ICDE*, pp.421-430, 2001.

[9] J. Chomicki, P. Godfrey, J. Gryz, and D. Liang, "Skyline with Presorting," In *Proc IEEE ICDE*, pp.717-719, 2003.

[10] Intel Lab data, "http://www.select.cs.cmu.edu/data/labapp3/index.html"

선 진 호
2007년 중앙대학교 컴퓨터공학과 학사
2007년~현재 한국과학기술원 전산학과
석사과정. 관심분야는 센서네트워크

정 진 완
정보과학회논문지 : 데이타베이스
제 36 권 제 2 호 참조