

센서 네트워크에서 효율적인 다중 이벤트 탐지 (An Efficient Multiple Event Detection in Sensor Networks)

양 동 윤 [†] 정 진 완 ^{††}
(Dong-Yun Yang) (Chin-Wan Chung)

요약 무선 센서 네트워크는 산업 공정 제어, 기계 및 자원 관리, 환경 및 서식지 모니터링 등 다양한 분야에서 활용되고 있다. 그리고 이러한 분야에서 무선 센서 네트워크를 사용하는 주된 목적 중 하나는 이벤트 탐지이다. 사용자의 요청에 따른 다수의 이벤트를 탐지하기 위해서는 센서 데이터와 이벤트의 조건들과의 조인(Join) 연산이 필요하다. 만약 이벤트의 조건들이 너무 많고 그에 비해 센서 노드의 저장 용량이 작을 경우, 센서 노드에 이벤트의 조건들을 저장하여 인-네트워크 조인(In-Network Join)을 할 수가 없다.

본 논문에서는 다수의 이벤트 조건들과 센서 노드의 제한적인 저장 용량을 고려하여, 에너지 효율적으로 다중 이벤트 탐지를 할 수 있는 조건 병합 기반의 인-네트워크 조인 방법을 제안하고자 한다. 이 방법은 원래의 이벤트 조건들 중에 일부를 병합된 조건으로 대체함으로써, 전체 이벤트의 조건의 개수를 줄인다. 메시지 전송에 대한 예측 모델을 만들어서 조건 병합의 대상을 선택하는 알고리즘에 적용하였다. 실험을 통하여 제안한 예측 모델에 대한 검증을 하고, 기존의 방법에 비해서 제안한 방법의 성능이 우수함을 입증하였다.

키워드 : 무선 센서 네트워크, 이벤트 탐지, 조인 연산

Abstract Wireless sensor networks have a lot of application areas such as industrial process control, machine and resource management, environment and habitat monitoring. One of the main objects of using wireless sensor networks in these areas is the event detection. To detect events at a user's request, we need a join processing between sensor data and the predicates of the events. If there are too many predicates of events compared with a node's capacity, it is impossible to store them in a node and to do an in-network join with the generated sensor data.

This paper proposes a predicate-merge based in-network join approach to efficiently detect multiple events, considering the limited capacity of a sensor node and many predicates of events. It reduces the number of the original predicates of events by substituting some pairs of original predicates with some merged predicates. We create an estimation model of a message transmission cost and apply it to the selection algorithm of targets for merged predicates. The experiments validate the cost estimation model and show the superior performance of the proposed approach compared with the existing approaches.

Key words : Wireless Sensor Networks, Event Detection, Join Operation

· 본 연구는 방위사업청과 국방과학연구소의 지원으로 수행되었습니다.

[†] 학생회원 : 한국과학기술원 전산학과
deniz@islab.kaist.ac.kr

^{††} 종신회원 : 한국과학기술원 전산학과 교수
chungcw@kaist.edu

논문접수 : 2009년 5월 22일
심사완료 : 2009년 6월 23일

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 제4호(2009.8)

1. 서론

무선 센서 네트워크는 센서, 저장, 연산, 통신 능력을 갖춘 자동화된 소형 장치인 센서 노드들로 구성되어 있는 무선 네트워크이다. 네트워크를 구성하는 센서 노드들은 특정 영역에 공간적으로 분배되어 서로 협력하여 온도, 습도, 소리, 진동, 압력, 동작, 오염물 등을 모니터링한다. 무선 센서 네트워크는 원래 전장 감시와 같은 군사적인 활용을 위해서 연구·개발되었지만, 현재는 산업 공정 제어, 기계 및 자원 관리, 환경 및 서식지 모니터링, 헬스 케어, 홈 자동화, 교통 통제 등 다양한 분야에

서 활용되고 있다[1,2].

무선 센서 네트워크의 센서 노드들은 소형 기기의 특성상 통신 대역폭이 제한되어 있으며, 소형 배터리를 통해서 전력 공급을 받으므로 전력 소비량이 제한되어 있다. 저가의 저전력 마이크로 프로세서와 메모리를 사용하므로 연산 능력과 저장 능력이 제한되어 있으며, 센서로부터 발생한 신호 데이터는 본질적인 불확실성과 주변의 환경으로부터 잡음을 지니고 있다[3]. 따라서 무선 센서 네트워크를 오랜 시간 동안 연속적으로 운영하기 위해서는 위와 같은 제한적인 하드웨어 조건을 고려한 효율적인 시스템 설계 및 운영이 필요하다. 무엇보다도 전력 소비량이 가장 고려되어야 할 사항이고, 에너지 소비의 가장 큰 비중을 차지하고 있는 센서 노드 간의 메시지 통신량을 줄이는 것이 가장 중요하다.

센서 네트워크의 중요한 활용 방안 중 하나는 이벤트 탐지이다. 센서 네트워크에서 이벤트란 센서 네트워크가 설치된 지역에서 발생하는 사용자가 관심을 갖고 있는 특정 현상을 말한다. 하나의 이벤트는 온도, 습도, 광도와 같은 센서 노드가 가지고 있는 속성의 상한 값과 하한 값의 조합으로 이루어진 하나의 조건(Predicate)으로 표현할 수 있으며, 다수의 이벤트 조건들을 모아서 조건 테이블(Predicate Table)로 표현할 수 있다[4]. 그림 1과 같이 무선 센서 네트워크의 사용자는 탐지하고자 하는 이벤트들을 각각 조건으로 정의하여 베이스스테이션에 요청한다. 베이스스테이션은 사용자의 이벤트 탐지 요청을 받아서 질의 수행 계획을 만들어서 센서 네트워크의 각각의 센서 노드들에게 보내 이벤트 탐지를 위한 질의 수행을 하도록 한다. 센서 네트워크의 센서 노드들은 각자 맡은 특정 지역이나 대상을 질의 계획에 따라 모니터링 하다가 이러한 이벤트 조건을 만족시키는 센서 데이터가 발생하면 베이스스테이션에게 이벤트의 발생 위치, 자원, 시간, 특성 등을 알려주고 베이스스테이션은 결과를 통합하여 사용자에게 이벤트 발생을 보고를 한다.

이러한 무선 센서 네트워크에서 사용자가 정의한 이벤트를 탐지하기 위해서는 센서 노드에서 매 주기마다 발생하는 센서 데이터와 사용자가 탐지를 요청한 이벤트들의 조건 테이블과 조인(Join) 연산이 필요하다. 단

순한 방법으로 센서 노드가 매 주기마다 측정된 값을 매번 베이스스테이션으로 모아서, 베이스스테이션에서 이벤트 조건 테이블과 조인하는 중앙 집중식 방법을 생각해 볼 수 있다. 하지만 중앙 집중식 방법은 각각의 센서 노드에서 매 주기마다 무선 통신을 통해서 베이스스테이션으로 보내야 할 센서 데이터로 인해 통신 비용의 소모가 크다.

이와 비슷한 문제를 해결하기 위하여 초창기 무선 센서 네트워크 연구에서 인-네트워크 프로세싱(In-Network Processing)[5,3,6]이라는 개념을 제안하였다. 이는 센서 노드들이 측정된 센서 데이터를 그대로 베이스스테이션으로 보내지 않고 자식 노드들로 받은 데이터와 함께 평균을 취하거나 하나의 메시지에 넣는 등의 데이터 집약(Data Aggregation)을 하여 원래 데이터의 크기를 줄여서 베이스스테이션으로 보냄으로써, 결과적으로 통신 비용을 줄이는 것을 말한다. 이는 이벤트 탐지를 위한 조인 연산에도 적용이 가능하다. 센서 노드에서 측정된 센서 데이터를 베이스스테이션으로 바로 보내지 않고 노드 내의 메모리에 저장된 이벤트 조건 테이블과 조인 연산을 하여 결과가 있는 경우만 베이스스테이션으로 보내는 인-네트워크 조인(In-Network Join)을 하여 전체 통신 비용을 줄일 수 있다.

센서 네트워크의 활용 수준이 고도화됨에 따라 앞서 설명한 단일 사용자에게 의한 단일 이벤트 탐지만만 아니라 다수 사용자의 다중 이벤트 탐지도 고려해야 한다. 즉, 무선 센서 네트워크 사용자는 단일 이벤트뿐만 아니라 여러 개의 서로 다른 이벤트를 정의하여 무선 센서 네트워크에 다중 이벤트 탐지를 요청할 수도 있다. 또한 단일의 사용자가 아닌 다수의 사용자가 각자가 탐지하고자 원하는 이벤트들을 정의하여 무선 센서 네트워크에 다중 이벤트 탐지를 요청할 수도 있다. 만약 사용자들이 정의한 이벤트가 너무 많아서 센서 노드가 수용할 수 있는 용량을 넘긴 경우, 앞서 언급한 인-네트워크 조인을 적용할 수가 없다.

이러한 다중 이벤트 탐지 환경에서 센서 노드의 저장 용량의 한계 때문에 생기는 문제점을 해결하기 위해 기존 연구[1]는 근접한 센서 노드들끼리 그룹을 형성하여 조건 테이블을 나누어 갖는 방식으로 노드당 할당된 이벤트 조건 테이블의 크기를 줄였으며, 그룹 구성원들끼리 센서 데이터를 주고 받아 분산 조인을 하였다. 하지만 매 주기마다 그룹 구성원들끼리 센서 데이터를 주고 받음으로써 소모되는 통신비용이 크며, 노드 간의 거리가 멀어서 그룹을 형성하기 어려운 경우, 이 방법을 쓸 수가 없다.

본 논문에서 기존 방법의 문제점을 인식하고, 다중 이벤트 탐지 환경에서 센서 노드의 제한적인 저장 용량을

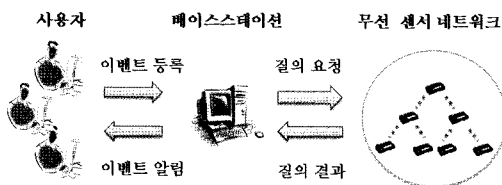


그림 1 무선 센서 네트워크에서 이벤트 탐지 과정

극복하고 통신 비용을 절약할 수 있는, 조건 병합 기반의 인-네트워크 조인 방법을 제안하고자 한다. 이것은 이벤트 조건 테이블의 조건들 중의 일부를 선택하여 그들의 최소 경계 사각형(MBR)으로 병합된 이벤트 조건을 만들어서 대체함으로써 원래 이벤트 조건 테이블의 크기를 줄이는 방법이다. 병합할 이벤트 조건들을 고를 때 각각의 조건들의 선택도(Selectivity)와 조건들의 영역 간의 중복(Overlap)을 고려하여 결과적으로 예상할 수 있는 통신 비용을 최소한으로 줄이는 방법을 제안한다.

1.1 논문의 공헌

본 논문은 다음과 같은 공헌이 있다.

- 이벤트 조건의 개수를 조건 병합을 통해서 줄임으로써, 센서 노드의 저장 용량의 한계를 극복할 수 있는 조건 병합 기반의 인-네트워크 조인 방법을 제안하였다.
- 메시지 전송 비용에 대한 예측 모델을 만들어서 이벤트 조건 병합으로 인한 메시지 전송 비용의 증가를 최소화 할 수 있는 이벤트 조건들을 조건 병합 대상으로 선택하는 방법을 제안하였다.
- 메시지 전송 비용의 예측 모델에 대한 검증과 함께, 합성 데이터와 실제 데이터를 이용하여 기존 방법들과 다양한 비교 실험을 수행함으로써, 본 논문에서 제안한 방법의 우월한 성능을 입증하였다.

지금부터 살펴 볼 본 논문의 구성은 다음과 같다. 우선 2장에서는 본 논문의 연구 주제와 관련된 기존 연구에 대해 소개한다. 그 다음 3장에서는 본 논문에서 제안하고자 하는 조건 병합 기반의 다중 이벤트 탐지 방법에 대해 자세히 설명한 다음, 메시지 전송 비용에 대한 예측 모델과 조건 병합 대상을 선택을 하는 방법에 대해 알아본다. 이어서 4장에서는 실험을 통해 성능을 알아본다. 최종적으로 5장에서 결론을 맺는다.

2. 관련 연구

초기 센서 네트워크 연구인 Cougar[3], TinyDB[5,6]에서는 데이터베이스의 관점에서 센서 네트워크를 위한 질의 기반 데이터 관리 시스템을 구축하였다. 사용자는 자신이 수집하고자 하는 센서 데이터를 SQL과 비슷한 질의어로 정의하고 요청하여, 무선 센서 네트워크 내부적으로 어떻게 질의를 처리하고 데이터를 수집하는지에 대한 신경을 쓸 필요 없이 원하는 데이터를 얻을 수 있다. 그리고 앞서 설명한 인-네트워크 프로세싱 개념을 적용하여 에너지 효율적인 운영이 가능하도록 하였다. 하지만 이들 연구는 단일 사용자에 의한 단일 질의에 최적화가 되어있기 때문에 다수의 복잡한 이벤트 탐지에는 적용하기가 힘들다.

DSWare[7]는 센서 데이터의 정확성과 연관성을 고

려하면서 효율적이며 견고하게 이벤트를 탐지하며, 이벤트 자체의 실시간 제약조건을 충족시키는 서비스를 제공하는 실시간 이벤트 탐지 서비스를 제공하는 무선 센서 네트워크 미들웨어(Middleware)다. 이를 위하여 이벤트를 원자(Atomic) 이벤트와 합성(Compound) 이벤트로 구분하여 이벤트 계층도를 구성하였으며, 합성 이벤트를 구성하는 종속(Sub) 이벤트들 간의 관계를 상대적인 중요도, 이전 데이터, 통계 모델 등을 고려하여 신뢰 함수(Confidence Function)로 표현하였다. 또한 절대 유효 구간(Absolute Validity Interval)을 각각의 하위 이벤트에 연관시켜 실제 환경과 탐지된 이벤트의 시간적인 일관성을 유지하였다. 하지만 DSWare는 다중 이벤트 탐지 환경에서 센서 노드의 저장 용량의 한계 때문에 생기는 문제점을 고려하지 않았다.

[8]는 단순 역치(Threshold) 기준의 이벤트가 아닌 좀 더 복잡한 이벤트 탐지를 위하여 시공간 패턴(pattern) 기반의 이벤트 탐지 방법을 사용하였다. 사용자는 시공간 패턴 기반의 일반화된 이벤트 정의나 자주 쓰이는 타입의 특정 시공간 이벤트 정의를 사용하여 원하는 이벤트 탐지 질의를 요청할 수 있다. 효율적인 패턴 매칭(Pattern Matching)을 위해서 센서 노드들로부터 얻은 센서 데이터를 네트워크 내에서 선형 회귀를 사용하여 센서 데이터 분포의 Contour Map을 구성하고 이를 사용자가 정한 이벤트 패턴과 매칭시켜 이벤트를 탐지한다. 하지만 이 연구는 패턴 기반의 복잡한 단일 이벤트 처리에 중점을 두었기 때문에, 본 논문에서 다루고자 하는 이벤트와 범위가 다르며, 다수의 이벤트 탐지나 센서 노드의 저장 용량의 한계를 고려하지 않았다.

[9]에서는 집약(Aggregation)이나 선택(Selection) 등의 단순 질의가 아닌 특정 시간 윈도우(Time Window) 내에서 자기 조인(Self-Join)을 통해 탐지가 가능한 복잡한 이벤트에 대한 에너지 효율적인 처리에 대해서 연구하였다. 단순 중앙 집중식이나 플러딩(Flooding) 방식이 아닌 2단계 인-네트워크 조인을 통하여, 문제를 해결하였다. 1단계에서는 사전 필터링을 통해서 최종 결과로 남을 가능성이 있는 후보 데이터를 베이스스테이션으로 모으고, 2단계에서는 그 데이터를 관련 있는 센서 노드들에게 전달하여 네트워크 내에서 시간 윈도우(Time Window) 조인을 하도록 한다. 하지만 이 연구는 본 논문에서 다루고자 하는 이벤트와 범위가 다르며, 다수의 이벤트 탐지에 대하여 고려하지 않았다.

REED[4]는 사용자가 탐지를 요청한 다중 이벤트의 조건을 테이블로 표현하여 센서 네트워크의 노드에 저장하여 센서데이터와 인-네트워크 조인을 통해서 효율적으로 이벤트를 탐지 할 수 있도록 하였다. 센서 노드의 저장 용량의 한계 때문에 생기는 문제점을 해결하기

위해 그룹 기반의 분산 조인 방법을 제안하였다. 근접한 센서 노드들끼리 그룹을 형성하여 이벤트 조건 테이블을 나누어 갖는 방식으로 노드당 할당된 이벤트 조건 테이블의 크기를 줄였다. 센서 노드에서 발생하는 센서 데이터로부터 이벤트 발생을 탐지하기 위해서 그룹의 구성원들끼리 매 주기마다 서로의 센서 데이터를 주고 받아 각각의 센서 노드에서 전체 이벤트 조건의 일부와 분산 조인을 하여 그 결과를 베이스스테이션에서 취합하였다. 또한 REED에서는 분산 조인의 성능을 높이기 위해서 센서 데이터가 사용자가 요청한 원래의 이벤트 조건들과 조인이 되는지를 알 수 있는 Bloom Filter[10]를 사용하고 있다. Bloom Filter를 이용하여 사용자가 요청한 원래의 이벤트 조건과 조인되지 않아 주변 그룹 구성원 공유할 필요가 없는 센서 데이터 공유를 일부분 막아주었다.

하지만 REED의 접근 방법은 전체적으로 매 주기마다 그룹 구성원들끼리 센서 데이터를 주고 받음으로써 소모되는 통신비용이 크다. 그리고 센서 노드의 용량이 많이 부족해 여러 노드들이 그룹을 형성할 필요가 있는데, 센서 노드 간의 거리가 멀어서 그룹을 형성하기 어려운 경우, 이 방법의 효율이 떨어진다. 또한 어떤 센서 노드가 한 주기에 센서 데이터와 조인되는 이벤트 조건들이 많은 경우, 하나의 노드에서 통합된 조인 결과 메시지나 센서 데이터를 보냄으로써 간단하게 해결되는 것들이 여러 노드에서 조인 연산을 하여 조인된 결과를 각자 따로 베이스스테이션으로 보내야하기 때문에 전체적으로 메시지 전송 비용이 더 들어간다.

3. 조건 병합 기반의 인-네트워크 조인 방법

3.1 개요

본 장에서는 무선 센서 네트워크에서 다중 이벤트 탐지를 위한 조건 병합 기반의 인-네트워크 조인 방법에 대해 예제와 함께 자세하게 소개하고자 한다.

본 논문에서 다루고자 하는 이벤트는 센서 노드가 가

지고 있는 속성의 상한 값과 하한 값의 조합으로 이루어진 조건(Predicate)로 표현이 가능한 이벤트이다. 다수의 사용자가 지속적으로 탐지 요청한 M 개의 이벤트 조건들이 존재할 때, m 개의 이벤트 조건을 수용할 수 있는 센서 노드들로 이루어진 무선 센서 네트워크에서 이벤트 탐지 결과를 보고하기 위해서 베이스스테이션으로 보내는 메시지 전송 비용이 되도록이면 줄이면서 효율적으로 이벤트를 탐지하는 것이 본 논문에서 해결하고자 하는 문제이다.

우선 이해를 돕기 위해서 측정 가능한 센서 속성의 개수가 2개인 센서 노드 2개로 구성된 무선 센서 네트워크 예제를 마련하였다. 그림 1처럼 다수의 사용자가 다수의 이벤트 조건들을 베이스스테이션에 등록하여 그림 2(a)처럼 5개의 이벤트 조건이 포함된 이벤트 조건 테이블이 존재하며, 그림 2(b)처럼 2개의 센서 노드에서 센서 데이터가 발생한다고 가정하자. 사용자들은 이벤트 탐지 시 결과로서 이벤트가 발생한 노드 번호, 시각, 이벤트 번호만을 알기를 원한다. 그리고 센서 데이터 발생 확률이 전체 정의역($attr_1, attr_2 \in [0, 100]$)에서 균일하다.

조건 병합 기반의 인-네트워크 조인 방법은 이벤트 조건의 개수(M)와 센서 노드의 용량(m)에 따라서 사례 1. 이벤트 조건의 개수가 센서 노드가 수용할 수 있는 용량보다 작거나 같은 경우($M \leq m$)와 사례 2. 이벤트 조건의 개수가 센서 노드가 수용할 수 있는 용량보다 큰 경우($M > m$)로 구분하여 처리한다.

사례 1. 이벤트 조건의 개수가 센서 노드가 수용할 수 있는 용량보다 작거나 같은 경우($M \leq m$)

전체 이벤트 조건들을 그대로 센서 노드에 저장하여 매 주기마다 측정된 센서 데이터와 인-네트워크 조인을 한다. 이벤트가 발생하여 베이스스테이션으로 보내주어야 할 조인의 결과가 있는 경우, 노드 번호, 시각, 이벤트 번호와 함께 사용자가 요청한 센서 데이터의 일부 속성값을 메시지에 담아 베이스스테이션으로 보내 사용자에게 이벤트 발생 보고를 한다.

Event No.	attr ₁ low	attr ₁ high	attr ₂ low	attr ₂ high
1	20	50	70	90
2	60	80	55	75
3	75	95	25	40
4	20	45	20	40
5	10	40	10	30

(a) 이벤트 조건 테이블

Node No.	time stamp	attr ₁	attr ₂
1	1	65	60
2	1	25	27
1	2	65	70
2	2	30	60
...

(b) 센서 데이터

그림 2 이벤트 조건 테이블과 센서 데이터 예제

예를 들어, 그림 2의 예제에서 센서 노드의 용량이 5보다 크다고 가정하면, 5개의 조건으로 이루어진 이벤트 조건 테이블을 그대로 센서 노드에 저장하여 인-네트워크 조인한다. 시각 1에 노드 1번에서 발생한 데이터는 이벤트 조건 2번과 조인되어 자신의 노드 번호 1, 시각 정보 1, 이벤트 번호 2를 조인 결과 메시지로 보낸다. 시각 1에 노드 2번에서 발생한 데이터는 이벤트 조건 4, 5번과 동시에 조인되어 자신의 노드 번호 2, 시각 정보 1, 이벤트 번호 4, 5를 하나의 조인 결과 메시지에 담아 보낸다.

만약 하나의 노드에서 다수의 이벤트가 한 주기에 동시에 발생할 경우, 조인 결과를 보내는 하나의 메시지에 여러 개의 이벤트 번호가 포함될 수 있다. 하지만 많은 이벤트 조건을 동시에 만족하여 그로 인해 메시지가 크기가 너무 커지는 경우, 세미 조인(Semi-Join)[11]을 적용하여 센서 데이터의 모든 속성 값을 담은 메시지를 베이스스테이션으로 보내 베이스스테이션에서 원래의 이벤트 조건들과 다시 조인을 하여 그 결과에 따라 사용자에게 이벤트 발생 보고를 한다.

사례 2. 이벤트 조건의 개수가 센서 노드가 수용할 수 있는 용량보다 큰 경우($M > m$)

이벤트 조건들 중의 일부를 선택하여 그들의 최소 경계 사각형(MBR)으로 병합된 조건을 만들어서 대체함으로써 전체 이벤트 조건의 개수를 센서 노드가 수용할 수 있는 이벤트 조건의 개수만큼 줄인다. 줄어든 이벤트 조건들을 센서 노드에 저장하면 센서 데이터와 인-네트워크 조인이 가능해지면서, 결과적으로 센서 노드와 베이스스테이션 간의 필요가 없는 데이터 전송을 줄일 수가 있다.

예를 들어, 그림 2의 예제에서 센서 노드의 용량이 4라고 가정하면, 5개의 조건으로 이루어진 이벤트 조건 테이블을 그대로 센서 노드에 저장하여 인-네트워크 조

인할 수가 없으므로, 조건 병합을 통하여 인-네트워크 조인을 하려면 전체 이벤트 조건의 개수를 4개로 줄일 필요가 있다. 만약 이벤트 조건 4번과 5번을 병합하여 병합된 이벤트 조건 6번으로 대체하는 경우, 전체 이벤트 조건의 개수가 4개로 줄어들어 하나의 센서 노드에서 전체 이벤트 조건을 센서 노드에 저장할 수 있다. 그림 3에서 이러한 조건 병합의 모습을 입체적으로 보여주기 위해서 이벤트 조건 테이블을 2차원 상에서 표현하였다.

센서 노드에서 발생한 센서 데이터가 원래의 이벤트 조건과 조인이 되는 경우, 앞서 언급한 이벤트 조건의 개수가 센서 노드가 수용할 수 있는 용량보다 작거나 같은 경우($M \leq m$)와 같은 방식으로 조인된 결과를 베이스스테이션으로 보내 사용자에게 조인 결과에 따른 이벤트 발생 보고를 한다. 하지만 센서 노드에서 발생한 센서 데이터가 병합된 이벤트 조건과 조인이 되는 경우, 축정한 센서 데이터의 모든 속성 값들을 그대로 베이스스테이션으로 보내 베이스스테이션에서 원래의 이벤트 조건들과 조인을 하여 그 결과에 따라 사용자에게 이벤트 발생 보고를 한다.

예를 들어, 그림 3의 조건 병합 예제에서 시각 1에 노드 1번에서 발생한 데이터는 병합되지 않은 원래의 이벤트 조건 2번과 조인되어 자신의 노드 번호 1, 시각 정보 1, 이벤트 번호 2를 조인 결과 메시지로 보낸다. 하지만 시각 1에 노드 2번에서 발생한 데이터는 병합된 이벤트 조건 6번과 조인되어 자신의 노드 번호 2, 시각 정보 1, 센서 데이터 25, 27을 메시지에 담아 베이스스테이션으로 보내, 그 곳에서 다시 원래의 이벤트 조건 테이블과 조인하여 결과를 사용자에게 이벤트 발생을 보고한다.

그런데, 조건 병합을 이용하여 전체 조건의 개수를 줄이는 방법은 센서 노드의 센서 데이터가 병합된 이벤트 조건과 조인된 경우 베이스스테이션으로 보내야 하는 센

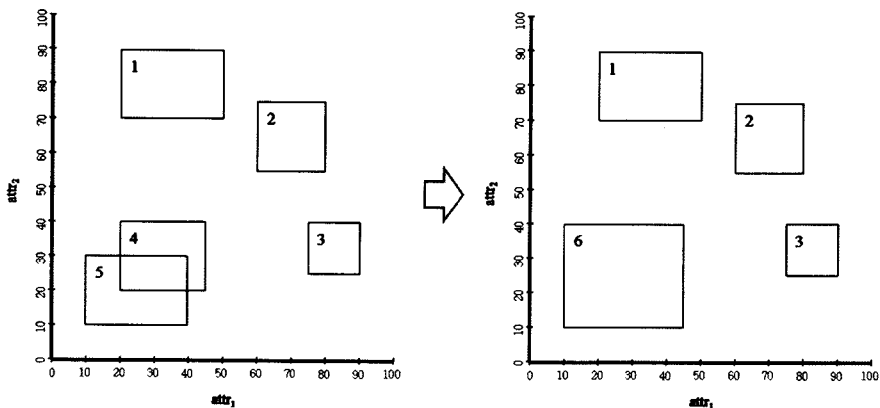


그림 3 조건 병합 선택 예제-1

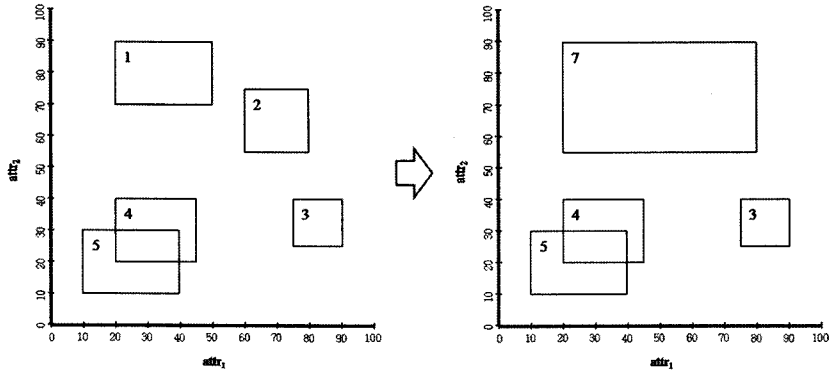


그림 4 조건 병합 선택 예제-2

서 데이터의 크기가 원래의 이벤트 조건과 조인되어 보내는 조인 결과 메시지보다 일반적으로 크다. 또한 발생한 센서 데이터가 병합되기 이전의 원래의 이벤트 조건들의 영역에 포함되지 않지만 병합된 이벤트 조건과는 조인이 되는 경우에도 센서 데이터를 베이스스테이션으로 보내야 한다. 따라서 조건 병합 대상을 잘못 선택한 경우, 그로 인한 메시지 전송 비용의 손실이 클 수가 있다.

구체적인 예를 들어, 그림 4는 조건 병합 대상을 잘못 선택한 경우이며, 이벤트 조건 1번과 2번을 병합하여 7번으로 대체하여, 그림 3보다 조건 병합으로 인한 손실이 크다. 왜냐하면 병합된 이벤트 조건 7번은 6번보다 면적이 커서 센서 데이터와 조인 확률도 더 크고, 자신의 영역 중 원래의 이벤트 조건들의 영역과 겹치지 않은 영역의 크기도 커서 일반적으로 베이스스테이션으로 보내는 메시지의 크기가 커지고 메시지의 전송 횟수가 많아지기 때문이다.

따라서 이러한 문제를 해결하기 위해서 조건 병합을 위한 적절한 대상을 선택하는 방법이 필요하다. 다음 3.2장에서 이를 위한 메시지 전송 비용에 대한 예측 모델과 3.3장에서 그 예측 모델을 이용하여 적절한 조건 병합의 대상을 선택하는 방법을 제안하고자 한다.

3.2 메시지 전송 비용 예측 모델

3.2.1 기본 가정

본 논문에서 제안하는 메시지 전송 비용 예측 모델에 대한 기본 가정은 다음과 같다.

첫째, 하나의 메시지를 노드 간에 전송하는데 드는 비용은 메시지의 크기와 비례한다. 현실적으로 메시지 내 헤더 정보, 통신 상태 등 다양한 변수가 존재하기 때문에 하나의 메시지를 전송하는데 드는 비용이 정확하게 메시지의 크기와 비례하지는 않는다. 하지만 메시지의 크기가 크면 클수록 어려울도 높고 전송하는데 더 많은 비용이 든다고 알려져 있으므로 문제 해결의 복잡도를 줄이기 위해서 메시지 크기에 비례하는 것으로 가정한다.

둘째, 메시지 전송 비용 모델에서는 아래와 같은 두 가지 종류의 메시지가 센서 노드로부터 베이스스테이션까지 전송되는 비용만 고려한다.

- 조인 결과를 보내는 메시지 M_{jr} : 노드 번호 + 시각 + 발생한 이벤트들의 번호들 + 센서 데이터의 일부 속성 값들
- 센서 데이터를 보내는 메시지 M_{sd} : 노드 번호 + 시각 + 센서 데이터의 모든 속성 값들

M_{jr} 와 M_{sd} 는 둘 다 센서 노드 내에서 센서 데이터와 이벤트 조건들과 조인 연산의 결과로서 베이스스테이션으로 전송되는 메시지로서 노드 번호와 보낸 시각이 포함되어 있다. 조인 결과를 보내는 메시지 M_{jr} 는 센서 데이터가 원래 이벤트 조건들과 조인되는 경우 베이스스테이션으로 보내는 메시지로서, 실제 발생한 원래 이벤트들의 번호들과 사용자가 요청한 센서 데이터의 일부 속성 값들만이 포함되어 있다. 센서 데이터를 보내는 메시지 M_{sd} 는 센서 데이터가 병합된 이벤트 조건들과 조인되거나 일정량 이상의 다수의 이벤트 조건들과 동시에 조인되는 경우, 베이스스테이션에서 원래 이벤트 조건들과 다시 조인(Semi-Join)을 하기 위해 베이스스테이션으로 보내는 메시지로서, 센서 데이터의 모든 속성의 값이 포함되어 있다.

그 밖에 무선 센서 네트워크 초기에 노드들 간의 네트워크를 구성하는 데 드는 메시지 전송 비용, 질의 수행 계획과 함께 이벤트 조건들을 베이스스테이션에서 노드로 보내는 데 필요한 메시지 전송 비용, 노드들 간의 송수신 상태를 확인하기 위한 메시지 전송 비용 등이 있을 수 있다. 하지만 이들은 센서 네트워크를 유지하기 위하여 기본적으로 소모되는 비용이고 이벤트 탐지 전략에 따라서 크게 영향을 받지 않기 때문에 제외하기로 한다.

셋째, 무선 센서 네트워크가 설치된 지역에서 발생하는 센서 데이터가 가질 수 있는 값의 일반적인 분포를

사전 지식이나 히스토그램을 통해 알 수 있다. 앞서 서론에서 언급한 산업 공정 제어, 기계 및 자원 관리, 환경 및 서식지 모니터링, 헬스 케어, 홈 자동화, 교통 통제 등 다양한 분야에서 센서 네트워크를 활용할 때, 설치하려는 지역이나 자원에 대한 정보가 이미 축적되어 있는 경우가 많다. 따라서 이러한 정보를 활용하면 센서 노드의 속성이 가질 수 있는 값의 분포를 미리 알 수 있으며, 좀 더 효율적인 운영이 가능해진다.

3.2.2 예측 모델

하나의 센서 노드에서 매주기마다 발생하는 센서 데이터에 의해서 베이스스테이션으로 보내는 메시지 전송 비용에 대해 알아보기로 하자. 본 예측 모델에서는 사용된 기호는 표 1을 참조하면 된다.

표 1 기호 설명 표

기호	설명
d	센서 노드의 깊이
D	센서 데이터의 정의역
A	센서 데이터의 특정 영역 (ACD)
p	이벤트 조건
P	이벤트 조건 테이블
MOL	이벤트 조건이 가장 많이 겹쳐진 영역의 겹쳐진 횟수
C_{sd}	하나의 센서 데이터를 보내는 메시지 M_{sd} 의 노드 간 단위 전송 비용
$C_{jr}(i)$	i 개의 이벤트 조건이 겹쳐진 영역에서 발생한 센서 데이터에 의한 조인 결과 메시지 M_{jr} 의 노드 간 단위 전송 비용
$A_o(A, P)$	영역 A 내에서 P 에 속한 원래의 이벤트 조건들이 차지하고 있는 영역
$OA_i(A, P)$	영역 A 내에서 P 에 속한 i 개의 이벤트 조건이 겹쳐진 영역
$Pr(A)$	매 주기당 영역 A 에서 센서 데이터가 발생할 확률
$A_m(A, P)$	영역 A 내에서 P 에 속한 병합된 이벤트 조건들이 차지하고 있는 영역
$A_{o-m}(A, P)$	$A_o(A, P) - A_m(A, P)$: A 의 영역 중 원래의 이벤트 조건들만 차지하고 있는 영역
$MBR(p_i, p_j)$	두 이벤트 조건 p_i 와 p_j 의 최소 경계 사각형(MBR)

병합 여부와 상관 없이 이벤트 조건은 p 로, 이벤트 조건을 모은 이벤트 테이블을 P 로 표현한다. 문제 해결을 위하여 센서 데이터의 정의역 D 가 아닌 특정 영역 A 에서 발생하는 센서 데이터에 대해서 비용 모델을 고려한다. 만약 A 가 D 와 같은 경우, 하나의 센서 노드에서 매주기마다 발생하는 센서 데이터에 의해서 베이스스테이션으로 보내는 메시지 전송 비용과 같아진다. 앞서 기본 가정에 언급한 센서 데이터가 가질 수 있는 값의 분포에 대한 사전 정보는 구체적으로 매 주기당 영

역 A 에서 센서 데이터가 발생할 확률 $Pr(A)$ 로 표현되었다.

조건 병합을 통해 크기가 줄어든 이벤트 조건 테이블 P 를 깊이가 d 인 센서 노드에 저장하여 인-네트워크 조인을 한다고 가정하자. 하나의 센서 노드에서 영역 A 에서 발생한 센서 데이터와 이벤트 조건 테이블 P 의 조인 연산의 결과로서 센서 노드에서 베이스스테이션으로 보내는 매 주기당 평균 메시지 전송 비용 $C(A, P)$ 은 원래의 이벤트 조건과 조인에 의한 메시지 전송 비용 $C_o(A, P)$ 과 병합된 이벤트 조건과의 조인에 의한 메시지 전송 비용 $C_m(A, P)$ 으로 나눌 수 있다.

$$C(A, P) = C_o(A, P) + C_m(A, P) \quad (6)$$

구체적인 예를 들어 앞서 그림 4의 조건 병합 예제의 결과로 나온 이벤트 조건 3, 4, 5, 7번으로 이루어진 이벤트 조건 테이블 $P1$ 을 깊이가 d 인 센서 노드에 저장하여 인-네트워크 조인을 하는 경우를 생각해보자. 그림 5에서 점선으로 표시한 영역 $A1$ 에서 발생한 센서 데이터와 이벤트 조건 테이블 $P1$ 의 조인 연산의 결과로서 센서 노드에서 베이스스테이션으로 보내는 메시지의 매 주기당 평균 전송 비용 $C(A1, P1)$ 은 $C_o(A1, P1)$ 와 $C_m(A1, P1)$ 의 합으로 표현할 수 있다. $C_o(A1, P1)$ 은 영역 $A1$ 에 포함된 원래의 이벤트 조건 4, 5 번의 전체 영역에서 발생한 센서 데이터에 의해서 발생하는 메시지 전송 비용이며, $C_m(A1, P1)$ 은 영역 $A1$ 에 포함된 병합된 이벤트 조건 7 번의 일부 영역에서 발생한 센서 데이터에 의해서 발생하는 메시지 전송 비용이다.

하나의 센서 노드에서 영역 A 에서 발생한 센서 데이터와 원래의 이벤트 조건과 조인에 의하여 베이스스테이션으로 보내는 메시지 M_{jr} 의 매 주기당 평균 전송 비용 $C_o(A, P)$ 은 노드의 깊이 d 에 비례한다. 그리고 매

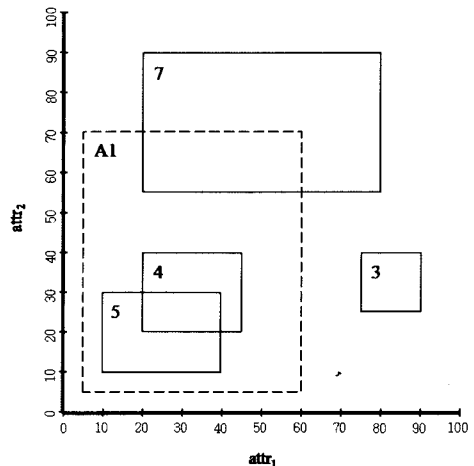


그림 5 비용 예측 모델을 위한 예제

시지 M_{jr} 의 노드 간 단위 전송 비용은 이벤트 조건이 겹치는 정도에 따라서 달라짐으로 이것을 겹치는 정도에 따라 비용이 달라지는 함수 $C_{jr}(i)$ 로 표현하였다. 센서 데이터가 병합된 이벤트 조건과 원래의 이벤트 조건 둘 다와 동시에 조인이 되는 경우, 병합된 이벤트와 조인 결과로서 보내는 원래의 센서 데이터만 베이스스테이션으로 보내면 되므로, 원래의 이벤트 조건과 조인에 의한 메시지 전송 비용 $C_o(A, P)$ 을 구할 때는 순수하게 원래의 이벤트 조건만 존재하는 영역 $A_{o-m}(A, P)$ 만을 고려하여 계산한다. 순수하게 원래의 이벤트 조건만 존재하는 영역 $A_{o-m}(A, P)$ 을 이벤트 조건이 겹치는 정도에 따라 영역을 나눠서($OA_i(A_{o-m}(A, P), P)$), 각각의 영역에서 센서 데이터가 발생할 확률 $Pr(OA_i(A_{o-m}(A, P), P))$ 과 대응하는 조인 결과의 메시지의 전송 비용 $C_{jr}(i)$ 과 노드의 깊이 d 를 곱해준 것을 전부 더해줌으로써 영역 A1에서 발생한 센서 데이터가 원래의 이벤트 조건과 조인에 의한 매 주기당 평균 메시지 전송 비용 $C_o(A, P)$ 을 예측할 수 있다.

$$\hat{C}_o(A, P) = \sum_{i=1}^{MOL} d * C_{jr}(i) * Pr(OA_i(A_{o-m}(A, P), P)) \quad (2)$$

그림 5의 예제에 식 (2)을 적용하면 다음과 같다. A1의 영역 중 원래의 이벤트 조건들만 차지하고 있는 영역 $A_{o-m}(A1, P1)$ 은, 영역A1에 포함된 원래의 이벤트 조건들이 병합된 이벤트들과 겹치는 부분이 없으므로, 원래의 이벤트 조건 4번과 5번의 전체 합집합 영역이다. 이 영역 $A_{o-m}(A1, P1)$ 은 원래의 이벤트 조건 4번 또는 5번만 존재하는 영역 $OA_1(A_{o-m}(A, P), P)$ 와 원래의 이벤트 조건 이벤트 조건 4번과 5번이 둘 다 존재하는 영역 $OA_2(A_{o-m}(A, P), P)$ 으로 나눌 수 있다. 이 두 영역에서 센서 데이터가 발생할 확률은 각각 $Pr(OA_1(A_{o-m}(A, P), P))$ 과 $Pr(OA_2(A_{o-m}(A, P), P))$ 이다. 그리고 이 두 영역에서 센서 데이터의 발생하는 겹치는 정도에 따른 조인 결과 메시지의 노드 간 단위 전송 비용은 각각 $C_{jr}(1)$ 과 $C_{jr}(2)$ 이다. 따라서 노드의 깊이 d 와 함께 각 영역에서의 조인 결과의 메시지의 노드 간 단위 전송 비용과 센서 데이터가 발생할 확률을 곱하면 영역 A1에서 발생한 센서 데이터가 원래의 이벤트 조건과 조인에 의한 베이스스테이션으로 보내는 매 주기당 평균 메시지 전송 비용 $C_o(A1, P1)$ 을 예측할 수 있다.

$$\hat{C}_o(A1, P1) = d * C_{jr}(1) * Pr(OA_1(A_{o-m}(A1, P1), P1)) + d * C_{jr}(2) * Pr(OA_2(A_{o-m}(A, P), P))$$

하나의 센서 노드에서 병합된 이벤트 조건과의 조인에 의하여 베이스스테이션으로 보내는 메시지 M_{sd} 의 매 주기당 평균 전송 비용 $C_m(A, P)$ 은 노드의 깊이 d 에 비례하고, 센서 데이터를 보내는 메시지의 일정한 전송 비용 C_{sd} 와 비례하므로, 병합될 영역에서 값이 발생

할 확률 $Pr(A_m(A, P))$ 을 함께 곱해줌으로써 다음과 같이 예측할 수 있다.

$$\hat{C}_m(A, P) = d * C_{sd} * Pr(A_m(A, P))$$

그림 5의 예제에 식 (3)를 적용하면 다음과 같다. 영역 A1 내에서 P에 속한 병합된 이벤트 조건들이 차지하고 있는 영역 $A_m(A1, P1)$ 는 영역 A1에 포함된 병합된 이벤트 조건 7번의 일부 영역이다. 이 영역에서 센서 데이터가 발생할 확률은 $Pr(A_m(A1, P1))$ 이다. 따라서 노드의 깊이 d 와 함께 조인 결과의 메시지의 노드 간 단위 전송 비용과 센서 데이터가 발생할 확률을 곱하면 영역 A1에서 발생한 센서 데이터가 원래의 이벤트 조건과 조인에 의한 베이스스테이션으로 보내는 매 주기당 평균 메시지 전송 비용 $C_o(A1, P1)$ 을 예측할 수 있다.

$$\hat{C}_m(A1, P1) = d * C_{sd} * Pr(A_m(A1, P1))$$

따라서 하나의 센서 노드에서 특정 영역 A에서 발생한 센서 데이터와 이벤트 조건 테이블 P의 조인으로 베이스스테이션으로 보내는 매 주기당 메시지 전송 비용 $C(A, P)$ 은 위 두 식 (2)와 (3)를 더하여 다음과 같이 예측할 수 있다.

$$\hat{C}(A, P) = \hat{C}_o(A, P) + \hat{C}_m(A, P) \quad (4)$$

3.3 병합 대상 선택 방법

이벤트 조건 테이블 P에 속하는 임의의 두 이벤트 조건 p_i, p_j 가 병합하여 하나의 병합된 이벤트 조건으로 바꾸는 경우에, 센서 노드의 센서 데이터가 그 병합된 이벤트 조건과 조인된 경우 베이스스테이션으로 보내야 하는 센서 데이터 메시지 M_{sd} 의 크기가 원래의 이벤트 조건과 조인되어 보내는 조인 결과 메시지 M_{jr} 의 크기보다 일반적으로 크다. 또한 병합된 이벤트 조건과 조인이 되어서 센서 데이터를 베이스스테이션으로 보내다 시 원래의 이벤트 조건들과 조인을 하였으나, 그 어떤 원래의 이벤트 조건들과도 조인이 되지 않는 경우도 있을 수 있다. 따라서 두 이벤트 조건 p_i, p_j 를 병합을 통해서 발생할 수 있는 메시지 전송 비용의 증가를 손실 비용 $C_{loss}(p_i, p_j)$ 으로 정의하여 $C_{loss}(p_i, p_j)$ 을 최소화하는 쪽으로 이벤트 조건 병합을 하고자 한다.

이벤트 조건 테이블 P에 속하는 두 이벤트 조건 p_i, p_j 가 병합되어서 이벤트 조건 테이블 P가 P'로 바뀌었다고 하자. 병합된 이벤트가 가지는 영역을 $MBR(p_i, p_j)$ 이라고 하면, 조건 병합 전 $MBR(p_i, p_j)$ 에서 발생하는 센서 데이터에 의해 예상되는 메시지 전송 비용 \hat{C}_{before} 와 조건 병합 후 $MBR(p_i, p_j)$ 에서 발생하는 데이터에 의해 예상되는 메시지 전송 비용 \hat{C}_{after} 를 앞서 소개한 식 (4)를 적용하여 아래와 같이 표현할 수 있다. 식 (6)에서 \hat{C}_{after} 를 구할 때, $\hat{C}_o(MBR(p_i, p_j), P')$ 항이 소거된 이유는 조건 병합 후에는 영역 $MBR(p_i, p_j)$ 내에서 원래

이벤트 조건만 차지고 하고 있는 영역이 없기 때문이다.

$$\hat{C}_{before} = \hat{C}(MBR(p_i, p_j), P) \quad (5)$$

$$\begin{aligned} \hat{C}_{after} &= \hat{C}(MBR(p_i, p_j), P') \\ &= \hat{C}_o(MBR(p_i, p_j), P') + \hat{C}_m(MBR(p_i, p_j), P') \\ &= \hat{C}_m(MBR(p_i, p_j), P') \end{aligned} \quad (6)$$

병합된 이벤트 조건은 두 이벤트 조건의 최소 경계 사각형의 영역 $MBR(p_i, p_j)$ 에 걸쳐있게 되므로, 두 이벤트 조건 p_i, p_j 의 병합은 $MBR(p_i, p_j)$ 영역 내에서 발생한 센서 데이터에 의한 메시지 전송 비용에만 변화를 주게 되며 $MBR(p_i, p_j)$ 영역 외에서 발생한 센서 데이터에 의한 메시지 전송 비용에는 영향을 주지 않는다. 따라서 $MBR(p_i, p_j)$ 에서 발생하는 센서 데이터에 의해 예상되는 조건 병합 전후 메시지 전송 비용 \hat{C}_{before} 와 \hat{C}_{after} 를 사용하여 이벤트 조건 테이블 P에 속하는 두 이벤트 조건 p_i, p_j 의 병합을 통해서 발생할 수 있는 손실 비용 $C_{loss}(P, p_i, p_j)$ 의 예상 값을 다음과 같이 정의 할 수 있다.

$$\hat{C}_{loss}(P, p_i, p_j) = \hat{C}_{after} - \hat{C}_{before} \quad (7)$$

그림 3의 예제에 식 (7)을 적용해 보자. 이벤트 조건 4번(p_4)과 이벤트 조건 5번(p_5)을 병합하여 병합된 이벤트 조건 6번(p_6)을 만드는 경우, 이런 조건 병합으로 인한 손실 비용 $C_{loss}(P, p_4, p_5)$ 는 조건 병합 전 $MBR(p_4, p_5)$ 에서 발생하는 센서 데이터에 의해 예상되는 메시지 전송 비용 \hat{C}_{before} 와 조건 병합 후 $MBR(p_4, p_5)$ 에서 발생하는 데이터에 의해 예상되는 메시지 전송 비용 \hat{C}_{after} 의 차로써 아래와 같다.

$$\hat{C}_{loss}(P, p_i, p_j) = \hat{C}(MBR(p_4, p_5), P) - \hat{C}_m(MBR(p_4, p_5), P')$$

이러한 조건 병합 손실 비용의 예상 값 $\hat{C}_{loss}(P, p_i, p_j)$ 를 활용하여 표 2와 같은 계층적 클러스터링 알고리즘을 실행시킬 경우 최적의 이벤트 조건 병합 대상을 구할 수가 있다. 클러스터링의 대상은 이벤트 조건들이며, $\hat{C}_{loss}(P, p_i, p_j)$ 를 두 이벤트 조건 간의 거리 척도(Distance Measure)로 하여 만들어지는 모든 클러스터에 대하여 조건 병합을 하여 새로운 병합된 이벤트 조건으로 대체를 한다. 이를 통하여 센서 노드의 용량만큼 전체 이벤트 조건의 개수를 줄이면, 최종적으로 원래의 이벤트 조건들과 병합된 이벤트 조건들로 이루어진 최적의 조합이 만들어지게 된다.

표 2의 조건 병합 대상을 선정하는 알고리즘의 입력 값은 이벤트 조건 테이블 P와 센서 노드가 수용할 수 있는 이벤트 조건의 개수인 m 이다. 우선 알고리즘의 결과값으로 보내줄 병합된 이벤트 조건 테이블 MP를 P로 초기화 시켜준다. 나머지 2~6의 반복문에서 한번씩

돌 때 마다 이벤트 조건 테이블 MP의 크기를 조건 병합을 통해서 하나씩 줄이면서 테이블의 크기가 노드의 용량 m 과 같아지면 종료한다. 5~15의 반복문에서는 현재 MP에서 속한 모든 서로 다른 이벤트 조건들의 쌍 (p_i, p_j) 에 대하여 조건 병합으로 인한 손실 비용 $PredicateMergeLossCost(MP, p_i, p_j)$ 을 구하여 그 중 최소값을 가진 조건 쌍을 선택하여 병합된 이벤트 조건 $MergedPredicate(p_i, p_j)$ 으로 교체한다.

표 2 조건 병합 대상을 선정하는 알고리즘

Predicate Merge Target Selection (P, m)	
P: predicate table	m: node capacity
01	MP ← P
02	while (MP.table_size > m)
03	minLossCost ← MAX_INT
04	minLossPair ← null
05	for each predicate p_i in MP
06	for each predicate p_j in MP
07	if ($p_i = p_j$) continue;
08	else
09	minLossCost ← MIN(PredicateMergeLossCost(MP, p_i, p_j), minLossCost)
10	minLossPair ← (p_i, p_j)
11	end if
12	end for
13	end for
14	MP.remove(minLossPair)
15	MP.add(MergedPredicate(p_i, p_j))
16	end while
17	return MP

그림 2의 예제와 같은 경우, 이벤트 조건 5개를 4개로 줄이기 위해서, 2~6의 반복문을 한번 돌면 된다. 5~15의 반복문에서는 이벤트 조건 5개로 만들어질 수 있는 20개의 서로 다른 이벤트 조건 쌍 (p_i, p_j) 에 대하여 조건 병합으로 인한 손실 비용을 구하여 그 중에 최소값을 보이는 조건 쌍을 찾아낸다. 센서 데이터 발생할 확률이 전체 정의역($attr_1, attr_2 \in [0, 100]$)에서 균일하다고 가정하였으므로, 이벤트 조건 4번과 5번의 쌍이 가장 작은 조건 병합으로 인한 손실 비용을 갖게 되어 병합 대상으로 선택된다.

P에서 속한 서로 다른 이벤트 조건들의 쌍 (p_i, p_j) 에 대하여 영역 $MBR(p_i, p_j)$ 내에서 조건 병합으로 인한 손실 비용을 구하는 알고리즘 PredicateMergeLossCost는 과 같다. 먼저, 1에서 조건 병합 전후의 메시지 전송 비용 beforeMergeCost, afterMergeCos를 초기화시킨 다음, 2~4에서 입력 값으로 받은 P, p_i, p_j 를 이용하여 $MBR(p_i, p_j)$ 내의 각각의 단위 영역에 대한 조건 병합 여부(unitMerged), 이벤트 조건들끼리 겹치는 정도(unitRegion-Overlap), 센서 데이터가 발생할 확률(unitSelectivity)을 배열로 가져와서 비용을 구하는 데 사용할 준비를 한다. 5~8의 반복문에서는 $MBR(p_i, p_j)$ 내의

각각의 단위 영역에 대하여 조건 병합 전후의 메시지 전송 비용을 앞서 설명한 예측 모델을 사용하여 구하여, 조건 병합 전후의 메시지 전송 비용 beforeMergeCost, afterMergeCost에 더한다. 마지막으로 9에서 조건 병합 전후의 비용의 차를 구하여 손실 비용 lossCost를 반환한다.

표 3 P에 속한 p_i 와 p_j 의 조건 병합 손실 비용을 계산하는 알고리즘

PredicateMergeLossCost(P, p_i, p_j)	
P:	predicate table p_i, p_j : predicate pair to merge
01	beforeMergeCost \leftarrow 0, afterMergeCost \leftarrow 0
02	unitMerged \leftarrow getMergedRegion(P, MBR(p_i, p_j))
03	unitRegionOverlap \leftarrow getRegionOverlap(P, MBR(p_i, p_j))
04	unitSelectivity \leftarrow getSelectivity(MBR(p_i, p_j))
05	for each unit region r_k in MBR(p_i, p_j)
06	afterMergeCost \leftarrow afterMergeCost + $d * C_{sa} * \text{unitSelectivity}[r_k.idx]$
07	beforeMergeCost \leftarrow beforeMergeCost +
	$d * C_{sa}(\text{unitMerged}[r_k.idx], \text{unitRegionOverlap}[r_k.idx]) * \text{unitSelectivity}[r_k.idx]$
08	end for
09	lossCost \leftarrow afterMergeCost - beforeMergeCost
10	return lossCost

그림 3의 예제에서 이벤트 조건 4번과 5번을 병합하여 병합된 이벤트 조건 6번을 만드는 경우, 이런 조건 병합으로 인한 손실 비용을 구하는 알고리즘 PredicateMergeLossCost는 5~8의 반복문에서는 4번과 5번을 병합되면 만들어질 이벤트 조건 6번 내의 각각의 단위 영역에 대하여 조건 병합 전후의 메시지 전송 비용을 예측 모델을 사용하여 구하여, 영역 전체의 조건 병합 전후의 메시지 전송 비용에 더한다. 최종적으로 구해진 영역 전체의 조건 병합 전후의 메시지 전송 비용의 차가 이벤트 조건 4번과 5번의 조건 병합으로 인한 손실 비용이 된다.

4. 실험 및 분석

4.1 실험 환경 및 데이터

실험에 사용된 시스템은 CPU Intel(R) Core(TM)2 6600 2.40 GHz이고 RAM은 2GB를 사용했다. 운영체제는 Windows XP, 개발 언어 및 플랫폼으로는 JAVA를 사용하였다.

본 논문에 제안 방법을 검증하기 위해 무선 센서 네트워크 시뮬레이션 환경을 구축하였다. 무선 센서 네트워크의 전체 센서 노드의 개수는 100개이며, 가로 축과 세로 축이 각각 10개씩 격자 형태로 배치되어 있고, 같은 축에서 붙어 있는 노드들 간의 거리는 10이며, 노드들 간에 서로 메시지를 주고 받을 수 있는 통신 거리는 15로 제한하였다. 사용자가 정의한 이벤트와 관련된 센서 노드의 측정 가능한 속성의 개수 2개이다. 사용자는

질의 결과로서 이벤트가 발생한 노드의 번호, 시각, 이벤트의 번호만 요청한다.

합성 데이터의 경우, JAVA의 난수 발생기를 이용하여 100개의 센서 노드의 2개의 속성에 대하여 0부터 100까지 정의역 내에서 평균과 표준 편차가 50, 10인 정규 분포로부터 10000개씩 값을 만들어냈다. 실제 데이터의 경우, 무선 센서 네트워크 실험에 자주 사용되는 Intel Lab Data[12] 중에서 온도과 습도 데이터를 사용하였다. Intel Lab Data의 54개의 센서 노드의 데이터 중에 샘플링 횟수가 적고, 이상한 데이터 분포를 보이는 3개의 센서 노드에 대한 데이터를 제외하였으며, 나머지 51개의 센서 노드의 데이터를 각각 10000개씩 추출해서 사용하였다. 이 중에서 정의역 내 단위 영역의 확률 분포를 알아내기 위한 히스토그램을 만들기 위해서 처음 1000개의 데이터를 사용하였다.

이벤트의 조건 데이터는 기본적으로 64개이며, 정의역 내에서 무작위하게 생성하였다. 실험에 따라서 센서 데이터와 조인 확률을 조절하기 위해, 이벤트 조건이 생성되는 지점의 범위를 센서 데이터의 분포 범위에 맞춰서 조절하였다. 예를 들어 센서 데이터와 조인 확률이 높은 이벤트 조건들은 센서 데이터의 분포 영역과 가까운 곳에서 주로 만들어지고, 낮은 이벤트 조건들은 센서 데이터의 분포 영역과 먼 곳에서 만들어질 수 있도록 이벤트 조건의 생성 지점을 조절하였다.

성능 비교의 기준이 되는 전체 센서 노드 간의 메시지 전송 비용은 비례 상수를 1로 하여 메시지의 크기에 비례하고, 노드 간의 전송 횟수에 비례하며, 메시지의 크기에 비례한다. 성능 실험의 비교 대상으로 앞서 설명한 중앙 집중식 방법과 REED[4]의 분산 조인 기반의 방법을 사용하였다. REED에서는 분산 조인의 성능을 높이기 위해서 센서 데이터가 사용자가 요청한 원래의 이벤트 조건들과 조인이 되는지를 알 수 있는 Bloom Filter를 사용하고 있다. Bloom Filter를 이용하여 사용자가 요청한 원래의 이벤트 조건과 조인되지 않아 주변 그룹 구성원들과 공유할 필요가 없는 센서 데이터 공유를 0.5의 확률로 막아주었다. 동등한 조건에서 실험하기 위해서 본 논문에서 제안한 조건 병합 기반의 방법에서도 베이스스테이션으로 보낼 필요가 없는 센서 데이터를 Bloom Filter를 이용하여 0.5의 확률로 막아주었다.

4.2 실험 결과 및 분석

먼저 문제 해결을 위하여 제안한 메시지 전송 비용에 대한 비용 모델을 검증하기 위한 실험을 하였다. 센서 데이터 고정시키고 이벤트의 조건들에 대한 조인 확률을 변화시켜 가면서, 같은 실험 조건에서 메시지 전송 비용 모델에 의한 전체 센서 네트워크의 메시지 전송 비용의 예측 값 C_e 와 실제 시뮬레이션 의해 나온 전체

센서 네트워크의 메시지 전송 비용 C_r 를 비교하였다. 다양한 이벤트 조건들에 대한 조인 확률에 대하여 280번의 실험 결과, $|(C_e - C_r)/C_r|$ 값의 평균과 표준 편차가 각각 0.00728과 0.0072가 나왔다. 이는 본 논문에서 제안한 메시지 전송 비용 예측 모델이 타당함을 실험적으로 증명해준다.

다음은 이벤트 조건의 개수가 고정된 상태에서 노드의 용량 변화에 따른 성능 변화를 알아보기 위한 실험을 하였다. 이벤트의 조건의 개수를 64개로 고정시키고 노드가 수용할 수 있는 조건의 용량을 4개부터 64개까지 늘려보았다. 이벤트 조건들에 대한 센서 데이터의 조인 확률에 따라서 낮은 확률, 중간 확률, 높은 확률 세 부류로 나누어서 각각의 부류의 노드당 주기당 평균 메시지 전송 비용의 평균을 구하여 결과를 그림 6, 그림 7, 그림 8에서 정리하였다. 각 부류의 조인 확률의 평균은 각각 0.13, 0.55, 0.94이다. 범례의 Centralized는 중앙 집중식 방법, REED는 REED의 그룹 기반의 분산 조인 방법, MERGE는 조건 병합 기반의 조인 방법을 가리킨다.

이벤트 조건을 저장할 수 있는 노드 용량이 증가함에 따라 소모되는 메시지 전송 비용이 줄어들음을 알 수 있다. 그리고 조건 병합 기반의 조인 방법이 전체적으로 메시지 전송 비용이 작다. 이는 노드 용량에 상관없이 본 논문에서 제안한 조건 병합 대상을 선택하여 전체 이벤트 조건의 개수를 줄이는 효과가 있음을 보여주고 있다. REED의 그룹 기반 분산 조인 방법의 경우, 노드 간의 통신 거리의 한계 때문에 노드 용량이 4, 8, 12 일 때 각각 크기 16, 8, 6의 그룹을 형성 하지 못 하고 Bloom Filter 기능이 추가된 중앙 집중식 방법을 사용하고 있다. 그룹 형성이 가능해진 16개보다 큰 용량에서도 조건 병합 기반의 조인 방법보다 성능이 떨어지는 것을 알 수 있다. 특히, REED의 그룹 기반의 분산 조인 방법은 센서 데이터의 조인 확률이 높은 경우 중앙 집중식 방법보다도 성능이 떨어지는 반면에, 조건 병합 기반의 조인 방법은 꾸준히 좋은 성능을 유지하고 있다. 센서 데이터의 조인 확률이 높은 경우, 대부분의 경우에 베이스스테이션으로 조인 결과나 센서 데이터를 보내야 하기 때문에 REED의 그룹 구성원들끼리 하는 센서 데이터의 공유가 효용성이 없어지고 성능이 떨어질 수 밖에 없다. 하지만 조건 병합 기반의 조인 방법은 필요한 조인 확률이 높아지더라도 병합 시에 예상되는 손실 비용이 큰 이벤트 조건들의 조합을 피해서 병합 대상을 선택하였기 때문에 영향을 덜 받을 수 있다.

다음은 무선 센서 네트워크를 구성하는 센서 노드들의 베이스스테이션까지의 평균 거리, 즉 깊이에 따른 성능 변화를 알아보기 위한 실험을 하였다. 열의 개수를 2

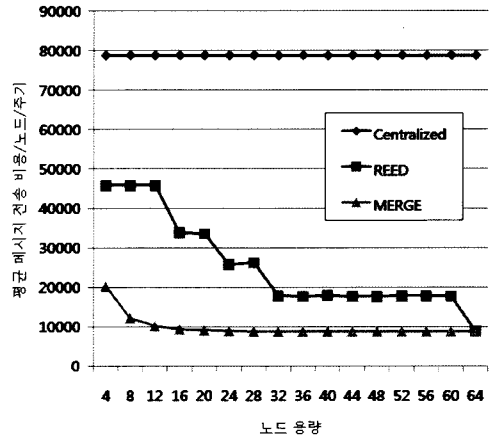


그림 6 합성 데이터를 이용한 노드 용량에 따른 성능 변화: 낮은 조인 확률(0.13)

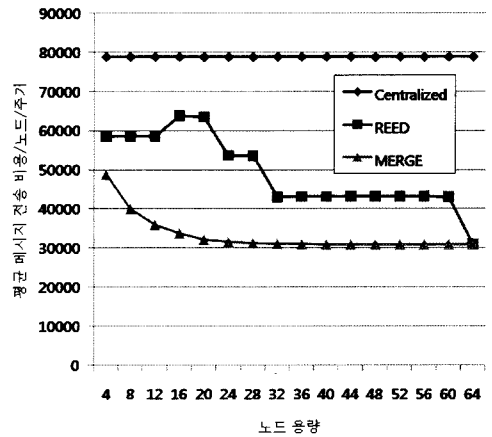


그림 7 합성 데이터를 이용한 노드 용량에 따른 성능 변화: 보통 조인 확률(0.55)

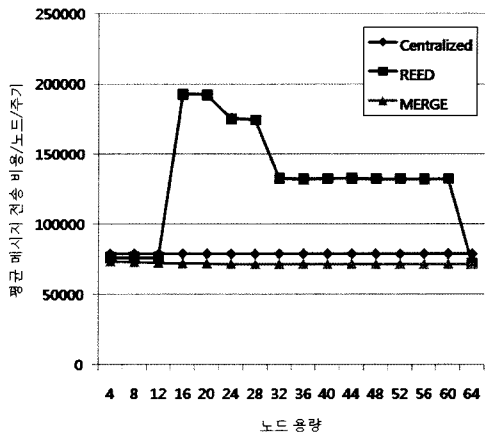


그림 8 합성 데이터를 이용한 노드 용량에 따른 성능 변화: 높은 조인 확률(0.94)

개로 고정시키고 행의 개수를 10부터 100개까지 10씩 증가시켜 보았다. 이벤트 조건들에 대한 센서 데이터의 조인 확률에 따라서 낮은 확률, 중간 확률, 높은 확률 세 부류로 나누어서 각각의 부류의 노드당 주기당 평균 메시지 전송 비용과 중앙 집중식 방법의 전송 비용을 기준으로 한 상대적 비율을 구하여 결과를 그림 9, 그림 10, 그림 11에서 정리하였다. 각 부류의 조인 확률의 평균은 각각 0.16, 0.55, 0.85이다.

센서 노드의 평균 깊이가 늘어남에 따라 노드당 주기

당 평균 메시지 전송 비용이 늘어나고 있다. 이는 노드의 깊이가 커짐에 따라 베이스스테이션으로 보내는 메시지에 들어가는 전송 비용이 증가하기 때문이다. 중앙 집중식 방법과 달리 조건 병합 기반의 조인 방법과 REED의 그룹 기반의 분산 조인 방법은 노드의 깊이가 커지더라도, 인-네트워크 조인을 통해서 베이스스테이션을 보낼 필요가 없는 센서 데이터의 메시지를 많이 줄이고 있어서, 영향을 덜 받고 있다. 비율 그래프를 살펴 보면 조건 병합 기반의 조인 방법은 노드의 깊이와 상

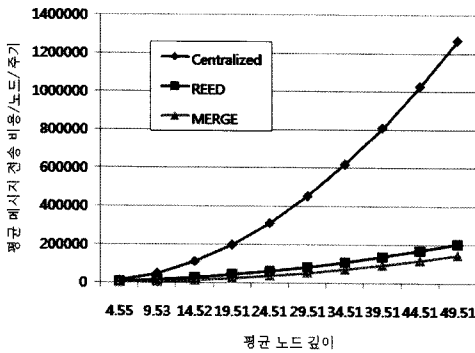


그림 9 노드의 깊이에 따른 성능 변화: 낮은 조인 확률(0.16)

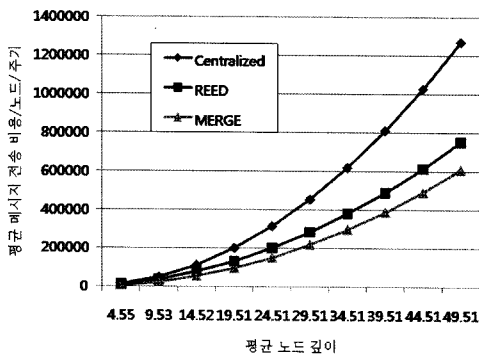
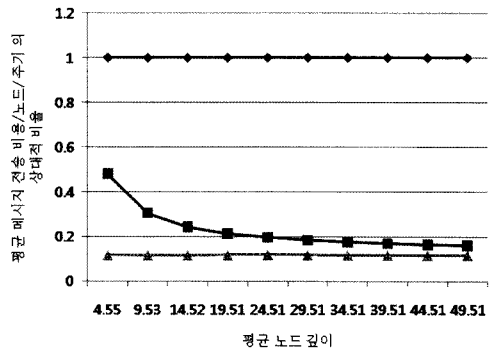


그림 10 노드의 깊이에 따른 성능 변화: 보통 조인 확률(0.55)

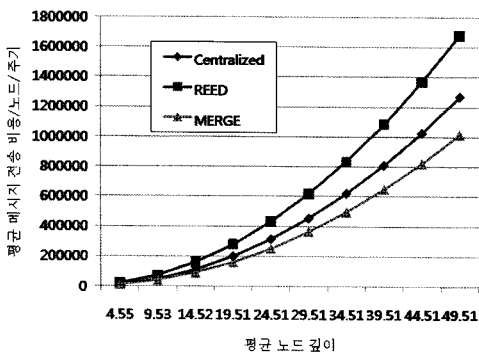
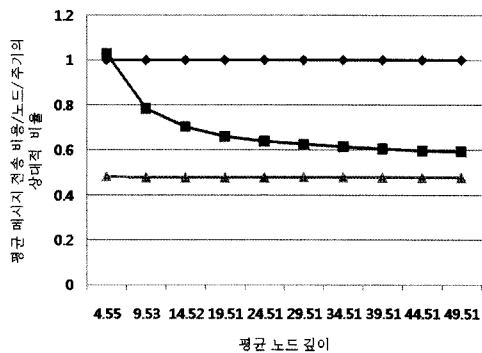
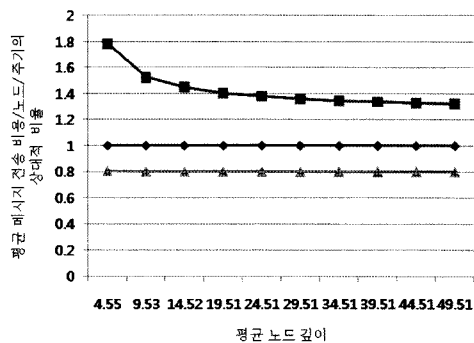


그림 11 노드의 깊이에 따른 성능 변화: 높은 조인 확률(0.85)



관 없이 중앙 집중식 방법과의 상대적 비율을 일정하게 유지하면서 메시지 전송 비용은 고르게 작다. 하지만 REED의 그룹 기반의 분산 조인 방법은 노드의 평균 깊이가 작은 경우, 중앙 집중식 방법과의 상대적 비율이 조건 병합 기반의 조인 방법보다 높게 나오고 있다. 노드의 평균 깊이가 작은 경우, REED의 그룹 기반의 분산 조인 방법은 그룹 구성원들과 센서 데이터를 공유하여 분산 조인을 하는데 들어가는 비용이 다른 방법에 비해 상대적으로 크기 때문이다.

다음은 실제 센서 데이터인 Intel Lab Data를 이용한 실험들을 소개할 것이다. 먼저 이벤트 조건의 개수가 고정된 상태에서 노드의 용량 변화에 따른 성능 변화를 알아보기 위한 실험을 하였다. 실험 방식은 앞선 합성 데이터 실험과 동일하다. 낮은 조인 확률(0.10), 중간 조인 확률(0.49), 높은 조인 확률(0.94) 세 부류로 나누어 정리한 노드당 주기당 평균 평균 메시지 전송 비용의 결과는 그림 12, 그림 13, 그림 14와 같다.

앞서 언급한 합성 데이터 실험과 비슷하게 조건 병합 기반의 조인 방법이 전체적으로 메시지 전송 비용이 작다. 합성 데이터 실험과 마찬가지로 이유로 조건 병합 기반의 방법이 더 나은 성능을 보인다고 할 수 있다. 특히, 이번 실험에서 REED의 그룹 기반의 분산 조인 방

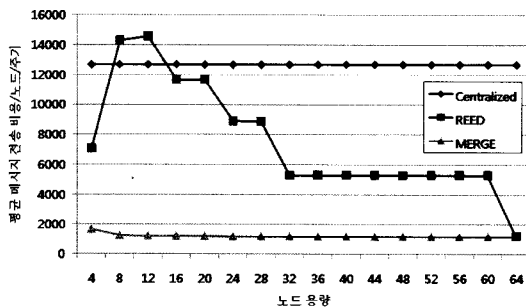


그림 12 실제 데이터를 이용한 노드 용량 변화에 따른 성능 변화: 낮은 조인 확률(0.10)

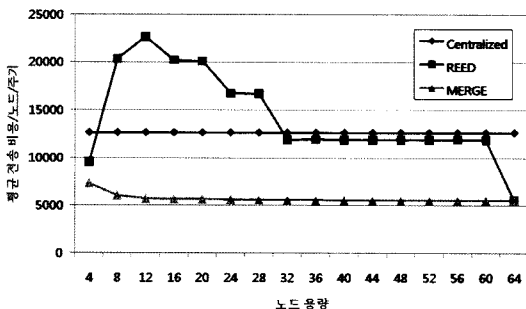


그림 13 실제 데이터를 이용한 노드 용량 변화에 따른 성능 변화: 중간 조인 확률(0.49)

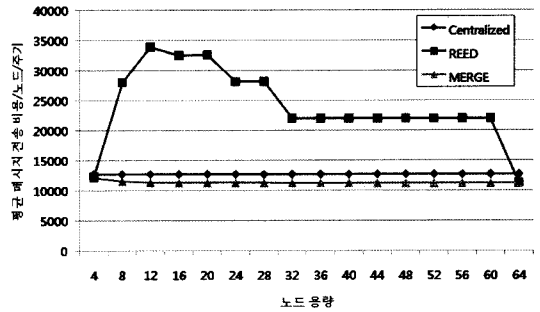


그림 14 실제 데이터를 이용한 노드 용량 변화에 따른 성능 변화: 높은 조인 확률(0.94)

법은 보통 조인 확률에서도 중앙 집중식 방법도 좋지 않은 성능을 보이고 있다. 이것은 Intel Lab Data를 구성하는 센서 노드의 개수가 작고, 센서 노드의 평균 깊이(1.94)가 크지 않은 것에 영향을 받을 거라 할 수 있다. 이는 앞서 언급한 합성 데이터를 이용한 센서 노드의 깊이에 따른 성능 변화 실험 결과와도 일치한다.

다음은 센서 노드의 용량이 고정된 상태에서 이벤트 조건의 개수에 따른 성능 변화를 알아보기 위한 실험을 하였다. 센서 노드의 이벤트 조건 저장 용량은 16개로 고정시키고 이벤트 조건의 개수를 8개에서 256개까지 8개씩 증가시켜 보았다. 이벤트 조건의 개수가 늘어남에 따라서 센서 데이터의 노드당 주기당 평균 조인 확률은 늘어난다. 그로 인해 메시지 전송 비용도 늘어나게 되는데, 그림 15와 같은 결과가 나온다.

조건 병합 기반의 방법은 이벤트 조건이 늘어나더라도, 완만하게 평균 메시지 전송 비용이 늘어나는 반면에, REED의 분산 조인 방법은 매우 급격하게 오르고 있다. 이벤트 조건이 늘어나더라도 조건 병합 기반의 방법은 메시지 전송을 비용을 최적으로 줄일 수 있는 이벤트 조건들의 병합을 통해서 전체 이벤트 크기를 줄이기 때문에 메시지 전송 비용의 증가량이 크지 않다. 반면에 REED의 분산 조인 방법은 이벤트 조건을 공유해야 할 그룹의 크기가 커짐에 따라서, 그룹 노드들과의 통신 비용이 늘어나게 된다. 또한 조건을 공유해야 할 그룹의

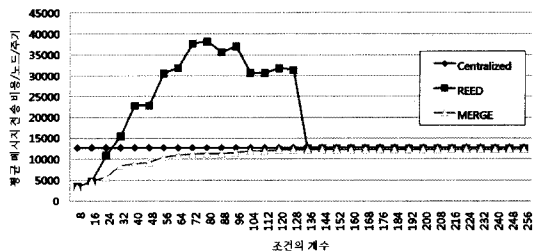


그림 15 이벤트 조건의 개수에 따른 성능의 변화

크기가 커짐에 따라서 이웃 노드들과 그룹을 형성할 확률이 줄어들어서 중앙 집중식 방법처럼 센서 데이터를 베이스스테이션으로 보내야 하는 노드들이 늘어나게 되어 조인 결과 메시지 전송 비용이 늘어나게 된다.

5. 결론

본 논문에서는 무선 센서 네트워크의 다중 이벤트 탐지 환경에서 센서 노드의 제한적인 저장 용량을 극복하고 통신 비용을 절약하기 위해, 사용자의 요청한 다수의 이벤트 조건들을 이벤트 조건 병합을 통해 줄여서 인-네트워크 조인을 하는 방법을 제안하였다.

이벤트 탐지 결과를 베이스스테이션으로 보내기 위해서 필요한 메시지 전송 비용에 대한 예측 모델을 만들어서 조건 병합의 대상이 되는 이벤트 조건들을 선택할 때 활용하여 조건 병합으로 인한 메시지 전송 비용의 증가를 최소화 할 수 있는 방법을 제안하였다.

메시지 전송 비용에 대한 예측 모델의 타당성을 실제 시뮬레이션에 통해 얻어진 비용과 비교를 통해 검증하였다. 그리고 제안한 방법의 성능 평가를 위하여 인위적 랜덤 데이터와 실제 데이터를 이용하여 기존의 연구와 다양한 비교 실험을 하였으며, 이를 통하여 본 논문에서 제안한 방법의 효율성을 입증하였다.

참고 문헌

[1] K. Romer and F. Mattern, "The design space of wireless sensor networks," *Wireless Communications, IEEE*, vol.11, no.6, pp.54-61, Dec. 2004.

[2] Salem Hadim and Nader Mohamed, "Middleware: Middleware challenges and approaches for wireless sensor networks," *IEEE Distributed Systems Online*, vol.7, no.3, Mar. 2006.

[3] Yong Yao and Johannes Gehrke, "Query processing in sensor networks," *Proc. Of CIDR*, 2003.

[4] Daniel J. Abadi, Samuel Madden, and Wolfgang Lindner, "REED: Robust, efficient filtering and event detection in sensor networks," *Proc. Of VLDB Conference*, pp.769-780, 2005.

[5] Samuel Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong, "TAG: A tiny aggregation service for ad-hoc sensor networks," *Proc. of OSDI*, pp.131-145, 2002.

[6] Samuel Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong, "TinyDB: an acquisitional query processing system for sensor networks," *ACM Trans. Database Syst.*, vol.30, no.1, pp.122-173, 2005.

[7] Shuoqi Li, Sang Hyuk Son, and John A. Stankovic. "Event detection services using data service middleware in distributed sensor networks," *Proc of IPSN*, pp.502-517, 2003.

[8] Wenwei Xue, Qiong Luo, Lei Chen 0002, and Yunhao Liu, "Contour map matching for event detection in sensor networks," *Proc of SIGMOD Conference*, pp.145-156, 2006.

[9] Xiaoyan Yang, Hock-Beng Lim, M. Tamer Özsu, and Kian-Lee Tan, "In-network execution of monitoring queries in sensor networks," *Proc of SIGMOD Conference*, pp.521-532, 2007.

[10] Burton H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Commun. ACM*, vo.13, no.7, pp.422-426, 1970.

[11] Philip A. Bernstein and Dah-Ming W. Chiu, "Using semi-joins to solve relational queries," *J. ACM*, vol.28, no.1, pp.25-40, 1981.

[12] Samuel Madden, Intel lab data, <http://db.csail.mit.edu/labdata/labdata.html>.



양 동 윤
2006년 한국과학기술원 전자전산학과 전산학전공 학사. 2009년 한국과학기술원 전자전산학부 전산학전공 석사. 관심분야는 스트림 데이터 및 센서 네트워크 데이터베이스

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