

컨텍스트 퓨전을 통한 센서 네트워크에서의 시간 제약 컨텍스트 탐지 기법

안성진^o 김대영
한국정보통신대학교
{sungjin^o, kimd}@icu.ac.kr

Time Bounded Context Detection using Context Fusion
in Wireless Sensor Networks

Sungjin Ahn^o Daeyoung Kim
Information and Communications University

요 약

본 논문은 시간 제약 조건을 갖는 컨텍스트(상황)에 대한 분산적 탐지 기법(PROCON)을 제안한다. 수집한 데이터를 베이스 스테이션으로 모아 컨텍스트의 발생 여부를 결정하는 기존의 중앙집중 기법과 달리 PROCON은 네트워크 내에서 능동적, 분산적 방식으로 센서 노드 간의 협업을 통하여 컨텍스트를 결정한다. 따라서, 컨텍스트의 발생여부 탐지와 이에 대한 대응(Actuation)이 베이스 스테이션에 독립적으로 네트워크 내에서 모두 이루어진다. 본 논문은 다양한 상황구성에 대한 시뮬레이션을 통해 PROCON의 분산 상황방식이 기존의 중앙집중 방식에 비해 상당한 에너지 절약과 서비스 신뢰성(reliability) 향상을 가져온다는 것을 보인다.

1. 서 론

넓은 지역에 대해 비교적 적은 비용으로 다양한 환경정보를 제공하는 센서 네트워크는 컨텍스트(상황) 인지를 위한 핵심 기반 기술로서 주목 받아 왔다. 하지만 센서 노드가 저수준의 센싱 정보를 제공하는 기능에 국한 된 기존의 중앙집중식 상황 결정 방식[1,2,3]은 하나의 상황이 서로 다른 발생 빈도, 목표 지역, 그리고 시간 제약 조건 등을 가지는 여러 개의 "이벤트"들로 구성 되는 "조합 컨텍스트" 일 경우에 많은 단점을 나타낸다.

첫째, 중앙집중식 상황인지 기법은 상황 조건을 만족시키지 못하는 저수준 센싱 정보를 베이스 스테이션으로 전송하는데 많은 에너지를 소모한다. 이벤트 간의 발생 빈도와 시간 제약 조건의 차이로 인해 발생하는 이 전송 에너지 낭비는 센서 노드의 수명을 단축시키고, 결과적으로 상황인지 서비스의 수명을 단축시킨다.

중앙집중 상황인지 기법의 또 다른 단점은 베이스 스테이션 주위의 노드들의 수명이 다른 노드들에 비해 짧다는 것이다. 센서 네트워크의 전형적인 통신 모델인 다대일(many-to-one) 통신 특성에 의해 발생하는 이 문제는 결과적으로 베이스스테이션을 네트워크로부터 고립시킴으로써 더 이상의 서비스를 불가능하게 한다. 더욱이, 건강상태 진단이나 화재 탐지 등의 중요 응용이 서비스 되고 있는 경우, 문제가 더욱 심각해진다.

이러한 문제점을 해결하기 위해 본 논문은 센서 네트워크를 이용하여 능동적이고 분산적인 방법으로 상황을 탐지하는 기법인 PROCON을 제안한다. PROCON은 노드의 네트워크 계층 위에서 상황을 관리하는 일종의 미들웨어로써 *컨텍스트 오버레이*로 연결된 센서 노드들 간의 협업을 통해 분산적 상황 결정을 통해 상황을 탐지한다. 센서 노드들은 이 컨텍스트 오버레이를 따라 다음 "이벤트 노드"에게 이벤트 알림 메시지를 보낸다. *컨텍스트 퓨전*이라고 하는 이 분산 상황 결정 기법은 여러 이벤트 정보를 융합하여 하나의 부분 컨텍스트를 만들어 낸다. 결과적으로, PROCON은 컨텍스트 퓨전을 통해 불필요한 이벤트

알림 메시지를 줄여 전체적으로 센서 노드의 에너지 소비를 최소화 한다. 또한, PROCON의 "이벤트 노드"들간 통신 모델은 베이스 스테이션으로의 집중적 데이터 전송을 피하므로 베이스 스테이션 주변 노드의 짧은 수명 문제를 해결한다. 그림 1은 PROCON의 전체 구조를 보여준다.

본 논문은 다음과 같이 구성된다. 먼저 관련연구에 대해 다음 절에서 살펴보고, PROCON 컨텍스트 모델을 3절에서 소개한다. 4절은 PROCON의 주요 설계 사항을 기술하고, 5장의 성능 평가를 통해 PROCON의 성능을 검증한다. 마지막으로 6장을 결론으로 본 논문을 맺는다.

2. 관련연구

베이스 스테이션에서 쿼리를 보내고 이에 대한 응답을 받는 구조[1,2,3]는 중앙집중식 상황 탐지에서 가장 많이 사용하는 방식이다. Directed Diffusion [1]은 목표 이벤트에 대한 요구사항을 "interest" 라는 패킷 형태로 네트워크에 플러딩하고 발생한 이벤트 정보를 베이스 스테이션으로 수집하는 방식이다. 베이스 스테이션으로의 전송 과정에서는 데이터융합(data aggregation)을 통해 전송되는 데이터의 양을 줄인다. TinyDB[2]와 Cougar[3]는 이러한 데이터융합 특성을 데이터 베이스 시스템에 적용한 예이다. 하지만 이들 방식은 평균 온도를 얻는 것과 같은 동일 타입 데이터에는 데이터 융합을 통해 전송 데이터의 양을 줄이지만 시간 제약을 가지는 여러 타입의 이벤트로 이루어지는 조합 컨텍스트에 대해서는 연구가 이루어지지 않았다.

센서 네트워크를 이용하여 조합 컨텍스트와 유사한 컨텍스트를 탐지하기 위한 기법이 [4]와 [5]에서 제안되었다. Trailblazer는 Peer-to-Peer 네트워크와 GHT[6]의 컨셉에 기반한 방법을 [5]는 "스마트 상황인지 패킷"을 주입하고 이 패킷이 주변 상황 정보를 모아오는 방법을 제안 하였다.

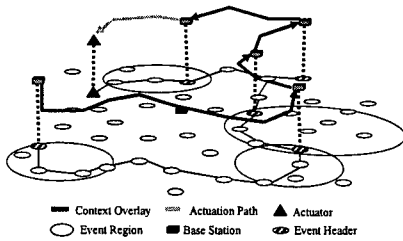


그림 1. PROCON 구조

3. 컨텍스트 모델

조합 컨텍스트는 논리적, 시간적 제약 조건을 가지는 둘 이상의 컴포넌트 이벤트의 집합으로 정의한다. 컴포넌트 이벤트는 목표 위치와 발생 조건을 기본으로 정의되는 이벤트로서 다른 컴포넌트 이벤트와의 논리 및 시간 조건을 만족시키며 발생할 경우 대응하는 조합 컨텍스트를 발생 시킨다. 예를 들어, 조합 컨텍스트 C가 1) 방의 온도가 25도 이상이고, 2) 한 사람 이상이 방에 있을 경우 발생하는 것으로 정의 될 경우, 25도 이상의 온도와 사람의 존재여부가 각각 컴포넌트 이벤트가 되고 두 이벤트는 정해진 시간간격을 만족하며 발생해야 한다.

하나의 컴포넌트 이벤트가 발생 조건을 만족시키고 있는 상태를 "TRUE 상태에 있다"고 하고, 그 반대의 경우에는 "FALSE 상태에 있다"고 정의한다. 따라서, 조합 컨텍스트의 논리 제약 조건은 컴포넌트 이벤트를 간의 Boolean 조합으로 정의한다. PROCON은 기본적으로 AND와 OR 두 개의 논리 연산자를 정의한다. 따라서, 컴포넌트 이벤트 간의 Boolean 연산이 최종적으로 TRUE가 될 때 대응하는 조합 컨텍스트가 발생한다.

유닛 컨텍스트는 AND 논리 연산자만을 갖는 조합 컨텍스트이다. Boolean 연산을 통해 모든 논리 조합은 유닛 컨텍스트들의 product 형태로 바꾸어 질 수 있다. 유닛 컨텍스트의 발생은 해당 유닛 컨텍스트에 포함되지 않는 다른 컴포넌트 이벤트의 상태에 상관없이 컨텍스트를 발생시키므로 다음의 알고리즘 들은 유닛 컨텍스트의 탐지를 목표로 한다.

두 컴포넌트 이벤트 E_i 와 E_j 간의 시간 조건 $T_{i \rightarrow j} = (L_{i \rightarrow j}, U_{i \rightarrow j})$ 는 E_j 가 시간 t 에 발생했을 경우 E_i 는 시간 구간 $(t_j + L_{i \rightarrow j}, t_j + U_{i \rightarrow j})$ 사이에 발생했을 경우에만 시간 조건을 충족시킨다는 것을 의미한다. 조합 컨텍스트가 발생하기 위해서는 모든 컴포넌트 이벤트들간에 정의된 시간 조건이 충족되어야 한다.

4. PROCON

4.1 컨텍스트 디스크립터

임의의 상황이 발생했을 때, 자동으로 원하는 액추에이션(actuation)을 하고자 하는 사용자는 PDA, 노트북과 같은 인터페이스 디바이스에서 제공하는 센서 네트워크 프로파일과 사용자 정의를 통해 컨텍스트 디스크립터를 만든다. 각각의 컴포넌트 이벤트에 대해 하나의 컨텍스트 디스크립터가 만들어지며, 이는 이벤트 id, 컨텍스트 id, 유닛 컨텍스트 id, 이벤트간 시간 조건, 다음 컴포넌트 이벤트, 이벤트 발생 조건 등, 이벤트와 조합 컨텍스트의 탐지에 필요한 모든 정보를 포함한다. 또한 액추에이터의 위치와 아이디 등의 정보도 담고 있어, 컨텍스트가 발생할 경우 바로 액추에이션 트리거 패킷을 액추에이터로 보내 대응동작을 실행할 수 있다. 이를 인터넷워크 액추에이션이라 한다. 그림2는 컨텍스트 디스크립터의 예이다.

```
Context Descriptor {
  cid: 1
  ucid: 2
  Event {
    eid: 1
    type: temperature
    condition: 10 °C
    area: [(-50,-30), (-10,10)]
    timing: 2, (-5 sec, +5 sec)
    next: 3, [(10,-10), (30,-40)]
  }
  Action {
    id: 4
    location: [(-50,20), (-40,10)]
  }
}
```

그림 2. 컨텍스트 디스크립터 예

4.2 컨텍스트 오버레이 구성

컨텍스트 오버레이는 컴포넌트 이벤트의 헤더 노드들을 연결한 체인으로써 컨텍스트 오버레이를 따라 이벤트 알림 패킷을 다음 이벤트 헤더 노드로 전송한다. 따라서 컨텍스트 오버레이를 구성하는 것은 오버레이 체인에서 각 이벤트가 위치하는 순서를 결정하는 것이다. 인터페이스 디바이스는 상황에 맞게 아래에 나열한 순서 결정 알고리즘 중 하나를 이용해 각 컴포넌트 이벤트의 순서를 결정하고 이 정보를 컨텍스트 디스크립터에 기술하여 각 이벤트 헤더에게 전달한다. 컨텍스트 디스크립터의 전달은 Geocasting을 통해 이루어진다.

이벤트 발생 빈도 기반 알고리즘은 낮은 발생률을 갖는 이벤트로부터 높은 발생률을 갖는 이벤트 순으로 컨텍스트 오버레이를 구성한다. 이 알고리즘은 이벤트 헤더들 간의 거리 차이가 적고 이벤트 발생률의 차이가 클수록 효율적인 컨텍스트 오버레이를 구성한다. 하지만, 컨텍스트 디스크립터를 만드는 시점에 이벤트의 발생 빈도를 예측할 수 없는 경우에는 적용이 불가능하다는 것이 이 알고리즘의 제약사항이다.

거리 기반 알고리즘은 이벤트의 발생빈도를 예측할 수 없는 경우에 사용할 수 있는 대안으로써 이벤트 발생빈도가 모두 같다 (λ)는 가정하에 이벤트 헤더들 간의 거리를 이용해 컨텍스트 오버레이를 구성하는 기법이다. 이 기법은 아래의 목적 함수를 최소화 하는 이벤트 순서를 구하는 최적화 문제로서 선형계획법 등을 이용하여 푼다. 여기서, $P = 1 - e^{-\lambda}$ 이고, D_i 는 i 번째 이벤트와 $i+1$ 번째 이벤트간의 거리이다.

$$F(n) = \lambda P D_1 + \lambda P^2 D_2 + \dots + \lambda P^{n-1} D_{n-1}.$$

하이브리드 최적 알고리즘은 이벤트 발생 빈도와 이벤트간의 거리를 모두 고려하여 최적의 컨텍스트 오버레이를 구성하는 알고리즘으로 다음 목적 함수를 최소화하는 순서를 구한다.

$$F(n) = \sum_{i=1}^{n-1} r_i D_i.$$

이때, r_i 는 1번째부터 i 번째 이벤트가 모두 시간 조건을 만족하며 발생하는 빈도이다.

4.3 컨텍스트 퓨전을 통한 분산적 상황 결정

오직 컨텍스트 오버레이의 첫번째 이벤트가 발생했을 때에만 이벤트 알림 패킷을 다음 이벤트로 전달함으로써 하나의 상황

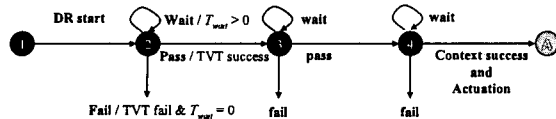


그림 3 컨텍스트 퓨전

결정 프로세스를 시작할 수 있다. 다른 이벤트들은 전달받은 현재 상황 결정 프로세스를 진행시킬 것인지, 종료할 것인지, 또는 일정 시간 동안 대기 시킬 것인지를 결정한다. 이 결정은 시간 제약조건 테스트의 결과에 따라 달라지는데, 현재의 이벤트와 이전 이벤트들 간의 시간 제약 조건이 모두 만족할 경우 현재까지의 이벤트 정보를 융합하여 다음 이벤트로 이벤트 알림 패킷을 전달함으로써 현재 상황 결정 프로세스를 한 단계 더 진행시키고 이를 컨텍스트 퓨전이라 한다. 만약 시간 제약 조건을 만족하지 않을 경우는 종료하거나 대기한다.

종료와 대기에 대한 결정은 두 이벤트간의 시간 제약 조건에 따라 달라진다. 대기할 경우에는 주어진 대기 시간 동안 이벤트가 발생하여 시간 제약조건을 만족시킬 경우 상황 결정 프로세스를 한 단계 진행시키고, 대기 시간이 다 지나도록 시간 제약 조건이 만족되지 않을 경우 현재 상황 결정 프로세스를 종료시키고 다음 상황 결정 프로세스가 진행되기를 기다린다. 만약 하나의 상황 결정 프로세스가 마지막 이벤트까지 성공하면 컨텍스트가 발생한다. 그림 2는 위의 과정을 나타낸다.

5. 성능평가

5.1 시뮬레이션 모델

50m X 50m의 센서 필드에 한 개의 베이스스테이션을 포함한 총 100개의 센서 노드에 대하여 시뮬레이션 하였다. 베이스스테이션은 정사각형 센서 필드의 한 구석에 위치시키고 다른 노드들은 유니폼 랜덤으로 배치하였다. 센서 노드 스택은 버클리대의 MICA2를 사용하였다.

중앙집중 방식의 라우팅 프로토콜로 최소홉카운트 라우팅을 사용한다. 테이블 기반의 이 라우팅 프로토콜은 테이블 갱신을 위해 주기적으로 약간의 에너지를 소모한다. 하나의 이벤트에 대한 데이터융합을 고려하지 않는다. PROCON의 라우팅으로는 AODV[7]를 사용한다. 즉, 각 컴포넌트 이벤트 헤더는 다음 이벤트 헤더로 이벤트 알림 패킷을 보낼 때 AODV를 사용한다.

조합 컨텍스트를 구성하는 컴포넌트 이벤트의 개수 (|C|) 변화, 컴포넌트 이벤트간의 발생률의 차 (R), 그리고 베이스스테이션을 기준으로 한 센서 노드의 위치에 따른 에너지 소모 패턴을 시뮬레이션 목표로 삼는다.

5.2 컴포넌트 이벤트 개수와 이벤트 발생률의 차의 영향

그림 4(a)와 4(b)는 각각 2개 8개의 컴포넌트 이벤트로 구성된 컨텍스트의 이벤트 발생빈도 변화에 따른 평균 에너지 소모에 대한 결과를 보여준다. 그림 4(a)에서 보듯이 PROCON이 중앙 집중 방식에 비해 월등한 에너지 절약 효과를 보인다. 또한 PROCON과 중앙집중방식 모두 이벤트 발생률의 차가 클수록 에너지 소모가 줄지만 PROCON이 더욱 큰 폭의 성능향상을 보이는 반면 중앙집중방식의 성능향상은 미약하다.

그림 4(b)에서 보듯이 컴포넌트 이벤트의 개수가 증가하면, 중앙집중 결정 방식은 이벤트 발생률의 차가 적을수록 에너지 소모가 더욱 커진다. 반면 PROCON은 컴포넌트 이벤트의 개수가 에너지 소모에 큰 영향을 끼치지 않는다.

5.3 베이스 스테이션의 위치를 기준으로 한 노드 에너지 소모

그림 5(a)와 5(b)는 각각 3000 단위시간 18000 단위시간이 지난 후 중앙집중 방식과 PROCON의 잔여 노드 에너지를 보여

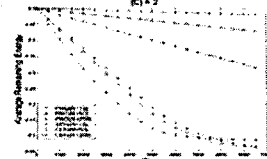


그림 4 (a) . |C| = 2

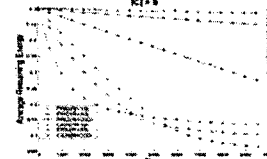


그림 4 (b) . |C| = 8

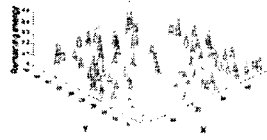


그림 5 (a) . 중앙집중식



그림 5 (b) . PROCON

준다. 뿔의 높이가 높을수록 많은 에너지를 나타낸다. 그림 5(a)에서 보듯이 중앙집중 결정 방식에서는 좌표 (0, 0)에 위치한 베이스 스테이션 주위에 위치한 노드들이 다른 노드들 보다 먼저 죽고 따라서 베이스 스테이션이 네트워크로부터 고립된다. 반면, 그림 5(b)에서 보듯이 PROCON은 특정 지역에 치우침 없이 균일하게 노드가 에너지를 소모하는 것을 볼 수 있다. 더욱이 절대적인 에너지 소모량 또한 중앙집중방식에 비해 극히 적은 것을 볼 수 있다.

6. 결론

본 논문은 센서 네트워크에서 분산적, 능동적 상황인지 기법인 PROCON을 제안하였다. 컨텍스트 오버레이를 통해 이벤트 발생 정보를 전달하고 컨텍스트 퓨전을 통해 분산적 상황 결정을 하는 이 기법은 노드간 협력을 통해 베이스 스테이션에 독립적으로 네트워크 내에서 상황을 탐지하고 이에 대응하는 액추에이션을 가능케한다. 이 기법이 기존의 중앙집중 방식의 컨텍스트 탐지 기법에 비해, 시간적 제약 조건을 가지는 여러 개의 이벤트로 구성된 컨텍스트의 탐지에 대해서 매우 적은 에너지 소모를 보이는 것을 시뮬레이션을 통해 보였다.

PROCON은 센서 네트워크가 저수준의 센싱 데이터를 베이스 스테이션으로 전달하는 단순한 데이터 전송 네트워크에서 더욱 고수준의 작업을 노드가 수행하는 분산 시스템으로 진화하는 좋은 예가 될 것이다.

참고문헌

[1] C. Intanagonwiwat, R. Govindan, and D. Estrin. *Directed diffusion: A scalable and robust communication paradigm for sensor networks*. In ACM MOBICOM, 2000.
 [2] S. Madden, et al. *TinyDB: An acquisitional query processing system for sensor networks*. In Transactions on Database Systems (TODS), 2005.
 [3] P. Bonnet, et al. *Towards Sensor Database Systems*. In MDM, 2001.
 [4] J.S. Hou, H.C. Hsiao, C.T. King, and C.N. Lu. *Context Discovery in Sensor Networks*. In IEEE ITRE, 2005.
 [5] F. Michahelles, M. Samulowitz, and B. Schiele. *Detecting Context in Distributed Sensor Networks by Using Smart Context-Aware Packets*. In ARCS, 2002.
 [6] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker. *GHT: A Geographic Hash Table for Data-Centric Storage*. In ACM WSN 2002.
 [7] C. Perkins, E. Belding-Royer, and S. Das. *Ad hoc On-Demand Distance Vector Routing*. In IETF RFC 3561, 2003.