

센서네트워크에서 다수의 이동 노드 추적 기법

이성민[○] 차호정
연세대학교 컴퓨터과학과
{sulee[○], hjcha}@cs.yonsei.ac.kr

Locating Method for Multiple Mobile Nodes in Wireless Sensor Networks

Sung-Min Lee[○] Hojung Cha
Dept. of Computer Science, Yonsei University

요 약

본 논문은 기존의 수동적 위치 추적 응용을 사용자가 시스템과 상호작용이 가능하도록 확장함으로써 지속적으로 이동하는 센서노드에게 사용자가 원하는 명령을 전달할 때 메시지의 양이 많아지는 문제점을 해결 하기위한 기법을 제시한다. 이 기법은 상황에 따라 전체 센서 필드에 데이터를 전파하는 full-scale flooding 과 지역적으로 데이터를 전파하는 constrained flooding을 상황에 따라서 선택 사용하는 방식으로 서로의 약점을 보완함으로써 최대의 효과를 얻는다. 시뮬레이션을 바탕으로 full-scale flooding만을 사용했을 때와 비교했을 때 이동 노드의 수가 늘어나는 과정에서 10%-90%까지 메시지의 양을 줄일 수 있었고, 최악의 상황에서도 full-scale flooding보다 그 결과가 나쁘지 않다.

1. 서 론

최근 많은 센서 네트워크 응용들의 실용화 가능성이 입증됨에 따라 관련 기술 연구가 활발하게 이루어지고 있다. 이중 위치 추적 응용은 최근 주목받고 있는 센서 네트워크 응용분야 중 하나이다. 현재 RF를 사용한 Motetrack[1] 그리고 초음파를 사용한 Cricket[2] 과 같은 위치 추적 응용들은 다수의 이동 센서 노드를 지원하지 않거나, 사용자가 시스템에 대한 제어 기능이 배제된 수동적 위치 추적에 대부분이다. 이와 같은 제한을 극복하기 위해서 능동적인 위치 추적 응용이 요구된다.

현재까지 능동적 위치 추적 연구를 위해 센서 필드 내의 이동 센서 노드들을 찾아서 사용자가 원하는 명령을 내리기 위해 사용될 수 있는 몇 가지 기존 연구들이 있다. Two-Tier Data Dissemination[3]은 이동 센서 노드를 일부 고려하고 있으나 데이터를 전파하는 노드가 움직일 때 마다 새로운 격자를 만들어야 하므로 비용이 많이 든다. Mobicast[4]는 대상의 움직임을 예측하여 그 주변으로 constrained flooding을 하는 기법으로 정확하고 효율적으로 대상을 찾는다. 이 기법에서는 이벤트를 계속해서 찾아야 하지만, 이는 본 연구 목적을 위해서는 불필요하다. 즉, 본 연구에서는 on-demand 형식의 시스템에 초점을 두기 때문에 계속해서 이동 센서 노드를 추적하는 것은 불필요한 메시지를 전파하여 자원의 낭비를 초래하므로 궁극적인 해결책이 될 수 없다. Full-scale flooding은 원시적 기법으로 문제를 해결할 수 있지만 많은 네트워크 메시지를 발생시키는 문제가 있다. 이는 메시지의 중복 전송, 회선 쟁탈(contention), 패킷 충돌 등의 문제가 발생하는 원인이 되며 결과적으로 시스템 에너지 사용량과 네트워크 그래픽을 증가시킨다[5].

본 논문에서는 다수의 이동 센서 노드 추적 기법을 제안한다. 이 기법은 사용자가 제공하는 데이터와 이동 노드에서 발생하는 데이터를 이용하여 상황에 따라 full-scale flooding과 constrained flooding 중 하나를 선택하는 방법이다. Constrained flooding은 이동 노드에서 발생한 최근 정보를 이용하여 정확하게 이동 센서 노드에 제어 메시지를 보내지만, 그 범위가 제한적이기 때문에 메시

지 양을 줄일 수 있게 된다.

2. 다수의 이동 노드 추적 기법

제한한 기법을 설계함에 있어 필요한 센서 노드의 종류들과 역할은 다음과 같다. 이동 노드는 이동성이 있는 물체에 부착이 되어 주기적으로 그 물체의 위치와 속도정보를 제공한다. Backbone노드는 위치 정보 데이터를 싱크 쪽으로 보내는 고정 노드 역할과 제어 메시지를 전송하기 위한 reverse path를 유지하는 역할을 한다. Head노드는 이동 노드가 처음으로 데이터를 보내는 backbone노드들 중 하나로 사용자가 제어 메시지를 전송 했을 때 constrained flooding을 하는 노드이다. Sink는 메인 시스템과 연결 되어있어서 충분한 자원과 전력이 공급된다고 가정한다. Backbone노드들은 각자 자신의 위치를 GPS를 통해서 알고 있고, topology는 격자라고 가정한다. 마지막으로 sink는 이동 센서 노드에서 전송된 자료들을 주기적으로 받고 저장한다. 이와 같은 가정을 바탕으로 제안 기법은 사전 과정으로 flooding 기법을 결정하기 위한 조사 단계를 거쳐, 이후 full-scale flooding 단계와 constrained flooding 단계인 추적과 포획단계로 나누어진다.

2.1 조사 단계

조사 단계에서는 제안 기법이 어떠한 방법으로 full-scale flooding 또는 constrained flooding을 선택할지 결정한다. 두 가지 flooding 방법을 병행해야 하는 이유는 사용자가 제어 데이터를 보내고 싶은 이동 노드의 수에 따라서 어느 한쪽의 flooding 기법이 다른 한쪽보다 우월하기 때문이다. 다시 말해 찾아야 할 이동 노드의 수가 적을 때는 constrained flooding의 메시지 수가 full-scale flooding에 비해서 적지만 노드의 수가 많으면 많아질수록 constrained flooding지역이 중복될 가능성이 많아진다. 이는 constrained flooding의 메시지 수가 full-scale flooding것 보다 어느 순간 많아질 수 있기 때문이다. 이러한 문제를 해결하기 위해서 조사 단계에서는 메시지의 양이 메시지의 전송 지역과 비례한다는 사실[6]을 바탕으로 아래의 flooding 지역의 크기 산출 수식을 설계 했다.

$$T = m \times n \quad (1)$$

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$$R = \sum_{i=1}^M \pi (v_{\max} \times \Delta t_i)^2 + \sum_{i=1}^M d_i$$

수식 1에서 T 는 full-scale flooding을 했을 경우 메시지가 전체 네트워크로 모두 전파되기 때문에 flooding영역을 topology의 가로(m) 곱하기 세로(n)로 나타낸다. d 는 sink에서부터 head노드까지의 거리를 구한 값으로 constrained flooding 지역 크기를 구한 후 그 값에 더할 가중치를 뜻한다. Constrained flooding 영역(R)은 각각의 이동 노드의 최고 속도(v_{\max})와 이동 노드로부터 받은 위치 추적 데이터의 경과시간(Δt)을 이용해서 전파 지역의 크기를 모두 합한 값에 거리의 가중치를 합한 값이다. 결국 T 와 R 을 비교하여 어떠한 flooding기술을 사용할지 결정하게 된다. 만약 T 가 R 보다 클 경우에는 constrained flooding을 쓰는 것이 유리하기 때문에 이 방식을 선택하고 반대의 경우에는 full-scale flooding을 선택하게 된다.

이 조사 단계에서 full-scale flooding이 선택 된다면 단순히 simple flooding으로 제어 메시지를 전체 네트워크에 송신하게 된다. 만약 constrained flooding이 선택 된다면 추적과 포획단계로 들어가게 된다.

2.2 추적 단계

추적 단계에서는 sink에서 head노드까지 제어 메시지의 전송이 이루어진다. 기존의 많은 데이터 전송 기법들은 노드의 자원 제한성이라는 특징 때문에 본 연구에는 적합하지 않다. 센서 노드는 이미 기존의 위치 추적 응용코드를 탑재하고 있으므로 그 위에서 동작하는 추적 기법은 복잡성을 피해야만 한다. 그러므로 추적 단계에서 간단하지만 효율적인 reverse path 라우팅을 사용한다. 이 라우팅은 이동 노드가 위치 추적 데이터를 주기적으로 sink를 향하여 전송한다는 사실을 이용해서 reverse path를 만들어 간다. 다시 말해 데이터가 head노드에서부터 sink쪽의 노드로 갈 때 마다 송신 노드는 출발지 노드의 주소를 기억하는 방법으로 경로를 만들어 가는 것이다. 그림 1은 이와 같은 과정을 보여주고 있다.

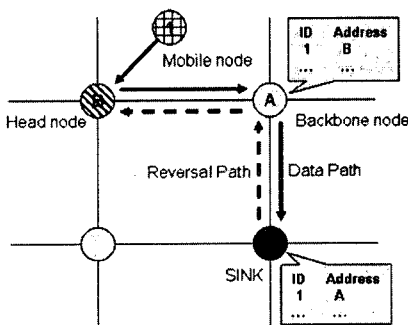


그림 1 reverse path 라우팅 과정

이와 같은 single path 기법은 코드 크기가 작기 때문에 신뢰성을 보완해주는 라우팅 기법이나 기존 연구에서 제안되었던 수동적 위치 추적 응용과 병행하여 사용 될 수 있다. 제안하는 기법은 센서 노드의 자원을 고려해서 최근의 reverse path 주소만을 저장하는 기법을 사용한다. 따라서 backbone노드는 하나의 이동 노드에 대해 하나의 주소 공간을 제공한다.

2.3 포획 단계

Head노드까지 도착한 제어 메시지는 constrained flooding을 통

해서 지역적으로 전파된다. 조사 단계에서는 최악의 상황을 고려해야 하기 때문에 이동 센서 노드의 최대 속도를 사용 했지만, 일반적으로 물체가 최대 속도를 계속 유지하는 일은 극히 드물다. 그러므로 포획 단계에서는 효율적이지만 정확히 이동 노드를 찾을 수 있는 정도의 flooding지역 크기를 결정한다.

제시하는 기법에 대하여 자세히 기술하기 전에 flooding 지역의 크기와 메시지 수의 크기가 어떤 상관관계를 갖는지를 분석함으로써 이 단계가 중요성을 입증한다. 다음 수식은 격자모습의 센서 필드 위에서 원의 모습을 한 flooding 지역이 커질 때 데이터 메시지를 발생시키는 노드의 수가 어떻게 늘어나는지를 보여준다.

$$g(x) = \left\lfloor \sqrt{\left(\frac{r}{l}\right)^2 - x^2} \right\rfloor, (0 < x < \left\lfloor \frac{r}{l} \right\rfloor, x \in \mathbb{Z}^+) \quad (2)$$

$$N = a \left(\left\lfloor \frac{r}{l} \right\rfloor + \sum_{x=1}^{\left\lfloor \frac{r}{l} \right\rfloor} g(x) \right)$$

수식 2에서 r 은 flooding 지역의 반지름이고, l 은 두 노드의 거리이다. 함수 $g(x)$ 는 원의 1/4안에 원호위의 점과 x 축의 정수 점을 y 축과 대칭으로 연결하는 선위에 몇 개의 노드가 있는지를 계산한다. 결국 $x=0$ 일 때를 제외하고 원의 1/4안에 있는 모든 선들 위에 노드 수를 모두 합한 후 $a(4)$ 를 곱한 값이 원안에 있는 대략적인 노드의 수가 된다. $x=0$ 일 때는 중복이 되기 때문에 별도로 계산하여 합하게 되면 최종적인 flooding 지역 안에 있는 정확한 노드의 수(N)를 계산할 수가 있다. 이 식에서 반지름이 늘어나면 노드의 수는 지수적으로 늘어남을 알 수 있다. 따라서 flooding지역을 제한하는 것이 중요함을 보여준다.

위에서 언급했던 바와 같이 물체가 최대 속도를 계속 유지 하는 경우가 드물다는 것에 착안하여 가속도를 이용해 새로운 flooding 지역을 만든다. 이와 관련된 수식은 아래와 같다.

$$a = \frac{(v_{\max} - v_{\text{latest}}) \Delta t}{((v_{\max})^2 - (v_{\text{latest}})^2)}$$

$$a = \frac{2r}{((v_{\max})^2 - (v_{\text{latest}})^2)}$$

$$r_{\text{optimal}} = \frac{2a}{2a} \quad (3)$$

수식 3의 첫 번째 식은 어떤 시간동안 처음 속도에서 나중 속도까지 올릴 때 필요한 가속도(a)를 구하는 식이고, 그 다음 수식은 거리와 관련된 가속도 식이다. 이 두식을 이용해서 새로운 최적화 반지름을 구하는 수식을 구할 수가 있다. 출발 속도를 v_{latest} (최근의 이동 노드 속도)로 정하고 마지막 속도를 노드의 최대 속도로 정한 후 그 최근 속도의 데이터가 얼마나 오래 됐는지를 알려주는 Δt 를 이용하면 가속도가 나온다. 그 가속도를 이용하면 최대 속도만을 이용해서 산출되는 반지름보다 작지만 그 물체가 갈 수 있는 최대 반경(r_{optimal})을 알 수가 있다. 예를 들어 최근 속도가 2m/s, 최대 속도가 5m/s이라고 할 때 가속도는 $1.5m/s^2$ 이고 이것을 이용해서 산출된 최적화 반지름은 7m이다. 반면에 최대 속도만을 이용한다면 반지름이 10m가 나오게 된다. 따라서 작은 크기의 flooding지역을 만들어 낼 수 있기 때문에 메시지 수를 현저히 줄일 수 있다.

3. 성능 평가

제시한 기법의 성능 평가를 위해 센서 네트워크를 위해 제작된 TOSSIM[7] 시뮬레이터를 사용했다. 센서 필드는 10x10 크기의 격자 방식을 채택했고, Backbone노드의 수는 100개 그리고 두 가지 종류의 이동 노드(3m/s의 노드, 6m/s의 노드)들을 사용했다. 이동 노드의 최대 속도는 6m/s로 설정했고, 두 노드의 거리는 3m로 설

정하였다.

제시한 기법에 대하여 300여회 정도 반복적인 실험을 실시하였고 이동 노드 추적에 실패한 경우는 1회도 발생하지 않았다. 이는 이동 노드의 속도가 최대 속도를 넘지 않거나 같기 때문에 flooding 지역을 벗어날 수 없으며 15m까지 데이터를 전파할 수 있는 라디오를 사용하였기 때문에 데이터의 손실이 일어나지 않은 것을 원인으로 들 수 있다. 또한 최대 속도를 이용해서 flooding 기법을 결정하기 때문에 constrained flooding의 메시지수가 full-scale flooding보다 많은 경우가 없었던 점도 다른 원인이다. 그림 2는 노드의 수가 늘어날수록 메시지의 수가 어떻게 변하는지를 보여주고 추적 단계에서 constrained flooding과 full-scale flooding을 잘 선택해서 제어 메시지를 전송하는 것을 보여준다.

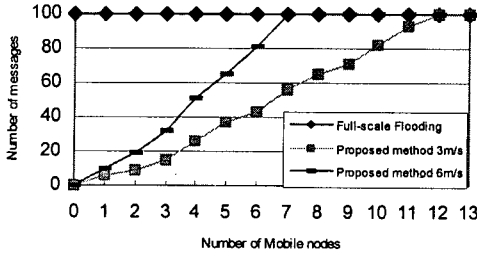


그림 2 이동 노드 수의 변화에 따라 메시지 수의 변화

더 정확한 데이터를 구하기 위해서 이동 노드의 출발 위치를 바꾸어 가면서 그 평균 메시지 수를 구했다. 또한 이동 노드의 속도도 3m/s과 6m/s 중 임의로 정해서 그 값을 구했다. 이 값을 바탕으로 full-scale flooding과 제안 기법을 사용했을 때 네트워크 트래픽 감소율을 구하고 에너지 사용률을 비교했다.

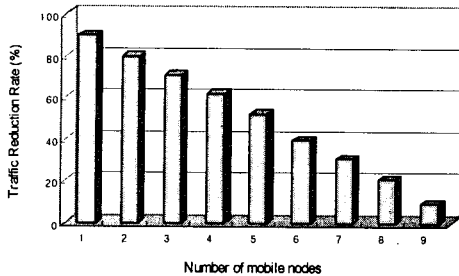


그림 3 네트워크 트래픽 감소율

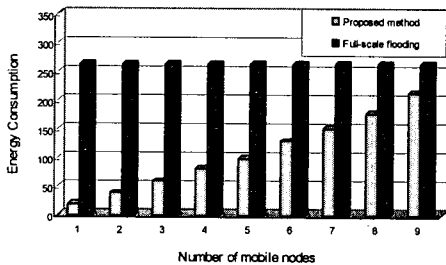


그림 4 에너지 사용률의 비교

그림 3은 constrained flooding을 사용할 경우 네트워크 트래픽이 full-scale flooding의 경우와 비교할 때 얼마나 감소하는지를 보이는 그래프로 감소율은 $((M_f - M_c) \times 100) / M_f$ 로 산출한다. 이때 M_f 는 full-scale flooding을 할 때 발생하는 메시지 수고, M_c 는 제시하는 기법을 사용할 때 발생하는 메시지 수다. 결과적으로 이동 노드의 수가 적음에 따라서 많은 양의 네트워크 트래픽 감소를 보인다.

그림 4에서는 두 flooding 기법의 에너지 사용량을 비교한 것이다. 에너지는 네트워크 트래픽과 달리 송신뿐 아니라 수신도 고려해야 한다. 따라서 MICA[8]의 명세서를 참고로 송신할 때 가중치를 24로 수신할 때 8로 정하고 에너지 율을 $M_s + M_r \times (8/24)$ 로 구했다. 여기서 M_s 는 송신 메시지의 수고, M_r 는 수신 할 때의 메시지 수다. 에너지를 구할 때 필요한 전압과 시간은 같기 때문에 생략했다. 이 결과에서도 제시한 기법이 율등한 전력 소모의 감소를 보여주고 있다.

4. 결론

본 논문에서는 기존의 센서 네트워크 위치 추적 응용을 능동적으로 확장할 때 발생하는 이동 노드의 추적 문제를 효율적으로 해결 하기 위해서 full-scale flooding과 constrained flooding을 함께 쓰는 기법을 제안했다. 또한 constrained flooding 지역을 효과적으로 줄일 수 있는 방법을 제시 했다. 이와 같은 기법을 적용함으로써 네트워크 트래픽과 에너지 사용량을 효과적으로 줄일 수 있음을 검증했다. 또한 제안한 기법은 간단함을 강조했다기 때문에 기존의 위치 추적 응용위에 적용 시킬 수 있으므로 기존의 기법들에 최소한의 수정으로 적용 시킬 수 있는 장점이 있다.

향후 연구로서 노드의 오작동이나 전력 공급의 실패로 인해서 reverse path가 유지되지 못하는 신뢰성 문제를 해결 중이다. 또한 constrained flooding의 문제인 중복 flooding 지역을 피할 수 있도록 하는 기법의 개발은 제시한 기법의 성능을 향상 시킬 수 있을 것이다.

참고문헌

- [1]Konrad Lorincz and Matt Welsh, "MoteTrack:A Robust, Decent realized Approach to RF-Based Location Tracking," *In Proc. the International Workshop on Location and Context-Awareness (LoCA 2005)*, May 2005.
- [2]Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan, "The Cricket Location-Support System," *In Proc. 6th ACM MOBICOM Conf.*, pp. 32-43, August 2000.
- [3]Fan Ye, Haiyun Luo, Jerry Cheng, Songwu Lu, and Lixia Zhang, "A Two-Tier Data Dissemination Model for Large-scale Wireless Sensor Networks," *In Proc. ACM MOBICOM*, pp. 148-159, September 2002.
- [4]Qingfeng Huang, Chenyang Lu, and Gruia-Catalin Roman, "Dissimulation: Spatiotemporal Multicast in Sensor Networks," *In Proc. ACM SenSys'03*, pp. 205-217, November 2003.
- [5]Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, and Jang-Ping Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," *In Proc. 5th ACM MOBICOM Conf.*, pp. 153-167, August 1999.
- [6]Marco Zuniga Z. and Bhaskar Krishnamachari, "Optimal Transmission Radius for Flooding in Large Scale Sensor Networks," *In Proc. 23rd IEEE International Conference on Distributed Computing System Workshops (ICDCSW'03)*, pp. 697-702, May 2003.
- [7]Philip Levis, Nelson Lee, Matt Welch, and David Culler, "TOSS IM: Accurate and Scalable Simulation of Entire TinyOS Applications," *In Proc. ACM SenSys'03*, pp. 126-137, November 2003.
- [8]Crossbow, *MICA Data sheet*, http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/6020-0042-06_B_MICA2.pdf