

유비쿼터스 센서 네트워크에서 위치와 장애물을 고려한 플러딩 방법

이현정^o 한규호 안순신

고려대학교

{hjlee^o, garget, sunshin}@dsys.korea.ac.kr

Flooding Scheme considering Location and Barrier in Ubiquitous Sensor Network

Hyunjung Lee^o Kyuho Han, Sunshin An
Korea University

요 약

통신모듈이 포함된 많은 컴퓨팅 장치들이 다가오는 미래에는 유비쿼터스 네트워크 디바이스가 될 것이다. 센서 네트워크도 그런 유비쿼터스 네트워크의 어플리케이션이다. 대부분의 경우 센서 네트워크는 플러딩기반의 on-demand 에드혹 라우팅 프로토콜을 사용하나, 플러딩에 따른 중복된 패킷 전송을 최소화하기 위해, 위치 정보를 사용하고 장애물까지 고려한다. 본 논문은 건물 내 유비쿼터스 센서 네트워크에서 위치 기반과 장애물을 고려한 플러딩을 제안한다.

1. 서 론

PDA, 노트, PC, MP3 플레이어와 같은 많은 개인용 컴퓨팅 장치들이 우리의 생활 공간에 퍼져있다. 이런 장치들은 통신모듈을 가지고 있으며 미래에는 유비쿼터스 컴퓨팅 장치가 될 것이다. 유비쿼터스 네트워크의 어플리케이션 중 센서 네트워크는 무선 유비쿼터스 센서 장치로 환경 정보를 수집하고 다른 여러 유비쿼터스 네트워크 어플리케이션을 지원한다. 이제까지 많은 센서 네트워크는 플러딩에 기반한 on-demand 에드혹 라우팅 프로토콜을 연구해왔다. 그러나 플러딩은 중복되는 패킷전송의 경우가 많고, 이를 위해 많은 연구들이 위치 정보를 사용했으나, 건물 내의 경우 장애물에 대한 고려는 없었다. 본 논문에서는 유비쿼터스 센서 네트워크의 여러 특징을 고려하여, 필요조건을 결정할 것이다. 본 논문의 중요한 목적은 건물 내 환경의 지원과 에너지 절감 및 분산 컴퓨팅 방법을 연구하는 것이다.

라우팅 프로토콜은 지리적으로나 시간적으로 최단거리를 찾기 내는데 위치 정보를 이용한다. 그러나 모든 노드가 이웃 정보를 유지하도록 요구하고, 이것은 제어 트래픽과 중복된 에너지 소모를 가져온다. 패킷 플러딩 범위는 목적지 지역과 각에 의해 제한되고 이웃에 대한 정보는 필요로 하지 않는다. 그러나 장애물에 대한 고려가 없고, 그래서 새로운 루트를 찾아내는데 실패하는 원인이 될 수도 있다.

본 논문은 장애물을 고려한 플러딩 방법을 소개하고, 각 노드는 분산 방법으로 자동 결정으로 중복된 에너지 소모 중 하나를 야기하는 중복된 패킷의 교환을 최소화

한다. 일반적인 플러딩 방법에서, 소스 노드는 목적지 노드로 패킷을 브로드캐스팅 한다. 그리고 만약 소스 노드의 라디오 전파 범위 내에 어떤 노드가 있으면, 노드는 동일 패킷을 받은 기록이 없는 노드로 패킷을 브로드캐스팅 한다. 이 메커니즘은 목적지 노드가 해당 패킷을 받을 때까지 반복된다. 그러나 일반적인 플러딩 방법은 중복된 에너지 소모의 원인이 되는 다수의 중복 플러딩 패킷을 일으키는 원인이 된다.

2. 관련 연구

유비쿼터스 센서 네트워크는 무선 에드혹 네트워크와 유사하다. 두 네트워크 모두 특정 infrastructure를 가지고 있지 않고, 에너지 제한적인 배터리를 통한 전력 공급을 받고, 무선 연결과 이동성 등의 공통적 특징을 가진다. 이런 특징으로 인한 문제점을 극복하기 위해, 본 논문은 PAMAS(Power aware Multi-Access Protocol with Signaling) 및 on-demand 라우팅 프로토콜 등을 사용한다.

그러나 무선 에드혹 네트워크와는 다른 점을 유비쿼터스 센서 네트워크는 가지고 있는데, 센서 노드는 더 작은 크기로 보다 제한적인 에너지를 가져 최대 수명이 약 6개월에서 1년 정도를 가질 필요가 있다. 그러나 크기는 작지만 많은 경우 에드혹 노드보다 일반적이고 normal한 트래픽을 만들지만 센서 노드는 different time, different 주기로 정보를 수집하기를 바라므로 보다 빈번한 트래픽을 요구한다. 또한 대부분의 유비쿼터스 센서 네트워크는 센서 데이터를 모다 특정 노드로 수집하고자 하므로

방향성을 가지는 트래픽이 많다.

실제 위치 정보를 사용하는 많은 라우팅 방법들이 있으나 이들은 크게 두 가지 목적을 가지는데, 위치 정보를 사용해서 최단거리를 찾는 것으로, MFR(Most Forward within Radius)나 NFP(Nearest with forward Progress)방법 등이 그러하다. 그러나 이런 위치기반 라우팅 방법은 이웃 노드에 대한 정보를 유지해야 해서, 제어 트래픽과 중복 에너지 소모를 가져온다. 다른 방법은 목적지의 범위와 각에 의해 패킷 플러딩 범위를 제한하는 것으로, LAR(Location Aided Routing)이나 DREAM과 같은 Directed Flooding이다. 이 방법은 이웃노드에 대한 정보는 필요 없지만, 소스 노드와 목적지 노드 사이에 위치한 장애물에 대한 고려가 없어서 route를 찾는데 실패할 가능성이 높다. 그림 1 이 바로 이러한 경우를 보여준다

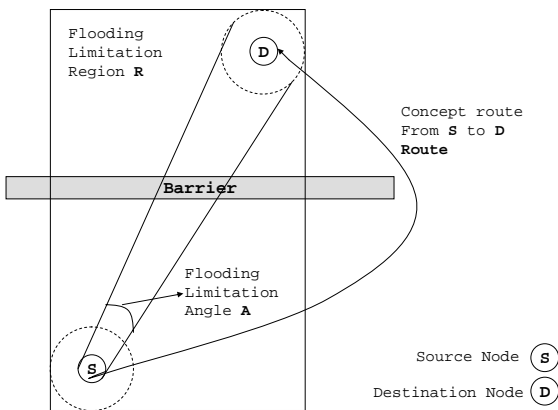


그림 1 route를 찾는데 실패하는 경우

3. 건물 내 유비쿼터스 센서를 위한 위치기반 플러딩

유비쿼터스 센서 네트워크 및 위치기반 라우팅 방법은 장애물이 센서 노드와 싱크 노드간에 위치한 건물 내 환경의 경우에 한해 적용되는 것으로 제한한다. 본 논문은 수신 측 결정 및 지연 결정에 의한 플러딩 방법을 제안할 것이다.

중복 패킷 브로드캐스팅을 최소화하기 위해, 중간 노드는 적절한 패킷만을 브로드캐스팅 하는 것이 이상적이나, 무선 통신 환경에서, 어떤 추가적 정보 없이 중간 노드의 이웃의 위치정보에 대해 알기는 어렵다. 유비쿼터스 센서 네트워크에서, 센서 노드는 매우 작은 capacity를 가지므로, 추가적 제어 정보를 사용하는 것은 에너지 관리 측면에서 나쁜 결과를 초래할 수 있다.

그림 2는 수신자 측면 결정에 대한 개념을 보여준다.

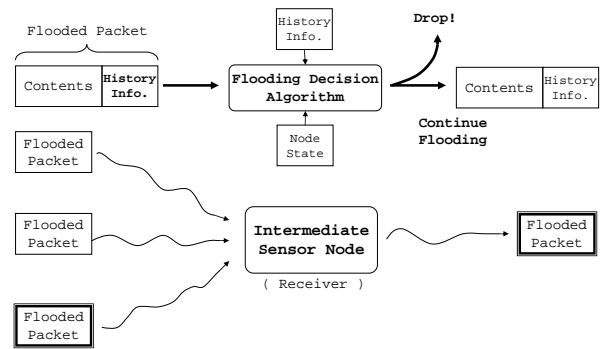


그림 2 수신자 측면 결정의 개념도

발송노드는 이웃이 좋은 수신자를 위한 후보자인지 어떤 결정도 내리지 않는다. 발송자가 패킷을 보낼 때, 추가 정보를 보내고, 이 정보는 그림 3에 있는 변수(variable)로 저장된다. 그림 3은 pseudo 코드로 작성된 본 논문이 제시하는 플러딩 방법을 기술한다.

```

struct Packet {
    location src, dst;
    int numOfhop;
    set len[Delay+1]; → len[0] = distance between src and dst;
}
* At each node *
static final location node;
void receive( Packet P ) {
    myLen = distance between node and dst;
    if ( Packet P is arrived at First ) {
        P.numOfhop++;
        if ( P.dst == nodeaddr )
            [Stop Flooding: its Destination]
        if ( len[--numOfhop] >= myLen ) {
            len[1] = myLen;
            P.numOfhop = 1;
            [Continue Flooding Packet P:This packet may flow to destination]
        }
        if ( P.numOfhop < Delay ) {
            P.len[numOfhop] = myLen;
            [Continue Flooding Packet P: Delayed determination]
        } else
            [ Stop Flooding: this packet may not flow to destination]
    }
    [Stop Flooding: It's duplicated Packet]
}
    
```

그림 3 pseudo 코드: 플러딩 결정 알고리즘

각 노드는 플러딩 패킷에 위치 정보를 추가한다. 그리고 나서, N-hop(그림 4에서, 3 hop) 후에, 중간 노드들(그림 4에서, 노드 4와 3)은 히스토리 정보를 분석해서 히스토리가 있건 없건 간에 플러딩을 결정한다. 그림 4에서, 노드 3은 플러딩 중지를 결정한다. 그러나 노드 5는 히스토리에 의해 플러딩을 계속한다.

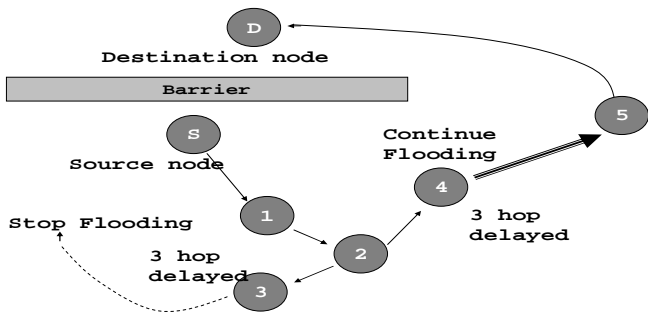


그림 4 delay 3의 지연결정의 예

플러딩 방법을 디자인하기 위해서, 본 논문은 N-hop 위에 위치하는 노드의 위치 축 정보를 더하고, 노드는 해당 위치 축(x, y)같은 세트에서 관심 값을 계산한다. 이런 계산을 분산하기 위해서, 우리는 히스토리 정보로 노드와 목적지 노드간의 거리를 결정한다. 그림 4 에서, 만약 소스 노드와 목적지 노드간에 장애물이 있으면, 방향성은 중요하지 않다. 장애물이 방향성을 가진 플러딩의 방해하기 때문이다. 그림 4 에서, 노드 4는 그 이전의 노드(노드 2)와 목적지 노드 사이의 거리를 가지고 자신과 목적지 노드간의 거리를 계산한다. 그리고 노드 4는 계산된 거리가 더 짧으므로 목적지 노드로 더 가까이 가도록 플러딩 패킷을 전송하도록 결정한다. 그리고 나서, 계속 플러딩하고 자신의 히스토리 정보를 갱신한다. 그러나 노드 3의 경우는, 경우가 반대이다. 노드 3은 이전 노드(노드 2)와 자신의 거리를 비교해보니 계산된 거리가 더 길다고 판단하여, 패킷을 브로드캐스팅 하는 것을 멈춘다.

4. 실험 결과

유비쿼터스 센서 네트워크에서의 건물 내 환경을 위한 위치기반 플러딩을 시뮬레이션 하기 위해, 간단한 건물 내 환경 및 거리 제한적인 무선 브로드캐스팅 하는 모델을 만들었다. 그리고 JAVA 프로그래밍 언어를 통해 플러딩 방법을 구현했다. 본 논문은 여러 요구사항 중 가능한 패킷 전송을 줄이는 것에 관심을 두었고, 이는 패킷 전송 수로 센서 노드가 더 이상 패킷을 플러딩 할 수 없을 때까지의 전송 수 이다. 또한 본 논문은 일반적인 플러딩 방법과 본 논문이 제시한 플러딩 방법간 패킷 전송 수를 비교 하였다.

4.1 실험 모델

실험을 단편화를 위해, 센서 노드는 grid하게 만들었고, grid내 inter-section에만 센서노드는 위치한다.

먼저, 건물 내 환경은 grid를 10X10 사이즈로 하여 grid의 중앙에 장애물을 위치하였다. Grid는 2구역으로 나뉘 하나는 소스노드로 다른 쪽은 싱크 노드로 하고,

각 구역에는 랜덤하게 30개의 노드를 둔다.

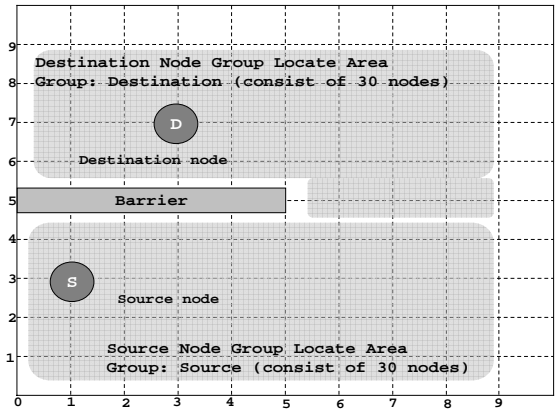


그림 5 실험환경모델

실험을 위한 JAVA의 구현을 단순하게 하기 위해, 사각형으로 거리 제한적 무선 브로드캐스팅 한다. 그림 6는 사각형의 무선 브로드캐스팅을 보여준다. 플러딩 노드 F는 패킷을 브로드캐스팅 하고, inter-section인 R1에서 R8에 위치한 노드들은 브로드캐스팅 된 패킷을 수신할 수 있다.

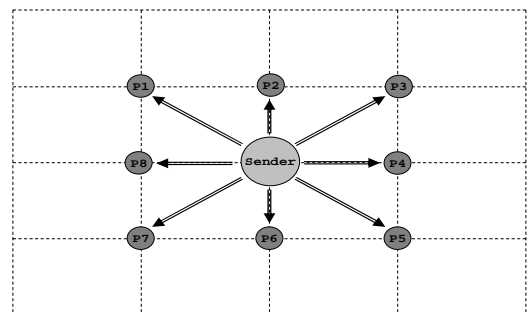


그림 6 사각형으로 브로드캐스팅

4.2 구현

각 노드는 랜덤하게 위치하고, 소스노드와 목적지노드도 랜덤하게 선택된다. 먼저 소스센서노드는 일반적인 플러딩 방법으로 패킷을 목적지 센서 노드로 플러딩 한다. 그리고 소스 노드는 본 논문이 제시한 플러딩 방법으로 패킷을 목적지로 플러딩 한다. 동일한 토폴로지와 동일한 소스 및 목적지 노드이다. 그리고 브로드캐스팅 한 수와 받은 패킷의 수를 비교한다. 이 실험은 각 delayed 결정 요인마다 1000번씩 반복한다.

4.3 실험 결과 & 분석

네트워크에 더 이상의 플러딩이 없을 때, 브로드캐스팅 및 receiving의 수를 trace한다. 아래 표는 실험 결과를 보여준다. Route Failure Rate는

route를 찾는 평균 값을 의미한다.

Delay가 증가함에 따라, route discovery failure rate은 감소하고 전송의 수는 증가함을 보여준다. Delay 4에서 가장 좋은 결과를 보여준다. 실제 건물 내 환경에서, 센서 노드는 문이나 복도 등을 통해 분산될 수 있고, 그럴 경우 delay 3의 결과를 얻을 것으로 생각한다. Delay 3의 총 전송 수는 약 50%가 감소된 결과를 보여준다.

표 1 RESULT AT DELAY 2

1000 experiments , Delay 2			
	Route Failure rate	Ave. total Broadcast	Ave. Total Receive
General Flooding	0	95	625
Proposed Flooding	0.776	14	87

표 2 RESULT AT DELAY 3

1000 experiments , Delay 3			
	Route Failure rate	Ave. total Broadcast	Ave. Total Receive
General Flooding	0	94	624
Proposed Flooding	0.324	49	321

표 3 RESULT AT DELAY 4

1000 experiments , Delay 4			
	Route Failure rate	Ave. total Broadcast	Ave. Total Receive
General Flooding	0	94	624
Proposed Flooding	0.075	70	465

표 4 RESULT AT DELAY 5

1000 experiments , Delay 5			
	Route Failure rate	Ave. total Broadcast	Ave. Total Receive
General Flooding	0	94	624
Proposed Flooding	0	82	546

5. 결론 및 Future Work

플러딩의 중복 패킷 전송을 최소화하고 건물 내 환경을 지원하기 위해, 본 논문은 위치기반 플러딩을 제안했다. 주요 요지는 수신 측 결정 및 delayed 결정이다. 이를 통해, 본 논문의 실험은 총 전송 수가 약 15%에서 50%까지 감소함을 보이거나 trade-off는 route failure rate이다.

추후에는 건물 내 환경에 따라 delay를 동적으로 바꿀 수 있는 환경에서의 유비쿼터스 센서 네트워크를 위한 adaptive 위치기반 플러딩을 제안할 것을 계획이다.

6. 참고문헌

[1] Edgar H. Callaway, Jr, Wireless Sensor Networks :Architectures and Protocols (AUERBACH, 2003)

[2] C.S. Raghavendra & S. Singh, PARMAS-Power-Aware Multi-Access Protocol with Signaling for Ad Hoc Networks, IEEE INFOCOM, 2002

[3] H. Takagi & L.Kleinrock, Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals, IEEE Transaction on Communication, 32,246-257, 1984

[4] T.Hou & V.Li, Transmisiion range control in multihop packet radio network, IEEE Transactions on Communication, 34,38-44, 1986

[5] Young-Bae Ko & Nitin H. Vaildya, Location aided routing(LAR) in mobile ad hoc networks, International Conference on Mobile Computing and Networking, Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking

[6] Basagni, I. Chlamtac, V. R. Syrotiuk, & B. A. Woodward. A Distance Routing Effect Algorithm for Mobility (DREAM). In Proceedingsof the fourth annual ACM/IEEE International Conference on Mobilecomputing and networking (MobiCom '98), pages 76-84, Dallas, Texas,October 1998.

[7] Uwe Hansmann, Lothar Merk, Martin S. Nicklous & Thomas Stober, Pervasive Computing