

무선 센서 네트워크의 데이터 전송 특성을 고려한 효율적인 플러딩 기법

안 상 현[†] · 임 유 진^{**} · 김 만 희^{***}

요 약

무선 센서 네트워크에서는 질의 분배(dissemination)나 이벤트 광고를 위해 플러딩이 요구된다. Simple(또는 blind) 플러딩은 폭주(implosion) 문제와 겹침(overlap) 문제를 야기하며, 따라서 simple 플러딩은 센서 네트워크의 수명을 감소시킬 수 있다. 따라서 본 논문에서는 플러딩 오버헤드를 줄이기 위해 플러딩 오버레이 구조(Flooding Overlay Structure: FOS)를 제안한다. 두 종류의 FOS 기법인 중앙형 FOS(Centralized FOS: CFOS)와 분산형 FOS(Distributed FOS: DFOS)를 제안하며, CFOS에서는 싱크가 네트워크 토폴로지 정보를 수집해서 그 정보를 기반으로 포워딩 노드를 선택하는 반면, DFOS에서는 각 센서 노드가 자신의 로컬 정보를 기반으로 자신이 브로드캐스트 패킷의 포워딩에 참여할지 여부를 결정한다. 제안한 FOS 기법들의 성능 분석을 위해 NS-2 기반의 시뮬레이션을 수행했으며, FOS 기법들과 simple 플러딩 및 gossiping의 성능을 비교했다. 시뮬레이션 결과, 제안한 FOS 기법들이 네트워크 수명 측면에서 simple 플러딩보다, 데이터 전달을 측면에서 gossiping보다 우수함을 알 수 있었다.

키워드 : 무선 센서 네트워크, 플러딩, 오버레이

Efficient Flooding Mechanisms with Considering Data Transmission Characteristics of the Wireless Sensor Network

Sanghyun Ahn[†] · Yujin Lim^{**} · Mahnhee Kim^{***}

ABSTRACT

In the wireless sensor network, flooding is required for the dissemination of queries and event announcements. The simple flooding causes the implosion and the overlap problems, so the simple flooding may result in the reduced network lifetime. Therefore, in this paper, we propose the flooding overlay structure (FOS) so that the overhead caused by flooding can be reduced. We propose two variants of FOS mechanisms, the centralized FOS (CFOS) and the distributed FOS (DFOS). In CFOS, the sink collects the network topology information and selects forwarding nodes based on that information. On the other hand, DFOS allows each sensor node to decide whether to act as a forwarding node or not based on its local information. For the performance evaluation of our proposed mechanisms, we carry out NS-2 based simulations and compare ours with the simple flooding and the gossiping. The simulation results indicate that the proposed FOS mechanisms outperform the simple flooding in terms of the network lifetime and the gossiping in terms of the data delivery ratio.

Keywords : Wireless Sensor Network, Flooding, Overlay

1. 서 론

일반적으로, 무선 센서 네트워크에는 특정 영역의 요구된 상황을 모니터링하기 위해 무선 센서 노드들이 살포되며(deploy) 센서 노드들은 무선 채널을 통해 싱크에게 센싱한

데이터를 전송한다. 무선 센서 네트워크는 이동 애드혹 네트워크(Mobile Ad Hoc Network; MANET)와 여러 면에서 차이가 있다. 센서 노드는 MANET 노드에 비해서 이동성이 거의 없거나 정적이며 에너지가 훨씬 제약되어 있고, 센서 네트워크의 노드 밀도가 MANET의 노드 밀도보다 일반적으로 높다는 특징이 있다. 센서 네트워크의 가장 중요한 이슈 중의 하나는 네트워크 수명(lifetime)을 연장시키는 것으로, 일반적으로 네트워크 수명은 센서 노드가 처음으로 에너지 고갈로 인해 동작을 멈추는 시점까지로 정의된다. 센서 노드의 에너지 소모는 대부분 무선 통신으로 인해 생기며, 예를 들어, 100m 떨어진 지점으로 1비트를 전송하는데

※ 이 논문은 2008년도 서울시립대학교 교내학술연구비에 의하여 연구되었음.
† 정 회 원: 서울시립대학교 컴퓨터과학부 정교수(교신저자)
** 정 회 원: 수원대학교 정보미디어공학과 전임강사
*** 준 회 원: 서울시립대학교 컴퓨터통계학과 석사과정
논문접수: 2008년 9월 17일
수정일: 1차 2009년 1월 7일, 2차 2009년 1월 20일
심사완료: 2009년 1월 29일

소요되는 에너지는 명령 3000개를 실행하는데 소요되는 에너지에 해당한다^[1]. 따라서 데이터 전송 횟수가 센서 노드의 에너지 사용 효율성에 많은 영향을 준다. 무선 센서 네트워크에서 질의(query; 예, Directed Diffusion^[2]의 INTEREST 패킷 분배)나 이벤트 발생을 분배하기(disseminate) 위해서 브로드캐스트 기법이 필요하다. 가장 단순하게 브로드캐스트를 할 수 있는 방법으로는 simple 플러딩 기법이 있지만, simple 플러딩은 상당히 비효율적이다. Simple(또는 blind) 플러딩 방식에서는 패킷을 수신한 각 센서 노드가 이 패킷을 자신의 이웃 노드들에게 포워딩해야 하며, 이로 인해 충돌, 재전송, 중복 패킷이 많이 발생하게 된다. 특히, 밀집된 센서 네트워크의 경우 simple 플러딩으로 인한 영향을 많이 받게 된다.

따라서, 본 논문에서는 단일의 고정 싱크를 갖는 센서 네트워크에서 빈번하게 플러딩을 하는 것으로 인한 오버헤드를 감소시키기 위해 브로드캐스트 데이터의 포워딩을 전달하는 센서 노드들로 구성된 플러딩 오버레이 구조(Flooding Overlay Structure; FOS)를 제안한다. 중앙형 FOS(Centralized FOS; CFOS)와 분산형 FOS(Distributed FOS)의 두 종류를 제안하며, 시뮬레이션을 통해 이들 기법의 성능을 분석한다.

본 논문의 구성은 다음과 같다. 2장에서는 센서 네트워크 환경에서 simple 플러딩의 오버헤드를 줄이기 위한 관련 연구들에 대해서 소개하고, 3장에서는 본 논문에서 제안하는 FOS 기법들에 대해서 자세히 설명한다. 4장에서는 시뮬레이션을 통해 FOS 기법들의 성능 분석을 수행하며, 5장에서 결론을 맺는다.

2. 관련 연구 및 문제 정의

센서 네트워크에서 질의나 이벤트 데이터를 분배하는데 사용되는 simple 플러딩의 폭주(implosion) 문제 및 겹침(overlap) 문제를 해결하기 위해 제안된 대표적인 방법들은 다음과 같다.

Rumor 라우팅^[3]에서는 agent라는 특수 패킷을 사용해서 이벤트마다 이벤트 경로를 생성하며 질의는 플러딩 대신 랜덤 경로(random walk)를 통해 이벤트 경로를 만날 때까지 전달된다. 이렇게 함으로써 질의 플러딩 및 이벤트 플러딩을 회피할 수는 있지만 메시지의 전달이 보장되지 않는다.

Pulse 프로토콜^[4]은 펄스 소스가 주기적으로 펄스 플러딩을 함으로써 스페닝 트리를 주기적으로 갱신하도록 한다. 펄스 플러딩은 주기적인 라우팅 정보 갱신뿐만 아니라 네트워크 전체의 동기화를 위한 시간 정보도 제공하게 된다.

SPIN(Sensor Protocols for Information via Negotiation)^[5]에서는 센서 노드가 데이터를 보내기 전에 이웃들에게 데이터를 광고하며, 이 데이터를 원하는 노드가 요청 메시지를 보내오면 그 때 해당 데이터를 보내준다. SPIN에서는 센서 노드가 자신의 자원을 검사해서 만일 에너지가 일정 수준 이하이면 다른 노드들의 데이터를 포워딩하지 않음으로써 플러딩 범위를 축소시킨다.

Gossiping^[6]은 노드가 데이터를 임의로 선택한 하나의 이웃

에게만 포워딩하게 함으로써 폭주 문제를 해결하는 방식으로, 데이터 전달이 보장되지 않고 지연이 커지는 단점이 있다.

TTDD(Two-Tier Data Dissemination)^[7]는 다수의 이동 싱크가 있을 때의 문제를 다루며, 이벤트가 발생하면 데이터 소스는 그리드 구조를 형성하고 그리드 포인트로부터 가장 가까운 센서들에게 포워딩 정보를 설정해준다. 싱크로부터의 질의는 두 개의 층(tier)을 거쳐서 소스에게 전달되며, 하위 층은 싱크가 현재 속해 있는 그리드 영역(square) 내에 존재하고, 상위 층은 그리드 포인트의 분배 노드들로 구성된다. 싱크가 자신이 속한 그리드 영역으로 질의를 플러딩하면, 요청된 데이터로부터 가장 가까운 분배 노드가 질의를 소스쪽 상향(upstream) 분배 노드로 포워딩한다. 소스에 도달하거나 또는 소스로부터 데이터를 이미 수신한 분배 노드에 도달할 때까지 질의를 상향 방식으로 포워딩한다.

플러딩 기법에 대한 성능 비교 요소로는 메시지 오버헤드와 데이터 전달 보장이 있을 수 있다. 이런 측면에서 rumor 라우팅과 gossiping은 포워딩 노드를 임의로 선택하기 때문에 데이터 전달을 보장할 수 없다. Pulse 프로토콜은 주기적인 플러딩을 기반으로 동작하기 때문에 플러딩 관련 제어 메시지 오버헤드가 큰 문제가 있다. TTDD는 플러딩 오버헤드를 줄이기 위해서 그리드 구조를 사용하지만 모든 센서 노드가 자신의 위치를 알고 있다고 가정한다. 따라서 본 논문에서는 센서 노드가 자신의 위치를 알지 못 하는 환경에서 플러딩 오버헤드를 줄이면서도 데이터 전달을 보장해주는 방식을 제안하며, 이를 위해 플러딩 오버레이 구조(Flooding Overlay Structure; FOS)를 제안한다. 기존에, MANET 환경을 위한 효율적인 플러딩 기법들이 제안되었으며, [8][9][10]에 이들 기존 연구에 대한 정리가 되어 있다. 그러나 MANET을 위한 효율적인 플러딩 기법은 무선 센서 네트워크 환경에 적합하지 않으며, 그 이유는 MANET 노드에 비해서 센서 노드들이 전력이나 컴퓨팅 능력면에서 제한이 훨씬 심하기 때문이다. 또한 대부분의 경우 센서 노드는 MANET 노드에 비해서 이동성이 없거나 한정되어 있으며, 따라서 MANET 플러딩 기법은 이동성이 많은 MANET을 위해 제안된 것이기 때문에 정적인 무선 센서 네트워크 환경에 적합하지 않을 뿐만 아니라 센서 노드의 전력 및 컴퓨팅 능력을 많이 소모하게 된다.

3. 플러딩 오버레이 구조 (Flooding Overlay Structure)

단일의 고정 싱크를 갖는 센서 네트워크에서의 빈번한 플러딩으로 인한 오버헤드를 줄이기 위해 일부 센서 노드들만 브로드캐스트 메시지의 전달에 참여하도록 하는 중앙형 FOS(Centralized FOS; CFOS)와 분산형 FOS(Distributed FOS; DFOS)를 제안한다. FOS 방식은 일부 노드만 브로드캐스트 메시지의 전달에 참여하게 함으로써 포워딩에 참여하지 않는 노드들의 전력 소모를 줄여서 에너지 효율적인 플러딩이 가능하도록 한다. 2장에서 언급한 기존의 다른 효율적인 플러딩 방법들의 경우 메시지를 브로드캐스트할 때

마다 매번 포워딩에 참여할지 여부를 각 센서 노드가 결정하며 이로 인해 센서 노드의 컴퓨팅 파워가 소모되게 된다. 본 논문에서 제안하는 방식에서는 네트워크 초기 설정 시에 FOS 노드가 일단 결정되면 그 이후에는 FOS 노드만 브로드캐스트 메시지 포워딩에 참여하고 FOS 노드가 아닌 센서 노드는 브로드캐스트 메시지 포워딩에 참여하지 않기 때문에 브로드캐스트 메시지 포워딩 여부를 매 브로드캐스트 메시지가 매 결정하지 않아도 되기 때문에 센서 노드의 전력 소모를 줄일 수 있게 된다.

CFOS에서는 싱크가 모든 센서 노드들로부터 토폴로지 정보를 수집하며, 수집한 정보로부터 데이터 포워딩에 참여하게 되는 FOS 노드들을 결정한다. 일반적으로 싱크는 센서 노드에 비해 자원이 덜 제약적이기 때문에 이 경우 싱크에서 FOS 노드를 결정하는 것은 문제가 되지 않으며, 이렇게 함으로써 센서 노드 각자가 플러딩에 참여할지 여부를 결정하는 오버헤드를 줄일 수 있다. CFOS는 전체 네트워크 토폴로지 정보를 사용하기 때문에 DFOS 보다 최적의 FOS 노드를 결정할 수 있다. CFOS에서 싱크는 네트워크 초기화 단계에만 토폴로지 정보를 수집하며, 따라서 정적인 네트워크에 적합하다.

DFOS에서는 각 센서 노드가 자신의 로컬 정보를 이용해서 FOS 노드가 될지 여부를 결정한다. 따라서 DFOS는 동적인 네트워크 환경에 잘 적응할 수 있으며, 단순한 1-홉 이웃 간의 정보 교환만 필요하며, MANET에서의 효율적인 플러딩 방식에서 일반적으로 사용되는 1-홉 이상의 이웃 노드와의 정보 교환이 없기 때문에 무선 링크 자원과 센서 노드 에너지가 한정적인 무선 센서 네트워크 환경에 보다 적합하다.

3.1 중앙형 FOS (Centralized FOS)

CFOS의 동작 과정은 다음과 같다:

1. 네트워크 초기화 시에, 싱크는 전체 네트워크에 TD (Topology Discovery) 메시지를 브로드캐스팅함으로써 각 센서 노드에게 토폴로지 정보를 요청한다.
2. TD 메시지를 수신한 각 센서 노드는 자신의 토폴로지 정보(예, 이웃 노드의 수, 이웃 노드 ID들의 목록 등)를 포함하고 있는 TR(Topology Response) 메시지를 싱크에게 전송한다.
3. 싱크는 수집한 TR 메시지들로부터 토폴로지 정보를 구성하고 이 정보를 기반으로 FOS 노드들을 결정한다. FOS 노드들은 브로드캐스트 패킷을 포워딩하는데 참여하는 노드들이다. 이때, 모든 센서 노드들이 브로드캐스트 패킷을 수신할 수 있도록 보장해주는 방법이 필요하며, 네트워크 연결성을 보장하는데 필요한 최소 FOS 노드를 결정하는 문제는 minimum size connected dominating set 문제^[11]이다. Dominating set은 네트워크의 노드들의 집합으로 네트워크의 모든 노드는 그 집합 내 임의의 노드의 이웃 노드가 되며, 이때 이 집합에 속하는 노드들에 의해서 형성되는 서브그래프가 연결되어 있으면 이 dominating set을 connected dominating set이라고 한다. 최소 크기의 connected dominating set을 구하는 문제는 NP-hard이다^[11]. 따라서 CFOS에서 정하고자 하는 최소

개수의 FOS 노드를 결정하는 문제는 최소 크기의 connected dominating set을 구하는 문제와 동일하다. MANET에서는 임의의 노드가 플러딩을 시작할 수 있으나, 센서 네트워크에서는 일반적으로 싱크가 플러딩을 시작하기 때문에 FOS에 싱크를 포함시킬 필요가 있다. FOS 노드를 결정하는 경험적(heuristic) 알고리즘은 다음과 같다.

```

L1:  $A \leftarrow 0$  // the set of nodes
L2:  $FOS \leftarrow 0$  // the set of FOS nodes
L3:  $Nbr_{sink} \leftarrow \{v : v \text{ lies within the transmission range of sink}\}$ 
L4:  $A \leftarrow A \cup Nbr_{sink}$ 
L5: while  $A$  does not include all nodes
L6:   find a node  $w$  whose  $|Nbr_w - (Nbr_w \cap A)|$  is the maximum among the nodes in  $A$  but not in FOS
L7:    $FOS \leftarrow FOS \cup \{w\}$ 
L8:    $A \leftarrow A \cup Nbr_w$ 
    
```

- 위에서 변수 FOS는 FOS 노드들의 집합이며, Nbr_i 는 센서 노드 i 의 전송 반경 내에 있는 이웃 노드들과 센서 노드 i 를 포함하는 집합이고, A 는 FOS 노드들에 의해서 커버되는 노드들의 집합이다. 위의 알고리즘에서 L4는 싱크로부터의 연결성이 가능하도록 A 의 초기값으로 A 에 싱크의 이웃 노드들을 포함시키는 과정이다. L5부터 L8은 A 에 모든 노드들이 포함될 때까지 (즉, 모든 노드들이 FOS 노드들에 의해서 커버될 때까지(L5)) A 에 속하고 FOS에 속하지 않는 노드들 중에서 A 에 속하지 않는 이웃 노드 수(즉, $|Nbr_w - (Nbr_w \cap A)|$)가 가장 큰 노드를 FOS 노드로 선정하는(L6) greedy 방식으로 동작한다.
4. 싱크는 선택된 FOS 노드들의 목록을 FOS 패킷에 포함시켜 브로드캐스트한다.
 5. FOS 패킷을 수신한 센서 노드는 자신이 FOS 노드인지 여부를 알 수 있으며, 만일 FOS 노드이면 브로드캐스트 패킷의 포워딩에 참여한다. FOS 노드들은 브로드캐스트 패킷을 포워딩하게 함으로써 simple 플러딩의 폭주 문제를 완화시킨다.
 6. FOS 노드의 에너지가 일정 수준 이하로 떨어지면 이웃 노드들에게 RD(Ready-to-Die) 패킷을 1-홉 플러딩으로 전송함으로써 자신의 에너지가 고갈되었음을 알린다. RD 메시지를 수신한 이웃 노드들은 자신의 상태를 FOS 노드로 설정하고 브로드캐스트 패킷의 플러딩에 참여한다. 이렇게 함으로써 FOS 노드의 에너지 고갈로 인한 FOS 구조의 분할을 방지할 수 있다.

3.2 분산형 FOS (Distributed FOS)

DFOS의 동작 과정은 다음과 같다:

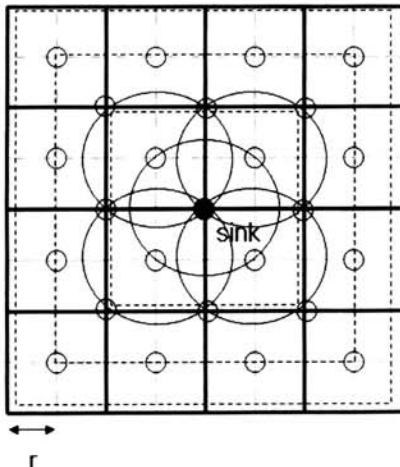
1. 싱크는 전체 네트워크로 TD(Topology Discovery) 메시지를 브로드캐스트 한다.
2. TD 메시지를 수신한 각 센서 노드는 TD_Ack 메시지를 이웃 노드들에게 1-홉 플러딩하며, 일정 개수(threshold)의 TD_Ack 메시지를 수신한 센서 노드는 FOS 후보 노드가 된다. Threshold 값은 다음 알고리즘에 의해서 결정된다.

```

L1: threshold ← 4
L2:  $Nbr_i \leftarrow \{v: v \text{ lies within the transmission range of node } i\}$ 
L3: if  $|Nbr_i| < Node\_density_{avg}$ 
L4:   decrease threshold by  $\delta$ 
L5: else
L6:   increase threshold by  $\delta$ 
    
```

이 알고리즘에서 변수 Nbr_i 는 센서 노드 i 의 전송 반경 내에 있는 이웃 노드들과 센서 노드 i 를 포함하는 노드들의 집합이며, $Node_density_{avg}$ 는 센서 네트워크의 평균 노드 밀도로 센서 네트워크에 살포되는 센서 노드 수와 센서 노드가 살포되는 영역의 면적을 기반으로 계산되어 각 센서 노드에 미리 설정되어 있다고 가정한다. δ 는 threshold의 증감 값을 의미하며, 이웃 노드 수가 평균 노드 밀도보다 작으면(L3) threshold를 δ 만큼 감소시키고(L4) 그렇지 않으면(L5) δ 만큼 증가시킨다(L6).

(그림 1)의 센서 네트워크에서 센서 노드가 균등하게(uniform) 분포되어 있고 노드의 전송 반경이 r , 센서 네트워크 영역의 면적이 $\pi \cdot (m \cdot \sqrt{2}r)^2$ 이라고 가정하면 (여



(그림 1) 무선 센서 네트워크의 예

기서 m 은 싱크로부터 싱크에서 가장 먼 센서 노드로의 홉 수이다), 최소 FOS 노드(그림 1)에서 FOS 노드는 작은

원으로 표시되어 있다)의 수는 $\sum_{i=1}^{\lfloor m/2 \rfloor} 4^i$ 가 된다. 따라서 본 논문에서는 threshold의 초기 값으로 4를 사용한다(L1).

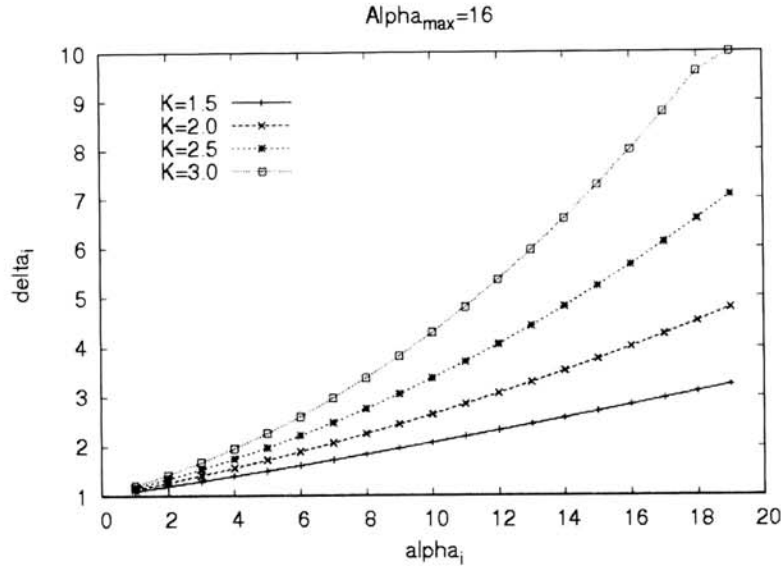
3. DFOS에서 네트워크 연결성을 제공하는 것은 CFOS에서 보다 더 어려우며, 네트워크 연결성을 제공하기 위해서 notice delay 개념을 도입한다. 각 FOS 후보 노드는 이웃 노드들의 수를 기반으로 notice delay를 결정한다. 기 설정된 notice delay 동안 이웃들로부터 Notice 메시지를 수신하지 못한 FOS 후보 노드는 자신의 상태를 FOS 노드로 변경한 후 1-홉 플러딩으로 이웃들에게 Notice 메시지를 송신함으로써 이웃들에게 자신이 FOS 노드가 되었음을 알려준다. 이웃으로부터 Notice 메시지를 수신한 FOS 후보 노드는 자신의 이웃 중에 FOS 노드가 있기 때문에 자신은 FOS 노드가 되는 것을 포기한다. 다음은 FOS 후보 노드가 notice delay를 계산하기 위해 사용하는 수식이다:

```

 $\tau \leftarrow$  average time for 1-hop transmission to succeed
 $\mu\tau \leftarrow$  maximum notice delay, constant  $\mu > 1$ 
 $\delta_i\tau \leftarrow$  notice delay of node  $i$ ,  $1 \leq \delta_i \leq \mu$ 
 $\alpha_i \leftarrow$  number of neighbor nodes of node  $i$ ,  $1 \leq \alpha_i \leq 12$ 
 $\alpha_{max} \leftarrow$  the maximum number of neighbor nodes (a constant greater than 12)
 $\delta_i = \left(\frac{\alpha_i}{\alpha_{max}} + 1\right)^k$ , if  $\delta_i > \mu$ , then  $\delta_i = \mu$ 
 $\delta_i\tau = \left(\frac{\alpha_i}{\alpha_{max}} + 1\right)^k \cdot \tau$ 
    
```

여기서 변수 τ 는 1-홉 전송이 성공적으로 이루어지기 위한 평균 시간이며, $\mu\tau$ 는 최대 notice delay이고, $\delta_i\tau$ 는 노드 i 의 notice delay이다. α_i 는 노드 i 의 이웃 노드 수이고, α_{max} 는 센서 노드들의 이웃 노드 수 중 가장 큰 값을 뜻한다. 어떤 노드의 notice delay는 이웃 노드가 많을수록 커지며 최대 notice delay인 $\mu\tau$ 보다는 커지지 않는다. 위 알고리즘에서 notice delay인 $\delta_i\tau$ 의 성능에 영향을 주는 파라미터들은 k 와 α_i 이고, 따라서 k 와 α_i 값의 변화에 따른 δ_i 값의 변화를 그려보면 다음(그림 2)와 같은 결과를 얻는다. 일반적인 센서 네트워크의 밀도(즉, 어떤 노드의 평균 이웃 노드 수 α_i)는 6~12[14] 이기 때문에 α_{max} 는 12보다 큰 값인 16으로 설정했으며, α_i 가 6~12일 때 δ_i 가 2~3 사이 값을 갖도록 하는 $k = 3$ 을 향후 성능 분석을 하는데 사용한다.

4. FOS 후보 노드가 아닌 센서 노드는 $\mu\tau$ 동안 이웃 노드들로부터 Notice 메시지를 하나도 수신하지 못 하면 FOS 노



(그림 2) α_i 값의 변화에 따른 δ_i 값의 변화

드가 되며, 1-홉 플러딩으로 Notice 메시지를 보냄으로써 이웃 노드들에게 자신이 FOS 노드가 되었음을 알린다.

5. FOS 노드의 에너지 양이 일정 수준 이하로 떨어지면 이웃 노드들에게 RD(Ready-to-Die) 패킷을 1-홉 플러딩으로 전송함으로써 자신의 에너지가 고갈되었음을 알린다. RD 메시지를 수신한 이웃 노드들은 자신의 상태를 FOS 노드로 설정하고 브로드캐스트 패킷의 플러딩에 참여함으로써 FOS 노드의 에너지 고갈로 인한 FOS 구조의 분할을 방지한다.

4. 성능 분석

제안한 FOS 기법의 성능을 분석하기 위해서 NRL 센서 네트워크 확장 패키지^[12]를 포함하는 NS-2 시뮬레이터^[13]를 사용해서 시뮬레이션을 수행했다. 시뮬레이션은 1000m x 1000m 영역에 100개의 센서 노드들이 랜덤하게 분포된 센서 네트워크에 대해서 수행했다. 센서 노드의 전송 반경을 조정함으로써 네트워크 노드 밀도를 6부터 12 사이의 값으로 변경했으며^[14], 노드 밀도는 각 노드의 평균 이웃 노드 수를 전송 영역의 면적으로 나눔으로써 계산했다. 초기 노드 에너지는 5J(Joule)로 설정했으며, 버퍼는 최대 50개까지의 패킷을 저장할 수 있다. 10^{-4} J 보다 작은 가용 노드 에너지는 노드가 더 이상 동작할 수 없음을 뜻하도록 했다. 이벤트는 2초마다 센서 네트워크의 임의의 지역에서 uniform하게 발생하게 했으며, 싱크는 1초마다 질의를 생성하게 했다. 자세한 시뮬레이션 환경은 <표 1>과 같다. 2절의 관련 연구에서 언급한 rumor 라우팅과 SPIN은 FOS 기법과는 다른 데이터 교환 패턴을 갖고 있고, pulse 프로토콜의 성능은 simple 플러딩 주기 값에 의존적이며, TTDD는 센서 노드들이 위치 정보를 알 수 있어야만 하기 때문에, 본 성능 분석에서는 FOS 기법들을 simple 플러딩 및 gossiping과만 비교했다.

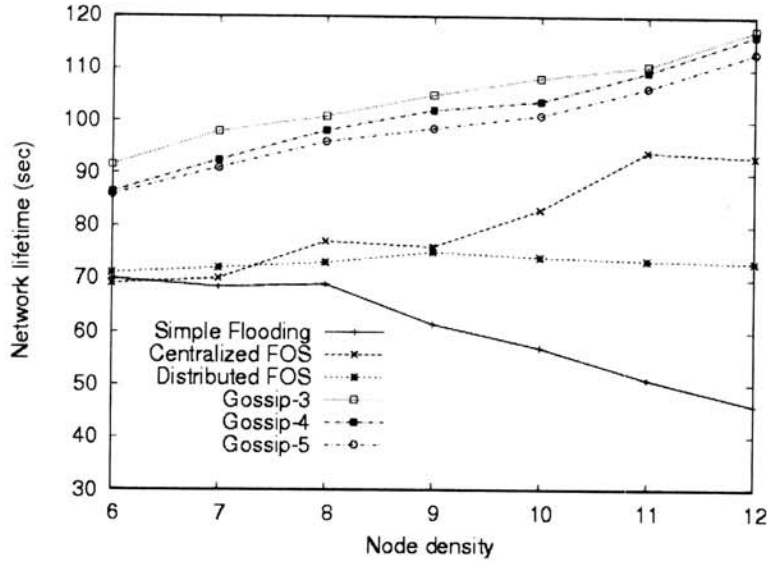
(그림 3)은 노드 밀도의 변화에 따른 네트워크의 수명을

보여준다. 여기서 네트워크의 수명은 에너지가 고갈되어 동작하지 못 하게 되는 노드가 처음으로 발생한 시점까지의 시간을 뜻한다. (그림 1)에 보인 것처럼, 노드 밀도가 증가함에 따라 simple 플러딩에서는 패킷 포워딩 오버헤드가 증가하여 네트워크 수명이 감소한다. 반면, CFOS는 FOS 노드만 브로드캐스트 패킷을 포워딩하기 때문에 노드 밀도가 증가함에 따라 네트워크 수명이 증가한다. DFOS는 CFOS보다 더 많은 포워딩 노드((그림 6) 참조)가 있고 더 많은 메시지((그림 4) 참조)를 생성하기 때문에 CFOS보다 네트워크 수명이 짧다. Gossiping은 (그림 4)에서 볼 수 있듯이 다른 기법들에 비해 적은 수의 메시지를 생성하기 때문에 네트워크 수명이 가장 길다.

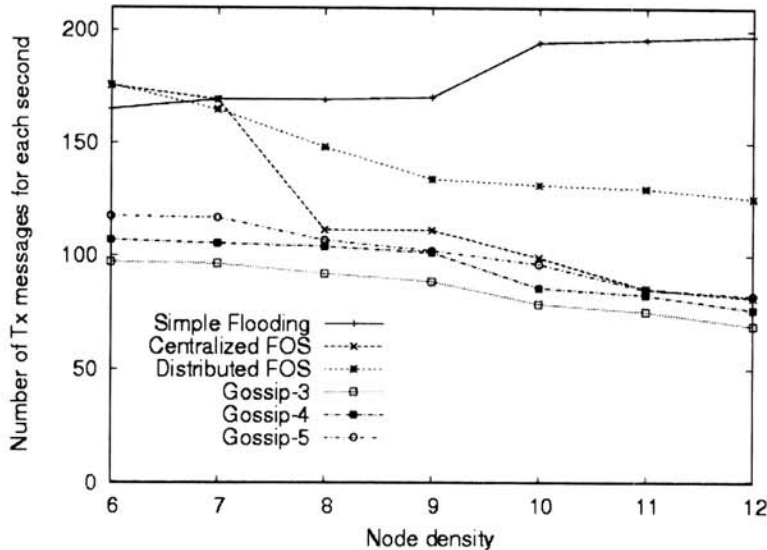
(그림 4)는 1초마다 전송되는 제어 및 데이터 패킷 수를 보여준다. simple 플러딩의 경우는 제어 패킷을 생성하지 않기 때문에 데이터 패킷만 고려했다. simple 플러딩에서 전송된 패킷의 수는 노드 밀도가 증가함에 따라 상당히 많이 증가하며, 그 이유는 모든 노드가 브로드캐스트 패킷을 포워딩하기 때문이다. CFOS는 센서 노드들로부터 수집한 전체 네트워크 토폴로지 정보를 이용해서 FOS 노드들을 선택하기 때문에 분산된 방식으로 FOS 노드를 결정하는 DFOS보다 전송 패킷 수 측면에서 성능이 더 좋다. Gossiping도

<표 1> 시뮬레이션 환경

파라미터	값
Propagation model	Free space model
MAC	IEEE 802.11
라우팅 프로토콜	AODV
네트워크 영역	1000m x 1000m
노드 수	100개
노드 초기 에너지	5J
트래픽 패턴	CBR



(그림 3) 노드 밀도의 변화에 따른 네트워크 수명



(그림 4) 노드 밀도의 변화에 따른 초당 전송되는 패킷 수

포워딩 노드를 적게 선택하기 때문에 전송 패킷 수가 작다.

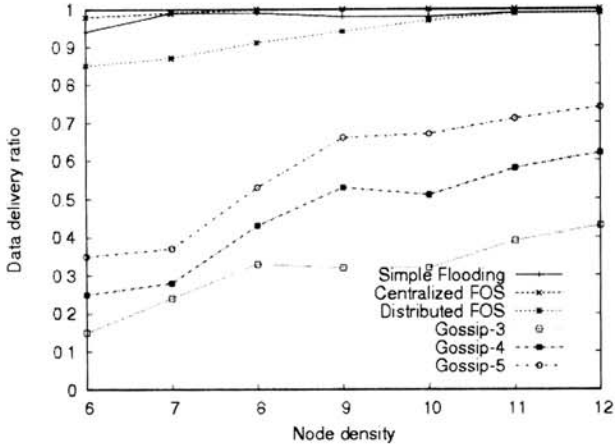
(그림 5)는 노드 밀도의 변화에 따른 데이터 전달율을 보여준다. 시뮬레이션 결과에 의하면 CFOS는 simple 플러딩과 거의 비슷한 성능을 보이며, DFOS는 85% 정도의 데이터를 성공적으로 전달한다. Gossiping은 노드 밀도가 커지면 데이터 전달율도 증가하지만, 기본적으로 데이터 전달율이 1/2 또는 1/3 수준으로 매우 낮기 때문에 simple 플러딩을 대체하기에는 역부족이다.

(그림 6)은 CFOS가 가장 적은 포워딩 노드를 가짐을 보인다. DFOS의 포워딩 노드 수는 3, 4개의 포워딩 노드를 갖는 gossiping의 포워딩 노드 수와 비슷하다. DFOS가 gossiping보다 적은 수의 포워딩 노드들로 훨씬 높은 데이터 전송율을 보임을 알 수 있고, 그 이유는 DFOS가 gossiping보다 효율적으로 포워딩 노드를 선택함을 뜻한다. Simple 플러딩 이외의 다른 기법들에서는 노드 밀도가 증가함에 따라 포워딩 노

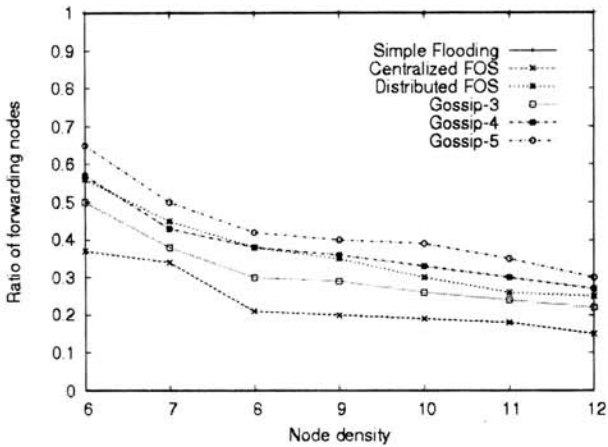
드 수가 감소하며, 따라서 네트워크 수명이 증가하게 된다.

(그림 7)은 노드 밀도의 변화에 따른 전송되는 제어 메시지의 수를 보여준다. CFOS와 DFOS가 상이한 노드 밀도에 대해서 제어 메시지 수 측면에서 성능이 거의 동일함을 알 수 있다. 그 이유는 대부분의 FOS 관련 제어 메시지가 네트워크 초기 단계에 simple 플러딩을 한 번만 사용해서 생성되기 때문이다. DFOS는 CFOS보다 거의 1.5배 더 많은 제어 메시지를 생성한다.

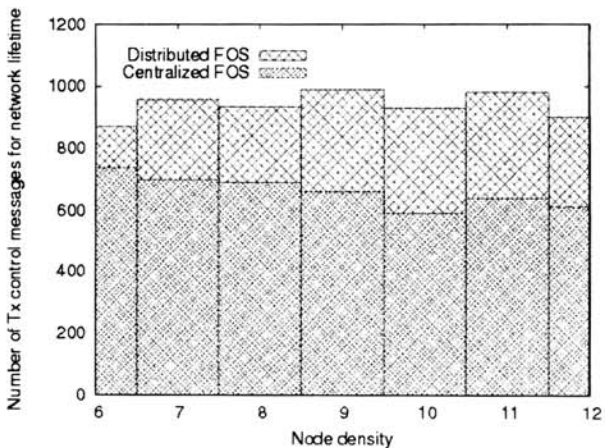
<표 2>는 DFOS에서의 다양한 노드 밀도에 따른 평균 threshold 값을 보여준다. 노드 밀도가 낮으면, 네트워크 연결성을 제공하기 위해서 더 많은 FOS 노드가 필요하며, 따라서 threshold 값이 작아진다. 노드 밀도가 증가하면, 적은 수의 FOS 노드들로 네트워크 연결성을 제공할 수 있기 때문에 threshold 값도 증가한다.



(그림 5) 노드 밀도의 변화에 따른 데이터 전달률



(그림 6) 노드 밀도의 변화에 따른 포워딩 노드의 비율



(그림 7) 노드 밀도의 변화에 따른 초당 전송되는 제어 메시지의 수

5. 결론

무선 센서 네트워크에서 가장 중요한 이슈 중의 하나는 네트워크 수명을 연장시키는 것이다. 센서 네트워크에서 질의나 이벤트를 분배하기 위해서 무선 센서 네트워크는 효율

<표 2> 노드 밀도의 변화에 따른 평균 threshold 값 (DFOS)

Node density	6	7	8	9	10	11	12
Average threshold	3.74	3.84	3.96	4.06	4.08	4.18	4.26

적인 브로드캐스트 기법을 필요로 한다. 가장 단순한 브로드캐스트 기법으로 simple 플러딩이 있으나, simple 플러딩은 폭주와 겹침 문제를 야기해서 자원이 제약된 무선 센서 네트워크의 수명을 감소시킬 수 있다. 따라서 본 논문에서는 플러딩으로 인한 오버헤드를 감소시키기 위해서 오버레이 네트워크 개념을 도입한 플러딩 오버레이 구조(Flooding Overlay Structure; FOS)를 제안했으며, 두 종류의 FOS 기법인 CFOS(Centralized FOS)와 DFOS(Distributed FOS) 기법을 제안했다. CFOS에서는 싱크가 중앙집중형으로 효율적으로 FOS 노드를 결정하기 때문에 정적인 무선 센서 네트워크 환경에 보다 적합하다. 반면, DFOS에서는 각 센서 노드가 자체적으로 브로드캐스트 메시지의 포워딩에 참여할지 여부를 결정한다. 제안된 FOS 방법들의 성능 분석을 위해 NS-2 시뮬레이터를 사용해서 simple 플러딩 및 gossiping과의 성능 비교를 수행했다. 시뮬레이션 결과, FOS 방법들이 simple 플러딩 방법에 비해서 네트워크 수명 측면에서 우월함을 알 수 있었으며, 데이터 전달률 측면에서는 FOS 방법들이 simple 플러딩과 성능이 거의 비슷하거나 gossiping보다 훨씬 우수함을 알 수 있었다.

참고 문헌

- [1] G. J. Pottie and W. J. Kaiser, "Wireless Integrated Network Sensors," Communications of the ACM, Vol.43, No.5, pp. 51-58, May, 2000.
- [2] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," ACM Mobicom, pp.56-67, Aug., 2000.
- [3] D. Braginsky and D. Estrin, "Rumor Routing Algorithm for Sensor Networks," ACM WSNA, pp.22-31, Sept., 2002.
- [4] B. Awerbuch, D. Holmer, H. Rubens, K. Chang and I.-J. Wang, "The Pulse Protocol: Sensor Network Routing and Power Saving," IEEE Milcom, pp.662-667, Nov., 2004.
- [5] J. Kulik, W. Heinzelman and H. Balakrishnan, "Negotiation-based Protocols for Disseminating Information in Wireless Sensor Networks," Wireless Networks, Vol.8, pp.169-185, 2002.
- [6] M.-J. Lin, K. Marzillo and S. Masini, "Gossip versus Deterministic Flooding: Low Message Overhead and High Reliability for Broadcasting on Small Networks," DISC, pp. 253-267, 2000.
- [7] F. Ye, H. Luo, J. Cheng, S. Lu and L. Zhang, "A Two-Tier Data Dissemination Model for Large-scale Wireless Sensor Networks," ACM Mobicom, pp.148-159, Sept., 2002.
- [8] F. Ingelrest, D. Simplot-Ryl and I. Stojmenovic, "Energy-

Efficient Broadcasting in Wireless Mobile Ad Hoc Networks," Resource Management in Wireless Networking, Kluwer Academic Publishers, 2004.

- [9] W. Lou and J. Wu, "Toward Broadcast Reliability in Mobile Ad Hoc Networks with Double Coverage," IEEE Transactions on Mobile Computing, Vol.6, No.2, pp.148-163, Feb., 2007.
- [10] S. Ashish, "On the Reduction of Broadcast Traffic in Mobile Ad Hoc Networks," IEEE WiCom, pp.1581-1584, Sept., 2007.
- [11] C. Adjib, P. Jacquet and L. Viennot, "Computing Connected Dominated Sets with Multipoint Relays," Technical Report 4597, INRIA, Oct., 2002.
- [12] NRL's Sensor Network Extension to NS-2, <http://nrlsensorsim.pf.itd.nrl.navy.mil/>
- [13] The Network Simulator, NS-2, <http://www.isi.edu/nsnam/ns/>
- [14] Y. Liu, L.M. Ni, and M. Li, "A Geography-free Routing Protocol for Wireless Sensor Networks", IEEE HPSR, pp. 351-355, May, 2005.



안 상 현

e-mail : ahn@uos.ac.kr
 1986년 서울대학교 컴퓨터공학과(학사)
 1988년 서울대학교 컴퓨터공학과(석사)
 1989년 University of Minnesota 컴퓨터학과(박사)
 1988년 (주)데이콤 연구원

1994년 세종대학교 컴퓨터학과 교수
 1998년~현 재 서울시립대학교 컴퓨터과학부 교수
 관심분야 : 애드혹 네트워크, 센서 네트워크, 홈 네트워크, 이동 통신, 라우팅 프로토콜 등



임 유 진

e-mail : yujin@suwon.ac.kr
 1995년 숙명여자대학교 전자계산학과(학사)
 1997년 숙명여자대학교 전자계산학과(석사)
 2000년 숙명여자대학교 전자계산학과(박사)
 2000년 서울대학교 Post-Doc
 2000년 서울시립대학교 연구교수

2003년 University of California Los Angeles, Post-Doc
 2003년 삼성종합기술원 전문연구원
 2004년~현 재 수원대학교 정보미디어학과 전임강사
 관심분야 : 센서 네트워크, 애드혹 네트워크, 홈 네트워크 등



김 만 희

e-mail : esterss@naver.com
 2008년 수원대학교 정보미디어학과(학사)
 2009년~현 재 서울시립대학교 컴퓨터통계학과 석사과정
 관심분야 : 이동 애드혹 네트워크, 차량 애드혹 네트워크, 라우팅 프로토콜