

이동 Ad Hoc 네트워크에서의 Flooding 기술

Flooding Techniques for Mobile Ad Hoc Networks

신재욱(J.W. Shin)

무선Ad Hoc연구팀 선임연구원

권혜연(H.Y. Kwon)

무선Ad Hoc연구팀 선임연구원

김응배(E.B. Kim)

무선Ad Hoc연구팀 책임연구원, 팀장

이동 Ad Hoc 네트워크에서 플러딩(flooding)은 제어 및 데이터 패킷을 브로드캐스팅하기 위해 주로 사용되고 있지만, 노드들간의 공유된 무선 채널 사용 및 무선 전송영역의 중첩으로 인하여 플러딩 수행 시에 발생하는 자원 경쟁, 패킷 충돌 및 패킷 중복 수신현상은 네트워크의 성능을 저하시키는 요인이 되고 있다. 본 고에서는 이동 Ad Hoc 네트워크에서 패킷 플러딩 수행 시에 발생하는 문제점을 기술하고 이를 해결하기 위해 제안된 다양한 플러딩 기법들을 소개한다.

I. 서론

이동 Ad Hoc 네트워크는 무선 인터페이스를 가진 다수의 노드들에 의해 자율적으로 구성되는 네트워크로서 무선 자원과 배터리 상의 제약을 가진다. 또한, 노드의 이동성으로 인하여 네트워크 토폴로지(topology)가 동적으로 변화하는 특성을 가지기 때문에 기존 유선 네트워크에서 사용되는 패킷 전달방식을 그대로 적용하기에는 많은 문제점이 있다.

플러딩은 패킷을 네트워크 전체로 브로드캐스팅하기 위한 패킷 전달방식으로, 이동 Ad Hoc 네트워크에서는 라우팅 프로토콜, 서비스 탐색 프로토콜, 주소 자동설정 프로토콜 등에서 이를 필수적으로 사용하고 있기 때문에 플러딩 기법의 효율성은 이동 Ad Hoc 네트워크의 성능을 결정짓는 주요 요소의 하나로 작용하고 있다. 특히, 이동 Ad Hoc 네트워크에서 모든 노드들은 서로 공유된 무선 채널을 사용하고 있고 인접 노드들간의 무선 전송영역이 서로 중첩되는 특성을 가지고 있기 때문에 플러딩 수행중에 무선 자원에 대한 경쟁(contention), 패킷 충돌

(collision) 및 브로드캐스트 패킷의 중복 수신(duplication)에 따른 네트워크 성능 저하가 큰 문제점으로 인식되고 있다[1],[2].

본 고에서는 이동 Ad Hoc 네트워크에서 패킷 플러딩 시에 발생하는 문제점을 기술하고, 이를 해결하기 위해 제안된 다양한 플러딩 기법들을 소개한다.

II. 이동 Ad Hoc 네트워크에서의 플러딩

1. 필요성

인터넷에서는 노드의 IP 주소를 보고 해당 노드가 어떤 서브넷에 접속되어 있는지를 알 수 있기 때문에 주소는 노드를 식별하는 용도뿐만 아니라 노드의 물리적인 위치를 판단할 수 있는 기준이 된다. 그러나, 이동 Ad Hoc 네트워크에서는 노드의 이동으로 인하여 네트워크 토폴로지가 동적으로 변화하는 특성을 가지기 때문에 노드의 주소는 단지 노드를

유일하게 식별하는 용도로만 사용되며 위치 정보를 포함하고 있는 것은 아니다. 이와 같은 환경에서 플러딩은 특정 노드로의 루트(위치)를 탐색할 경우에 유용하게 사용될 수 있다. 이와 같은 라우팅 프로토콜을 포함하여, 이동 Ad Hoc 네트워크에서 플러딩이 사용되고 있는 경우는 다음과 같다.

- 테이블 기반(table-driven) 라우팅 프로토콜에서 네트워크 토폴로지 정보 전달을 위한 제어 메시지를 네트워크 전체로 브로드캐스팅하는 경우
- 요구 기반(on-demand) 라우팅 프로토콜에서 루트 탐색을 위한 루트 요청(route request, RREQ) 메시지를 네트워크 전체로 브로드캐스팅하는 경우
- 멀티캐스트 프로토콜에서 제어 메시지를 네트워크 전체로 브로드캐스팅하는 경우
- 주소 자동 설정(address auto-configuration) 프로토콜에서 중복된 주소 검출(duplicated address detection)을 위한 제어 메시지를 네트워크 전체로 브로드캐스팅하는 경우
- 서비스 탐색(service discovery) 프로토콜에서 서비스 요청 메시지를 네트워크 전체로 브로드캐스팅하는 경우
- 멀티캐스트 또는 브로드캐스트 데이터 패킷을 전달하는 경우

2. 특성

이동 Ad Hoc 네트워크에서 노드들은 일반적으로 하나의 공유된 채널을 통해서 통신하고 있다. 예를 들어, IEEE 802.11에서 제공하는 Ad Hoc 모드 통신의 경우 Carrier Sense Multiple Access/Collision Avoidance(CSMA/CA) 방식에 의해 다수의 사용자가 하나의 무선 채널을 공유하기 때문에 사용자 간에 무선 채널에 대한 경쟁이 발생한다. 특히, 브로드캐스트 패킷의 경우는 유니캐스트 패킷과 달리 Hidden Terminal 문제 해결을 위한 RTS/CTS 메커니즘을 적용하고 있지 않기 때문에 패킷 충돌 현상이 빈번히 발생할 수 있다. 또한, 인접한 노드는 무선 전

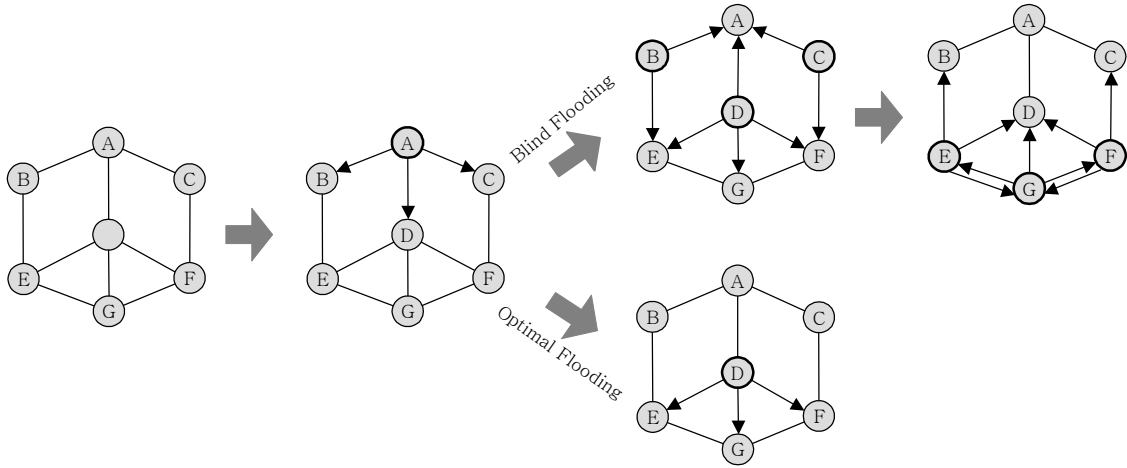
송 영역이 서로 중첩되기 때문에 중복된 패킷의 수신 이 가능하다. 이와 같은 문제점을 완화시키기 위해 브로드캐스트 패킷을 전송할 경우 일반적으로 다음과 같은 메커니즘을 기본적으로 적용하고 있다[1].

- ① 브로드캐스트 패킷을 초기 생성한 소스 노드의 주소와 패킷 식별자를 저장함으로써 동일 브로드캐스트 패킷의 중복 수신 여부를 판단하며, 중복 수신된 브로드캐스트 패킷은 폐기한다.
- ② 최초 수신된 브로드캐스트 패킷을 다시 플러딩하기 전에 일정 지연 시간을 두며, 이 시간 동안에 중복적으로 수신된 브로드캐스트 패킷의 정보에 따라 플러딩 여부를 다시 결정한다.
- ③ 브로드캐스트 패킷을 플러딩하기 전에 임의의 지연 시간을 두으로써 이웃 노드간의 동시 플러딩에 의한 패킷 충돌을 방지한다.

3. 문제점

플러딩의 가장 기본적인 형태인 블라인드 플러딩(blind flooding)은 브로드캐스트 패킷을 수신한 노드가 이를 다시 이웃 노드에게 플러딩하는 방식이다. 네트워크 상에 N개의 노드가 존재할 경우 블라인드 플러딩을 사용하면 N번의 플러딩이 발생한다. (그림 1)에서와 같이 7개의 노드로 구성된 이동 Ad Hoc 네트워크에서 블라인드 플러딩을 사용할 경우 7번의 플러딩을 수행해야 한다. 그러나, 노드들의 무선 전송영역이 서로 겹치기 때문에 최적화된 플러딩(optimal flooding)의 경우 2번의 플러딩만으로 모든 노드들에게 브로드캐스트 패킷을 전달할 수가 있다. 블라인드 플러딩은 구현이 용이한 장점이 있는 반면 브로드캐스트 패킷을 중복되게 수신할 가능성이 크며 이웃 노드간의 자원 경쟁 및 패킷 충돌 현상을 발생시킴으로써 네트워크 성능에 심각한 영향을 주게 된다. 이와 같은 현상을 브로드캐스트 스톰(broadcast storm) 문제라고 한다[1],[2].

주어진 네트워크 토폴로지 상에서 최적의 플러딩을 수행하는 것은 Minimal Connected Dominant Set(MCDS)을 계산하는 문제에 해당된다. Domi



(그림 1) Blind Flooding과 Optimal Flooding의 비교

nant Set이란 모든 노드가 이 집합에 속하거나 또는 이 집합에 속하는 임의의 노드와 1-홉 이웃 노드인 경우에 해당된다. Dominant Set에 속하는 모든 노드들이 서로 연결되어 있을 경우를 Connected Dominant Set(CDS)이라고 하며, 최소 수의 노드로 구성된 CDS를 MCDS라고 한다. MCDS에 속하는 노드들만이 플러딩을 수행하게 함으로써 최적의 플러딩이 가능하다. 그러나, MCDS를 구하는 것은 NP-Complete 문제로 분류되기 때문에 이를 직접 계산하기는 어렵다[3],[4]. 또한, 이동 Ad Hoc 네트워크에서는 노드의 이동으로 인하여 네트워크 토폴로지가 수시로 변하기 때문에 글로벌한 토폴로지 정보에 기반한 최적의 플러딩을 더욱 어렵게 하고 있다.

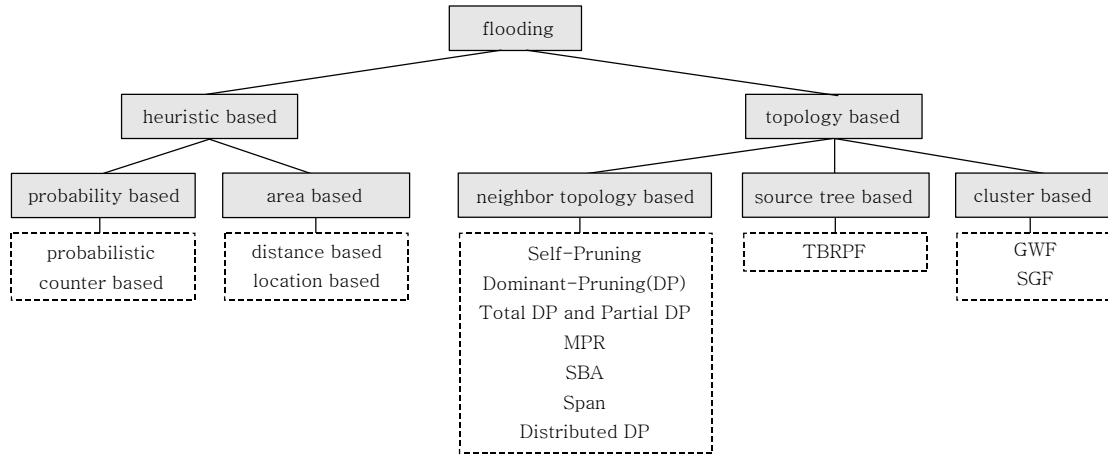
이동 Ad Hoc 네트워크를 위한 효율적인 플러딩 방식의 설계는 에너지 또는 전송대역과 같은 제한된 자원을 절약하고 브로드캐스트 스톱과 같은 문제를 해결하고자 하는 데에 목적이 있다. 브로드캐스트 패킷을 실제로 플러딩하는 데 참여한 노드의 비율이 플러딩 방식의 효율성을 판단하는 기준으로 사용될 수 있으며, 브로드캐스트 패킷이 네트워크 노드 전체에 수신되기까지 소요되는 시간 및 브로드캐스트 패킷이 전달되지 않은 노드의 비율 등이 성능 평가 요소로 함께 사용될 수 있다.

III. 주요 플러딩 기술

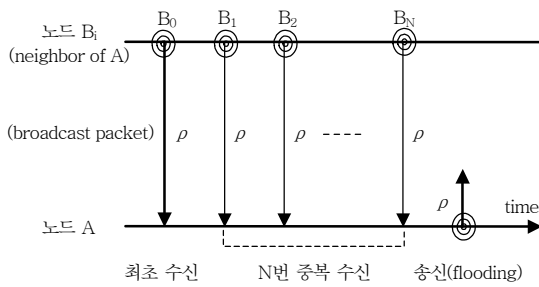
(그림 2)에서와 같이 플러딩 방식은 중복적으로 수신된 브로드캐스트 패킷의 정보를 이용하는 휴리스틱 기반(heuristic based) 방식과 토폴로지 정보를 이용하는 토폴로지 기반(topology based) 방식으로 크게 나눌 수 있다. 휴리스틱 기반 방식은 확률 기반(probability based) 방식과 영역 기반(area based) 방식으로 나눌 수 있으며, 토폴로지 기반 방식은 이웃 토폴로지 기반(neighbor topology based) 방식, 소스 트리 기반(source tree based) 방식 및 클러스터 기반(cluster based) 방식으로 나눌 수 있다[5],[6].

1. 휴리스틱 기반 방식

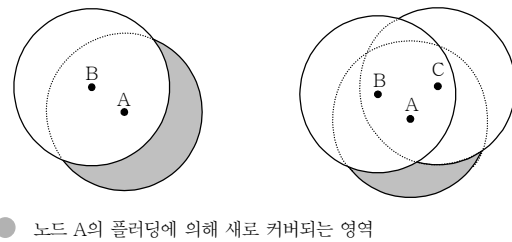
노드 A가 이웃 노드 B_i로부터 수신한 브로드캐스트 패킷을 자신의 다른 이웃 노드에게 플러딩하기 위해서는 해당 패킷을 MAC 계층으로 전달하여야 하며, MAC 계층에서는 공유된 무선 채널이 가용할 경우 물리 계층에게 전달하여 실제 전송이 이루어지게 한다. 따라서, (그림 3)에서와 같이 노드 A가 브로드캐스트 패킷을 최초로 수신한 시각과 해당 브로드캐스트 패킷을 물리 계층에서 실제로 송신하기까지는 다소의 시간이 소요되며, 이 시간 동안에 다수



(그림 2) 플러딩 방식의 분류



(그림 3) 브로드캐스트 패킷의 중복 수신



● 노드 A의 플러딩에 의해 새로 커버되는 영역
 (a) 브로드캐스트 패킷을 한 번 수신한 경우
 (b) 브로드캐스트 패킷을 두 번 수신한 경우

(그림 4) 브로드캐스트 패킷의 중복 수신에 의한 효과

의 중복된 브로드캐스트 패킷이 수신될 수 있다. 따라서, 이 기간 동안 중복 수신된 브로드캐스트 패킷의 정보를 이용하여 플러딩 여부를 다시 결정한다 [1],[2].

가. 확률 기반 방식

노드 A와 노드 B가 서로 이웃 노드일 경우 두 노드의 무선 전송영역은 서로 겹치게 된다. (그림 4a)는 노드 A가 브로드캐스트 패킷을 노드 B로부터 최초로 수신한 후 이를 다시 플러딩할 경우를 나타낸다. 노드 A에 의해 새로 커버되는 영역은 노드 A의 무선 전송영역에서 노드 B의 무선 전송영역을 제외한 부분으로 계산되며, 이 영역의 면적은 노드 A의 전체 전송영역 대비 최대 61%, 평균 41%로 계산된다[1],[2]. 즉, 노드 A의 이웃 노드 중에서 노드 B가 플러딩할 때 브로드캐스트 패킷을 수신한 노드의

비율은 평균 59%이고 최소 39%이다. (그림 4b)는 노드 A가 노드 B와 노드 C로부터 동일한 브로드캐스트 패킷을 수신한 후 이를 다시 플러딩할 경우를 나타낸다. 노드 A의 플러딩에 의해 새로 커버될 수 있는 영역은 최대 19%로 계산된다[1],[2]. 확률 기반 플러딩 방식은 이와 같이 동일한 브로드캐스트 패킷을 중복 수신한 횟수가 커질수록 해당 브로드캐스트 패킷을 다시 플러딩하여 얻을 수 있는 효과는 급격히 감소한다는 사실에 기반하고 있다. 확률 기반 플러딩 방식에는 확률적(probabilistic) 방식과 계수 기반(counter based) 방식이 있다[1],[2].

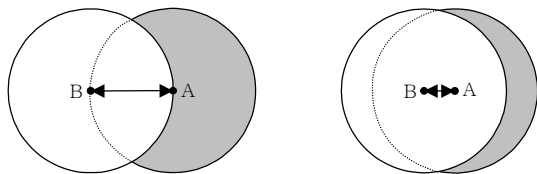
확률적 방식에서는 노드가 브로드캐스트 패킷을 처음으로 수신할 경우 기준 확률 P에 기반하여 플러딩 여부를 결정한다. P = 1인 경우는 블라인드 플러딩에 해당된다. 계수 기반 방식에서는 확률이 아닌

브로드캐스트 패킷의 중복 수신 횟수를 사용하여 플러딩 여부를 결정한다. 즉, 노드 A가 브로드캐스트 패킷을 수신하고 이를 다시 플러딩하기 전에 다른 이웃 노드로부터 중복적으로 수신된 브로드캐스트 패킷의 숫자가 일정 임계값(threshold) 보다 크면 플러딩을 취소한다. 확률기반 방식은 가장 간단하게 구현할 수 있으나 기준 확률값과 임계값을 어떻게 결정하는냐가 성능에 중요한 요소로 작용한다.

나. 영역 기반 방식

두 노드의 위치가 가까울수록 이들이 커버하는 무선 전송 영역의 중첩 부분은 더 커지게 된다. (그림 5a)는 노드 A와 노드 B가 이웃 노드로서 가장 멀리 떨어진 경우를 나타내며 (그림 5b)는 그렇지 않을 경우를 나타낸다. 노드 A와 노드 B의 위치가 동일하다면 두 노드의 무선 중첩영역은 서로 겹치게 된다. 따라서, 중첩된 영역의 크기가 작아질수록 플러딩의 효과는 미비해진다. 영역 기반 플러딩은 이와 같은 노드의 전송 영역에 기반하고 있으며 거리 기반(distance based) 플러딩과 위치 기반(location based) 플러딩이 있다[1],[2].

거리 기반방식에서는 노드 A가 이웃 노드로부터 중복적으로 브로드캐스트 패킷을 수신한 경우, 해당 브로드캐스트 패킷을 플러딩한 이웃 노드 중 노드 A와 가장 가까이 위치한 노드와의 거리가 임계값 이하이면 플러딩을 취소하는 방식이다. 노드간의 거리는 수신된 신호의 세기로 계산할 수 있다. 각 노드의



● 노드 A의 플러딩에 의해 새로 커버되는 영역
 (a) 노드 A와 노드 B의 거리가 최대인 경우 (b) 노드 A와 노드 B의 거리가 최대가 아닌 경우

(그림 5) 두 노드간의 거리에 따른 전송 영역의 중첩

위치를 GPS 등을 사용하여 파악이 가능하다면 동일 브로드캐스트 패킷을 중복 수신한 상태에서 노드 A가 플러딩을 수행했을 경우 새로 커버될 수 있는 영역의 크기를 수학적으로 계산할 수 있다. 위치 기반 방식에서는 이 크기가 일정 임계치 이하이면 플러딩을 취소하는 방식을 사용하고 있다.

2. 토폴로지 기반 방식

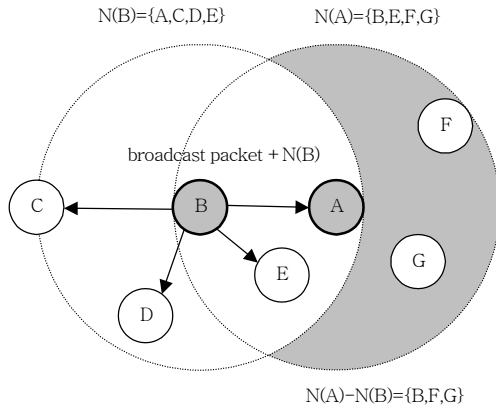
토폴로지 기반 방식에서는 네트워크 토폴로지 정보를 이용하여 브로드캐스트 패킷의 플러딩 횟수를 최소화하고자 하며, 대부분이 MCDS를 근사적으로 계산하기 위한 다양한 기법들을 제안하고 있다.

가. 이웃 토폴로지 기반 방식

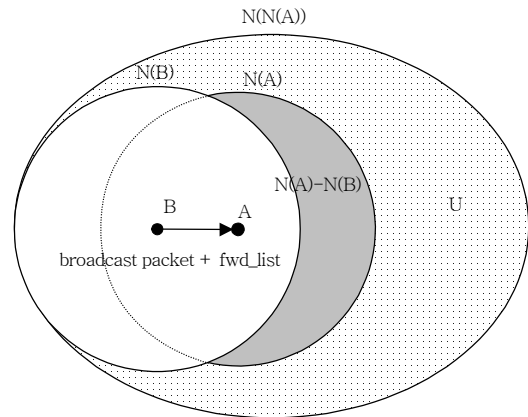
1-홉 또는 2-홉 거리에 위치한 이웃 노드의 정보를 이용하여 브로드캐스트 패킷의 플러딩 횟수를 최소화하고자 한다. 대부분의 방식이 2-홉 이웃 노드 정보를 이용하여 MCDS에 근사한 CDS를 계산하고, CDS에 포함된 노드(포워드 노드)들을 통해서 브로드캐스트 패킷을 플러딩하고자 한다.

1) Self-Pruning

Self-Pruning[3] 방식에서는 1-홉 이웃 노드의 토폴로지 정보를 이용하여 플러딩을 수행한다. Self-Pruning 방식의 경우, 모든 노드는 계층-2의 BEACON 신호 또는 계층-3의 HELLO 메시지의 주기적인 브로드캐스트를 통해 자신과 1-홉 거리에 있는 이웃 노드에 대한 리스트를 관리한다. 노드가 브로드캐스트 패킷을 플러딩할 경우 자신의 이웃 노드 리스트를 패킷에 포함시킨다. (그림 6)에서와 같이 노드 A가 노드 B로부터 브로드캐스트 패킷을 수신할 경우 이 패킷에 포함된 노드 B의 이웃 노드 리스트인 $N(B)$ 를 현재 자신의 이웃 노드 리스트 $N(A)$ 와 비교하여, $N(A) - N(B) - \{B\} \neq \emptyset$ 인 경우에 해당 브로드캐스트 패킷을 다시 플러딩한다. 이 때 자신의 이웃 노드 리스트인 $N(A)$ 를 브로드캐스트 패킷에 포함시킨다.



(그림 6) Self-Pruning을 이용한 플러딩



(그림 7) Dominant-Pruning에 의한 플러딩

이 방법은 브로드캐스트 패킷에 이웃 노드 리스트를 포함시키기 때문에 패킷의 크기가 커지는 문제점이 있다. 이를 해결하기 위한 방법으로, 노드가 브로드캐스트 패킷을 수신할 경우 패킷에 포함된 인접 노드 리스트를 저장하고, 브로드캐스트 패킷을 플러딩할 경우 이전에 브로드캐스트 패킷을 플러딩한 이후 이웃 노드 리스트에 변경이 없으면 이웃 노드 리스트를 브로드캐스트 패킷에 포함시키지 않는 방법을 사용할 수 있다. 이웃 노드 리스트가 포함되지 않은 브로드캐스트 패킷을 수신한 노드는 이전에 저장된 이웃 노드 리스트를 그대로 사용한다.

2) Dominant-Pruning(DP)

HELLO 메시지에 1-홉 이웃 노드 리스트를 포함시켜 전달함으로써 이를 수신한 노드가 자신의 2-홉 이웃 노드 정보를 알 수 있게 한다. Self-Pruning 방식에서는 브로드캐스트 패킷을 수신한 노드가 패킷의 플러딩 여부를 결정하였지만, Dominant-Pruning[3] 방식에서는 브로드캐스트 패킷을 송신하는 노드가 이를 수신한 이웃 노드에서의 플러딩 여부를 결정한다. 패킷을 포워딩할 수 있는 1-홉 이웃 노드의 리스트는 포워드 리스트(forward list) F 로서 브로드캐스트 패킷에 포함된다. 노드 A가 이웃 노드 B로부터 브로드캐스트 패킷을 수신하고 노드 A의 주소가 포워드 리스트에 포함되어 있을 경우 해당 브로드캐스트 패킷을 다시 플러딩할 수 있다.

플러딩을 수행하기 전에 노드 A는 자신의 2-홉 노드를 모두 커버할 수 있는 최소의 포워드 리스트를 계산하여 브로드캐스트 패킷의 헤더에 포함시킨다.

(그림 7)과 같이 노드 A로부터 1-홉 거리에 있는 이웃 노드의 집합을 $N(A)$ 로 표기할 때 노드 A로부터 2-홉 이내에 존재하는 모든 노드의 집합은 $N(N(A))$ 로 표기할 수 있다. $N(N(A))$ 에 속하는 노드 중 B, A, $N(B)$ 는 노드 B의 플러딩에 의해 이미 브로드캐스트 패킷을 수신하였으며, $N(A)$ 는 A가 플러딩할 때 패킷을 수신하게 될 것이다. 따라서, 노드 A는 $U = N(N(A)) - N(B) - N(A)$ 에 포함된 모든 노드가 패킷을 수신할 수 있도록 $N(A)$ 로부터 포워드 리스트를 결정하여야 한다. $N(N(B))$ 에 속하는 노드는 B가 패킷을 플러딩할 때 이미 패킷을 수신할 수 있도록 보장 받기 때문에 $N(A) - N(B)$ 에 속하는 노드 중에서 포워드 리스트를 결정하면 된다. 따라서, $\bigcup_{c \in F} (N(c) \cap U) = U$ 인 최소의 크기를 갖는 $F \subseteq N(A) - N(B)$ 를 결정한다. F 를 계산하는 것은 Set Cover 문제와 동일한 것으로서 NP-complete에 속하기 때문에 Greedy Set Cover 알고리즘을 사용하여 이를 근사적으로 계산하고 있다.

3) Total DP and Partial DP

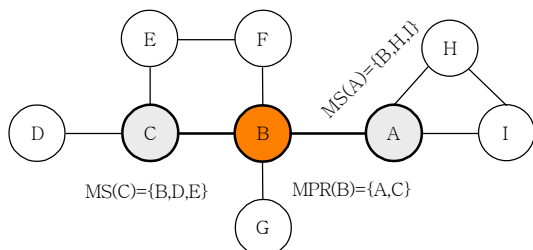
Dominant-Pruning 방식의 성능 향상을 위해 1-홉 이웃 노드에 의해 커버되어야 하는 2-홉 이웃 노드의 개수를 감소시키기 위한 방법으로 Total

Dominant Pruning(TDP)과 Partial Dominant Pruning(PDP) 방식이 제안되고 있다[4]. TDP에서는 노드 B가 브로드캐스트 패킷을 플러딩할 때 $N(N(B))$ 를 패킷 헤더에 포함시켜 전달한다. 노드 A가 노드 B로부터 브로드캐스트 패킷을 수신하면 $U = N(N(A)) - N(B) - N(A)$ 대신에 $U = N(N(A)) - N(N(B))$ 를 사용함으로써 U의 크기를 좀 더 줄일 수 있다. PDP에서는 2-홉 이웃 노드 정보를 브로드캐스트 패킷에 포함시키지 않고 Dominant Pruning에서와 동일한 이웃 노드 정보를 이용한다. 그러나, $U = N(N(A)) - N(B) - N(A)$ 대신에 $U = N(N(A)) - N(B) - N(A) - N(N(A) \cap N(B))$ 를 사용함으로써 U의 크기를 감소시키고 있다. U의 크기가 작아지면 F의 크기도 감소할 가능성이 크다.

4) Multipoint Relay(MPR)

Optimized Link State Routing(OLSR) 라우팅 프로토콜에서는 Multipoint Relay(MPR)에 기반한 플러딩 방식을 제안하고 있다[7]. 모든 2-홉 이웃 노드를 커버할 수 있는 1-홉 이웃 노드의 집합을 MPR Set이라고 하며, MPR Set에 속하는 노드를 MPR이라고 한다. 최소 수의 노드로 구성된 MPR Set을 계산함으로써 플러딩 오버헤드를 최소화하고자 한다. (그림 8)에서와 같이 노드 A가 노드 B로부터 브로드캐스트 패킷을 수신한 경우 노드 A가 노드 B의 MPR에 해당될 경우에만 해당 브로드캐스트 패킷을 다시 플러딩한다.

주기적으로 전달되는 HELLO 메시지를 이용하여 자신의 1-홉 이웃 노드 정보와 자신의 MPR Set을 이웃 노드에게 전달한다. 따라서, 이를 수신



(그림 8) MPR과 MPR Selector Set

한 노드는 1-홉 이웃 노드의 1-홉 이웃 노드, 즉 자신의 2-홉 이웃 노드에 대한 정보를 얻을 수 있으며 이를 통해 자신의 MPR Set을 계산할 수 있다. 노드 A가 노드 B의 MPR일 때 노드 B는 노드 A의 MPR Selector Set(MS)에 속한다. 노드 A가 노드 B로부터 HELLO 메시지를 수신하였을 때, 이 메시지에 포함된 MPR Set에 노드 A의 주소가 존재하면 노드 A는 노드 B의 주소를 자신의 MS에 포함시킨다. MS에 속한 노드로부터 브로드캐스트 패킷이 수신될 경우에만 이를 다시 플러딩한다.

$N(x)$ 를 노드 x의 1-홉 이웃 노드 집합, $N^2(x)$ 를 노드 x의 2-홉 이웃 노드 집합이라고 할 때 노드 x에서 다음과 같은 휴리스틱한 방법을 사용하여 MPR 집합을 계산한다.

- ① $MPR(x) = \{ \}$ // MPR Set of x
- ② 노드 $N^2(x)$ 에 속하는 특정 노드의 유일한 이웃 노드가 $N(x)$ 에 존재할 경우 이 1-홉 이웃 노드를 $MPR(x)$ 에 포함시킨다.
- ③ $MPR(x)$ 에 의해 커버되지 않는 노드가 아직 $N^2(x)$ 에 존재할 경우, 이들 2-홉 이웃 노드를 최대한 커버하는 $N(x)$ 에 속하는 1-홉 이웃 노드를 찾아 $MPR(x)$ 에 포함시킨다.
- ④ 모든 2-홉 이웃 노드가 커버될 때까지 ②~③ 과정을 반복한다.

5) Scalable Broadcast Algorithm(SBA)

SBA[8]는 Self-Pruning 방법과 유사하나 HELLO 메시지를 이용하여 2-홉 이웃 노드 정보를 획득한다. 노드 A가 노드 B로부터 브로드캐스트 패킷을 수신한 경우 $N(A) \subseteq N(B) \cup \{B\}$ 인 경우에는 해당 브로드캐스트 패킷(m)을 다시 플러딩하지 않는다. 그렇지 않을 경우, 브로드캐스트 패킷을 다시 플러딩하는데, 즉시 플러딩하는 대신에 일정 시간의 백오프 지연(back-off delay)을 가진다. $\lambda(A, m) = N(B) \cup \{B\}$ 라고 정의할 때 백오프 지연 시간 동안에 다른 이웃 노드 C로부터 동일 브로드캐스트 패킷이 다시 수신되면 $\lambda(A, m) = \lambda(A, m) \cup N(C) \cup \{C\}$

로 설정하고 수신된 브로드캐스트 패킷은 폐기(discard)한다. 백오프 지연 시간이 경과된 후 $N(A) \leq \lambda(A, m)$ 이면 플러딩을 취소하고 그렇지 않으면 플러딩을 수행한다. 이후에 수신되는 브로드캐스트 패킷은 무시된다. 백오프 지연은 이웃 노드 개수에 기반하여 동적으로 적용 가능하다. 즉, 이웃 노드 수가 많을수록 이 값을 작게 설정함으로써 이웃 노드보다 먼저 패킷을 플러딩하여 다른 이웃 노드에 의한 중복된 플러딩을 방지할 수 있다.

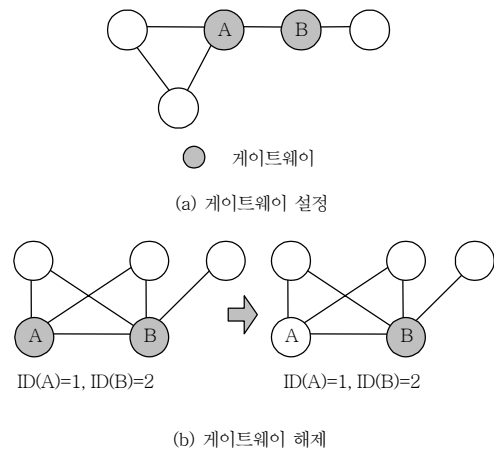
6) Span

Span[9]에서는 이웃 노드 정보에 기반하여 노드 상태를 coordinator 또는 non-coordinator로 설정하며, coordinator 상태의 노드는 포워드 노드의 역할을 수행한다. 각 노드에서 coordinator와 non-coordinator 간의 상태 천이는 이웃 노드의 동적인 토폴로지 변화에 따라 이루어진다. 즉, non-coordinator 노드가 가진 두 이웃 노드가 직접 연결, 1개의 coordinator를 경유한 연결, 또는 2개의 coordinator를 경유한 연결 중 어느 것에도 포함되지 않을 경우 해당 노드는 coordinator 상태로 변경된다. Non-coordinator 상태에서 coordinator 상태로 변경되기 전에 에너지 레벨, 노드 degree 등의 정보에 기반한 우선 순위에 따라 백-오프 지연 시간을 가진다. 즉, 백-오프 지연 시간이 짧을수록 상대적으로 coordinator가 될 가능성이 커진다. Coordinator가 위의 조건을 만족하지 않을 경우에는 다시 non-coordinator로 변경된다. Span에서 각 노드는 HELLO 메시지를 통해 coordinator 여부, 자신의 현재 coordinator, 이웃 노드 리스트 등의 정보를 전달한다.

7) Distributed Dominant Pruning

Distributed Dominant Pruning[10]에서는 CDS에 기반한 플러딩을 위해 게이트웨이 설정(making) 및 해제(pruning) 규칙을 사용하고 있다. 게이트웨이 설정 과정에서, 직접 연결되지 않은 두 이웃 노드를 가진 노드는 게이트웨이 노드로 설정한

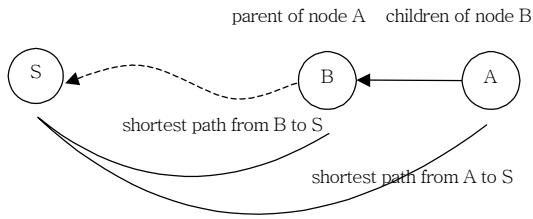
다. 이와 같은 절차를 수행하여 얻어진 게이트웨이 노드의 집합은 CDS를 구성한다. 계산된 CDS의 크기를 감소시키기 위해 두 가지의 게이트웨이 해제 규칙을 제안하고 있다. 각 노드는 유일한 식별자(ID)를 가지며 ID 간에는 순서(우선 순위)가 존재한다. 규칙-1의 경우, 두 인접한 게이트웨이 A와 B에 대해 $N[A] \leq N[B]$ 이고 $ID(A) < ID(B)$ 일 경우 A를 게이트웨이 노드 집합에서 제거한다. $N[A] = N(A) \cup \{A\}$ 로 정의된다. 규칙-2의 경우, 게이트웨이 노드 A의 이웃 게이트웨이 B, C에 대해 $N(A) \leq N(B) \cup N(C)$ 이고 $ID(A) = \min\{ID(A), ID(B), ID(C)\}$ 일 경우 A를 게이트웨이 노드 집합에서 제거한다. (그림 9a)는 게이트웨이 설정에 따라 노드 A와 노드 B가 게이트웨이로 설정된 것을 보여 주고 있다. (그림 9b)는 게이트웨이 해제 규칙-1에 의해 노드 A가 게이트웨이에서 해제되는 것을 보여 주고 있다. 이와 같은 게이트웨이 해제 규칙은 이웃한 게이트웨이의 수를 k로 설정함으로써 규칙-k(k-dominant pruning)로 확대 가능하다[11].



(그림 9) 게이트웨이 설정 및 해제

나. 소스 트리 기반 방식

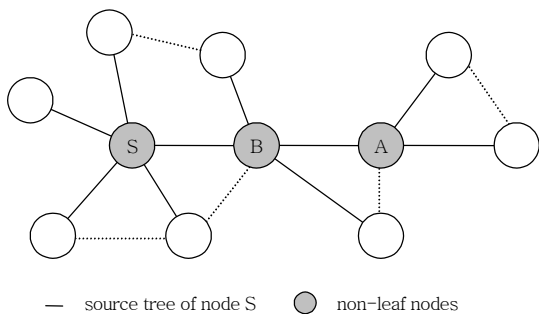
TBRPF(Topology Broadcast Based on Reverse-Path Forwarding) 라우팅 프로토콜에서는 소스-트리에 기반한 플러딩 방식을 사용하여 네트



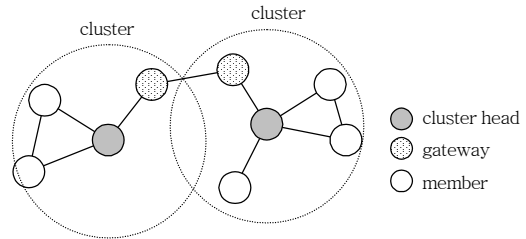
(그림 10) Parent 노드와 Children 노드간의 관계

워크 토폴로지 정보를 브로드캐스팅 한다[12]. 각 노드는 네트워크 내의 다른 모든 노드로의 최소-홉 수 경로로 구성되는 소스 트리를 유지한다. (그림 10)에서 노드 A는 이웃 노드 B를 경유하여 목적(destination) 노드 S로 가는 최소-홉 수의 경로를 가진다. 이 때 노드 B는 노드 S에 대해 노드 A의 parent 노드가 되며, 반대로 노드 A는 노드 B의 children 노드가 된다. 노드 S에 대한 모든 parent 노드는 노드 S에 대한 최소-홉 spanning 트리(소스 트리)를 구성한다.

브로드캐스트 패킷의 플러딩은 소스 트리에서 non-leaf 노드를 통해서만 이루어진다. (그림 11)에서와 같이, 노드 S가 생성한 브로드캐스트 패킷을 노드 A가 자신의 이웃 노드인 B로부터 수신한 경우, 노드 S에 대해 노드 B가 노드 A의 parent 노드인 경우에만 해당 브로드캐스트 패킷을 다시 플러딩한다. TBRPF에서는 HELLO 메시지를 사용하여 자신의 1-홉 이웃 노드에 대한 정보를 획득하고, 이웃 노드로부터 수신된 토폴로지 정보를 이용하여 자신의 소스 트리를 구성한다. 자신의 소스 트리 상에 변



(그림 11) 소스 트리에 기반한 플러딩



(그림 12) 클러스터링에 기반한 이동 Ad Hoc 네트워크

경된 토폴로지 정보를 자신의 소스 트리를 이용하여 네트워크 전체로 플러딩한다.

다. 클러스터 기반 방식

클러스터는 클러스터 헤드라는 하나의 노드와 이 노드로부터 직접 통신 거리, 즉 1-홉 무선 거리에 존재하는 모든 노드의 집합으로서 정의된다. 이동 Ad Hoc 네트워크를 위한 다양한 클러스터링 기법들이 연구되고 있으며 경우에 따라 클러스터가 서로 중첩되거나, 그 어떤 클러스터에도 속하지 않는 노드(non-cluster node)가 존재할 수도 있다. 클러스터 노드 중 게이트웨이는 다른 클러스터의 헤드와 1-홉 거리에 있는 노드, 즉 둘 이상의 클러스터에 속하는 노드로 정의된다. (그림 12)와 같이 클러스터가 서로 중첩되지 않게 구성될 경우, 게이트웨이는 인접 클러스터의 일반 클러스터 노드와 직접 통신하는 노드로 정의된다.

클러스터를 이용한 플러딩을 위해서는 먼저 클러스터를 구성하는 클러스터링이 수행되어야 한다. 클러스터링 방식의 예로 Linked-Cluster Algorithm(LCA)과 Highest-Connectivity Algorithm(HCA)을 들 수 있다. 노드 ID 간에 순서(order)가 있다고 할 때 LCA에서는 특정 노드가 자신의 모든 이웃 노드보다 작은 값의 ID를 가지고 있을 경우 클러스터 헤드가 된다. HCA에서는 특정 노드가 자신의 모든 이웃한 노드보다 큰 값의 degree(이웃 노드의 개수)를 가질 때 클러스터 헤드가 된다. 인접한 두 노드가 동일한 degree를 가질 경우 노드 ID 값에 의해 클러스터 헤드를 결정한다.

클러스터링에 의해 클러스터가 구성되었을 경우



(그림 13) 클러스터링 방식에서의 포워딩 노드 선택

Gateway Forwarding(GWF) 방식에서는 (그림 13a)와 같이 클러스터 헤드, 게이트웨이 및 클러스터에 속하지 않는 노드만이 포워딩 노드가 되어 브로드캐스트 패킷을 플러딩한다. 그러나, 게이트웨이 노드들간의 플러딩은 서로 중복될 가능성이 있기 때문에 클러스터 헤드에 의해 선택된 일부 게이트웨이만이 플러딩을 수행할 수 있는 Selected Gateway Forwarding(SGF) 방식을 사용함으로써 플러딩 성능을 향상시킬 수 있다. 즉, (그림 13b)에서와 같이 클러스터 헤드는 2-홉 또는 3-홉에 위치한 이웃 클러스터 헤드 또는 클러스터에 속하지 않는 노드와의 연결성을 가진 최소의 게이트웨이 노드 집합을 구하고 여기에 속하는 게이트웨이를 대표(representative) 게이트웨이로 설정한다. 게이트웨이 중에서 대표 게이트웨이만이 포워딩 노드가 된다[13].

IV. 맺음말

본 고에서는 이동 Ad Hoc 네트워크에서의 플러딩 수행 시의 문제점 및 이를 해결하기 위한 주요 플러딩 기법들을 기술하였다. 이동 Ad Hoc 네트워크에서는 노드들이 공유된 무선 채널을 사용하고 있고, 노드들간의 무선 전송 영역이 서로 겹치는 경우가 많기 때문에 플러딩 수행시 동일 브로드캐스트 패킷의 중복된 수신, 무선 자원에 대한 경쟁 및 충돌 현상이 빈번히 발생하는 문제점이 있다.

이동 Ad Hoc 네트워크는 토폴로지가 동적으로 변하기 때문에 전체 네트워크 토폴로지 정보에 기반한 최적의 플러딩을 수행하기는 매우 어렵다. 그러나, 플러딩은 이동 Ad Hoc 네트워크의 성능을 결정짓는 라우팅 프로토콜 등에서 필수적으로 사용되고 있기 때문에 보다 효율적인 기법들에 대한 연구가 지속되리라고 본다.

참고 문헌

- [1] Y-C Tseng, S-Y Ni, Y-S Chen, and J-P Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," *ACM/Kluwer Wireless Networks*, No. 8, 2002, pp. 153-167.
- [2] Y-C Tseng, S-Y Ni, and E-Y Shih, "Adaptive Approaches to Relieving Broadcast Storms in a Wireless Multihop Mobile Ad Hoc Network," *IEEE Transactions on Computer*, Vol. 52, No. 5, May 2003, pp. 545-557.
- [3] H. Lim and C. Kim, "Flooding in Wireless Ad Hoc Networks," *Computer Communications Journal*, Vol. 24, No. 3, Feb. 2001, pp. 353-363.
- [4] W. Lou and J. Wu, "On Reducing Broadcast Redundancy in Ad Hoc Wireless Networks," *IEEE Trans. on Mobile Computing*, Vol. 1, No. 2, Apr. 2002, pp. 111-123.
- [5] Y.J. Yi, M. Gerla, and T.J. Kwon, "Efficient Flooding in Ad hoc Networks: a Comparative Performance Study," *Proc. of IEEE ICC*, Vol. 2, No. 11-15, May

- 2003, pp. 1059-1063.
- [6] B. Williams and T. Camp, "Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks," *Proc. of the 3rd ACM International Symposium on Mobile Ad hoc networking & computing(MOBIHOC)*, 2002, pp. 194-205.
- [7] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint Relaying for Flooding Broadcast Message in Mobile Wireless Networks," *Proc. of the 35th Hawaii International Conference on System Sciences*, Jan. 2002.
- [8] W. Peng and X. Lu, "On the Reduction of Broadcast Redundancy in Mobile Ad Hoc Networks," *Proc. of MOBIHOC '00*, 2000, pp. 129-130.
- [9] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An Energy-efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," *ACM/Kluwer Wireless Networks*, Vol. 8, No. 5, Sep. 2002, pp. 481-494.
- [10] J. Wu and H. Li, "On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks," *Proc. of the 3rd Int'l Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, 1999, pp. 7-14.
- [11] F. Dai and J. Wu, "Distributed Dominant Pruning in Ad Hoc Networks," *Proc. of ICC*, May 2003, pp. 353-357.
- [12] B. Bellur and R.G. Ogier, "A Reliable, Efficient Topology Broadcast Protocol for Dynamic Networks," *Proc. of IEEE INFOCOM '99*, Vol. 1, Mar. 1999, pp. 178-186.
- [13] K. Mase, Y. Wada, N. Mori, K. Nakano, and M. Sengoku, "Flooding Schemes for Clustered Ad Hoc Networks," *IEICE Trans. on Communications*, Vol. E85-B, No. 3, Mar. 2002, pp. 605-613.