

이동 애드 혹 망에서 안정된 응용 계층 멀티캐스트

우부재^o, 최영환, 박수창, 이의신, 전야, 박호성, 김상하

충남대학교 컴퓨터공학과

{yufc, yhchoi, winter, eslee, tianye, hspark}@cclab.cnu.ac.kr and shkim@cnu.ac.kr

Stable Application Layer Multicast for Mobile Ad hoc Networks

Fucaí Yu^o, Younghwan Choi, Soochang Park, Euisin Lee, Ye Tian, Hosung Park and Sang-Ha Kim

Dept. of Computer Engineering, Chungnam National University

요 약

이동 애드 혹 망에서 안정된 멀티캐스트 데이터 전달 구조는 데이터 전송률을 높이고 제어 부하를 낮추는데 기여한다. 이동 애드 혹 망의 동적 위상 때문에 애드 혹 환경에서 안정된 멀티캐스트 데이터 전달 구조를 구성하는 것은 어려운 문제이다. 이 논문에서 우리는 이동 애드 혹 망을 위한 안정된 응용 계층 멀티캐스트를 제안한다. 이 프로토콜에서는 각각의 멤버 노드가 잔여 에너지와 지역 이동성에 따라 자신의 안정성을 미리 계산한다. 소스가 데이터 패킷을 전송해야 할 때, 변형된 지연 전달 방안을 통해 소스 기반 오버레이 멀티캐스트 트리를 만든다. 오버레이 구성은 각각의 멤버 노드의 안정성에 따른다. 상대적으로 높은 안정성을 가진 멤버 노드는 하위 트리들을 가질 확률을 높여준다. 제안한 설계는 불안정한 멤버 노드들을 멀티캐스트 트리의 가장자리로 밀어내어 멀티캐스트 라우팅 안정성을 개선한다. 시뮬레이션 결과는 우리의 설계가 패킷 전송률, 링크 단절에 의한 손실률, 제어 부하 면에서 ODOMP[1]에 비해 우수하다는 것을 보여준다.

1. 서 론

이동 애드 혹 망을 위한 많은 멀티캐스트 프로토콜이 제안되어 왔다. 이들을 두 가지 범주로 분류할 수 있다: 프로액티브(proactive)와 리액티브(Reactive). 프로액티브 프로토콜들은 데이터 전달 구조가 데이터 소스와 상관없이 유지, 갱신된다. 소스가 데이터 패킷을 전송하기 시작할 때, 데이터 패킷은 미리 정해진 데이터 전달 구조를 통해 목적지까지 전달된다. 그러나 프로액티브 프로토콜들은 소스로부터 멀티캐스트 데이터 전송이 없더라도 데이터 전달 구조 유지를 위해 주기적인 제어 패킷의 교환이 필요하다. 프로액티브 프로토콜들은 많은 망 자원(대역폭과 전력 등)을 낭비하고 대량의 제어 부하를 일으킨다. 프로액티브 프로토콜과 대조적으로, 리액티브 프로토콜들[2]-[4]은 소스가 전송할 데이터를 가진 경우에만 데이터 전달 구조를 갱신, 유지하여 이런 결정을 피한다. 이렇게 함으로써 제어 메시지 교환을 계속해서 주기적으로 하지 않는다. 따라서 망 자원을 절약하고 제어 부하를 감소시킬 수 있다.

이동 애드 혹 망을 위한 오버레이 멀티캐스트 프로토콜[5][6] 또한 많이 제안되었다. 이런 오버레이 멀티캐스트 프로토콜들은 멤버들로 이루어진 그룹을 기반으로 어떠한 유니캐스트 프로토콜도 사용할 수 있는 그룹 위상을 목표로 한다. 이런 방안은 높은 동적 환경에서도 오버레이 멀티캐스트를 더 안정되도록, 더 낮은 제어 부하를 가지도록 해준다.

ODOMP[1]는 최근에 제안된 리액티브 오버레이 멀티캐스트 프로토콜이다. ODOMP에서 오버레이 트리는 소스가 데이터 패킷을 보내야 할 경우에만 구성된다. 이런 리액티브 방법은 오버레이 유지 부하를 상당히 줄인다. ODOMP는 소스가 망에 JREQ 메시지를 브로드캐스트로

전송하는 방법으로 오버레이 구성(또는 재구성)을 시작한다. 한번 오버레이 멀티캐스트 트리가 만들어지면, ODOMP는 주기적인 오버레이 재구성을 통해 오버레이를 갱신한다. 그러나 너무 빈번한 주기적 오버레이 재구성은 높은 망 부하를 낳게 되고, 이것이 ODOMP의 주요 결점이 된다. ODOMP는 오버레이를 최적화하기 위해 빈번한 오버레이 재구성을 필요로 한다. 주요 원인은 다음과 같다: 첫째, ODOMP는 가장 짧은 경로로 오버레이 트리를 구성한다. 오버레이의 안정성은 고려하지 않는다. 망의 이동성 때문에 한번 구성된 오버레이 트리는 오랫동안 최적화된 상태를 유지하지 못한다. 둘째, 새로 합류한 그룹 멤버는 소스에 직접 연결되기 때문에 소스의 하위 멤버가 된다. 또한 하나의 멤버 노드가 작동하지 않는다면, 그 노드의 모든 하위 멤버들이 오버레이에 재합류하기 위해 소스에 직접 연결된다. 이런 새로운 오버레이 연결이 최적화가 아니란 것은 명백하다. 이런 이유들이 빈번한 오버레이 재구성을 필요하게 만들고, 네트워크 부하를 크게 증가시킨다.

이 논문에서 우리는 안정성 기반 오버레이 멀티캐스트 프로토콜(Stability-Oriented Overlay Multicast Protocol: SOOMP)이라는 이동 애드 혹 망을 위한 리액티브 멀티캐스트 프로토콜을 제안한다. SOOMP의 목표는 애드 혹 망에서 멀티캐스트 통신을 위한 안정된 오버레이 트리를 구성하는 것이다. SOOMP에서는 각각의 멤버 노드가 잔여 에너지와 지역 이동성에 따라 자신의 안정성을 미리 계산한다. 소스 노드는 전송할 데이터 패킷이 있는 경우에만 오버레이 멀티캐스트 트리 구성을 시작한다. SOOMP는 오버레이를 만들기 위해 변형된 지연 전달 방안을 이용한다. SOOMP의 지연 시간은 상위 멤버 노드의 안정성 기반으로 한다. 다시 말해, 더 안정된 멤버 노드가 그렇지 않은 멤버 노드에 비해 하위 트리를 차지할

확률이 높다.

이후 논문은 다음과 같이 구성된다: 2단원에서는 멤버 노드의 안정성을 계산하는 방안을 기술한다. 3단원에서는 SOOMP의 오버레이 구성, 유지 그리고 데이터 전달 과정에 대해 상세히 설명한다. 제안된 방안의 성능 분석은 4단원에서 이어진다. 마지막으로, 5단원에서는 결론을 내리고, 향후 연구 방향을 언급한다.

2. 멤버 노드의 안정성

트리 기반 멀티캐스트 프로토콜의 가장 큰 결점 중 하나는 안정성이 떨어진다는 것이다. 트리의 한 부분이 작동하지 않거나 그룹에서 벗어나면 그 부분의 모든 하위 트리는 다시 트리에 연결될 때까지 데이터 패킷을 수신할 수 없다. 이것은 멀티캐스트 프로토콜의 성능에 영향을 미친다. 잔여 에너지가 많은 노드는 더 긴 수명을 가지고, 지역 이동성이 떨어지는 노드는 링크를 단절할 가능성이 적다. 그래서 잔여 에너지가 많을수록, 지역 이동성이 낮을수록 확실히 안정하다. SOOMP 오버레이 멀티캐스트 트리 구성의 주요 원리는 안정된 멤버 노드가 서브트리를 가질 기회가 많아야 한다는 것이다. 이런 방법으로 우리는 멀티캐스트 트리의 안정성을 강화할 수 있다.

SOOMP 오버레이 트리 구성은 각각의 멤버 노드의 안정성(Member Node Stability: MNS)을 기반으로 한다. 멤버 노드의 MNS는 자신의 지역 이동성과 잔여 에너지에 따라 계산된다. 우리는 MNS를 다음과 같이 정의한다:

$$MNS = \%Battery / M_{local} , \quad (1)$$

M_{local} 은 멤버 노드의 지역 이동성이며, $\%Battery$ 는 멤버 노드의 잔여 에너지이다. M_{local} 은 MOBIC[7]에서 제안된 방식에 의해 계산할 수 있다.

3. SOOMP 오버레이 구성과 유지

3.1 SOOMP 오버레이 구성

앞서 언급했듯이 SOOMP는 소스 기반의 안정된 오버레이 멀티캐스트 트리를 목표로 한다. 이를 위해 오버레이 구성에 변형된 지연 전달 방안을 사용한다. ODOMP에서 설명한 지연 전달 방안과 비교했을 때, 차이점은 지연 시간이다. 지연 시간이란 JREQ 패킷이 비멤버 노드들을 거치면서 지연될 시간을 말한다. ODOMP에서는 지연 시간을 JREQ 패킷을 보낸 멤버 노드로부터의 홑거리로 결정한다. 반면 SOOMP에서는 지연 시간을 멤버 노드의 안정성에 기초하여 결정한다. 우리는 그림 1에서 SOOMP의 오버레이 구성에 대해 상세히 설명할 필요가 있는 부분만을 뽑아내어 묘사했으며, 무선 장치의 모든 전송 지연은 무시했다.

우선, 소스 노드 A가 멀티캐스트 데이터 패킷을 보내려 할 경우, 이 패킷에 유효한 오버레이 트리가 없다면

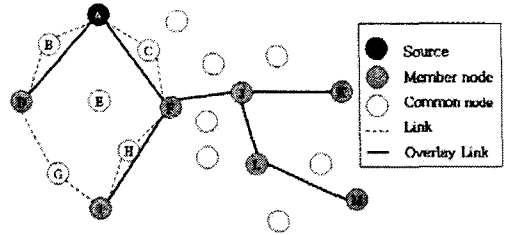


그림 1. SOOMP 오버레이 구성

패킷을 버퍼에 저장하고 Join REQuest (JREQ) 패킷을 브로드캐스트로 전송하여 오버레이 구성을 시작한다. JREQ는 SrcAddr, GrpAddr, SeqNumber, LastMember, T_DELAY 항목을 포함한다. SrcAddr은 소스의 주소, GrpAddr은 멀티캐스트 그룹 주소로 설정한다. JREQ 패킷은 소스의 주소와 순서 번호를 이용해서 유일하게 구분할 수 있다. LastNumber 항목은 JREQ를 마지막으로 전달할 멤버 노드의 주소로 설정한다. 최초에, 소스 노드는 LastMember 항목을 자신의 주소로 설정한다. T_DELAY는 비멤버 노드에게 이 JREQ 패킷을 얼마나 오랫동안 지연시켜야 하는지 알려준다. T_DELAY는 다음과 같이 계산한다.

$$T_DELAY = T_{unit} / MNS = T_{unit} * (M_{local} / Battery), \quad (2)$$

T_{unit} 은 작은 상수 시간이다. 높은 안정성을 가진 멤버 노드는 작은 T_DELAY 값을 산출한다. 소스 노드는 자신의 안정성에 따라 T_DELAY를 계산하고, JREQ 패킷의 T_DELAY 항목에 설정한다.

JREQ를 수신한 노드는 우선 중복 패킷이 아닌지 점검한다. 중복 패킷이라면 무시한다. 그림 1의 노드 B와 C 처럼 비멤버 노드라면 JREQ를 수신한 때로부터 T_DELAY 만큼의 시간을 지연하고, 이웃 노드들에게 다시 JREQ를 브로드캐스트로 전송한다. 이 때, T_DELAY는 받은 JREQ에 들어있는 값을 사용한다. 노드 B와 C는 모두 소스 A에 의해 계산되고 JREQ 패킷에 저장된 T_Dealy 만큼의 시간을 기다리는 것이다. 이것이 우리의 변형된 지연 전달 방안이다.

그림 2에서 노드 D와 F 같은 멤버 노드들은 JREQ 패킷을 수신하면, 그 안의 LastMember 항목을 이 그룹에서 자신의 상위 멤버 노드로 저장한다. 그 후 JREQ의 LastMember 항목을 자신의 주소로 설정하고, T_Dealy 항목을 자신의 안정성에 따라 (2)로 계산한 새로운 값으로 설정한다. 다음으로 JREQ를 이웃 노드들에게 브로드캐스트로 전송한다. 위의 예에서 멤버 노드 D와 F는 각각 동시에 비멤버 노드 B와 C로부터 JREQ를 수신하고, 둘 모두 상위 멤버 노드로 소스 A를 저장한다. 그리고 D와 F는 각각의 안정성에 따라 계산된 값을 JREQ의 T_DELAY 항목에 설정한다. 그리고 나서 D와 F 모두 그들의 이웃 노드들에게 수정된 JREQ를 브로드캐스트로 재전송한다. 여기서 우리는 멤버 노드 D의 안정성이 멤버 노드 F보다 떨어진다고 가정했고($MNS_D < MNS_F$), 따

라서 노드 D가 (2)로 계산한 지연 시간이 노드 F보다 크다($T_DELAY_D > T_DELAY_F$). 따라서 비멤버 노드 G와 H가 동시에 각각 D와 F로부터 JREQ를 수신했다고 해도, F로부터 JREQ를 수신한 H가 D로부터 JREQ를 수신한 G보다 더 빨리 JREQ를 전송한다. 그러므로 멤버 노드 I는 G보다 H로부터 먼저 JREQ를 수신하게 되고, F를 상위 멤버 노드로 저장한다. 더 늦게 도착한 G의 JREQ는 중복 패킷으로 인식되어 무시한다. 여기서 우리는 멤버 노드 I가 D와 F까지 같은 홉 수를 가진다고 하더라도 멤버 노드 F가 더 높은 안정성을 가졌기 때문에 더 짧은 T_DELAY 를 만들었고, 따라서 I가 F로 연결된 것을 알 수 있다. 이 과정은 모든 멤버 노드들이 오버레이에 참여할 때까지 이어진다.

소스 노드를 제외한 각각의 멤버 노드들은 정확하게 하나의 상위 멤버 노드를 갖는다. JREQ를 수신한 멤버 노드는 JOIN REPLY(JREP) 패킷을 만들어 상위 멤버 노드에게 유니캐스트로 전송한다. 상위 멤버 노드는 JREP를 전송한 노드를 자신의 하위 멤버 노드들 중에 하나로 저장한다. 이 과정에서 소스 기반 오버레이 트리가 생성된다.

4. 성능 분석

SOOMP의 성능 분석은 망 시뮬레이터 QualNet 3.8로 실행했다. SOOMP의 성능은 ODOMP의 성능과 비교했다. ODOMP는 지금까지 제안된 멀티캐스트 연구들 중에 가장 우리의 연구와 비슷하다.

4.1 시뮬레이션 환경

시뮬레이션은 $1500 \times 1500m^2$ 의 공간에서 100개의 이동 호스트들이 무작위로 위치하도록 설계했다. 각각의 노드는 전송 범위가 250m이며, 채널 용량이 2Mbit/s이다. 이동 모델은 정지 시간이 변화하는 임의 중간지점 모델을 사용한다. 무선 전송 모델은 송수신 경용 기반 모델을 사용했다. 데이터는 상수 비트율로 발생한다(1 bits/second). 마지막으로, MAC 계층 프로토콜로 IEEE 802.11을 사용했다. 속도 범위는 0m/s에서 2m/s로 설정했으며, 시뮬레이션 시간은 600s이다.

4.2 측정 요소

4.2.1 링크 단절에 의한 손실률: 이 비율은 멤버 노드의 전체 개수에 대한 링크 단절에 의해 영향을 받은 멤버 노드들의 평균 개수 비율로 정의한다.

$$R = \sum_0^m (A_m / N - 1) / M, \quad (3)$$

A_m 은 하나의 연결이 단절되었을 경우 영향을 받는 멤버 노드들의 개수이다. 즉, 단절된 연결의 하위 트리에 위치하는 멤버 노드들의 개수이다. N 은 그룹의 크기이다. m 은 시뮬레이션 시간 도중에 일어나는 연결 단절의

개수이다. 알다시피, 단절된 연결의 하위 트리의 모든 멤버 노드들은 일시적으로 데이터 패킷을 수신할 수 없다. 이 문제는 단절된 링크의 하위 트리에 위치한 멤버 노드들의 개수가 많을수록 더욱 심각해진다. SOOMP는 안정된 오버레이 트리를 구성하는 것에 초점을 둔다. 이 목표를 달성하기 위해 안정된 멤버 노드에게 하위 트리를 가질 확률을 높여준다. 다시 말해서 SOOMP는 불안정한 멤버 노드들이 트리의 가장자리에 위치하도록 시도한다. 트리의 가장자리와 불안정한 멤버 노드들 사이의 링크 단절은 프로토콜의 효율성에 미치는 영향이 적다. 그러므로 이 요소를 통해 우리는 프로토콜의 안정성을 측정할 수 있다.

4.2.2 패킷 전송률: 패킷 전송률은 소스가 전송한 데이터 패킷들의 수에 대한 모든 멤버들이 받은 데이터 패킷의 평균 개수 비율로 정의한다.

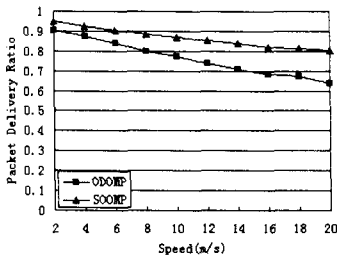
$$R = \sum R_m / N - 1 / T_s, \quad (4)$$

N 은 그룹의 크기이고, $\sum R_m$ 은 모든 멤버들이 수신한 데이터 패킷의 전체 개수이며, T_s 는 소스가 전송한 패킷의 개수이다. 이 요소는 멀티캐스트 프로토콜의 효율성을 반영한다.

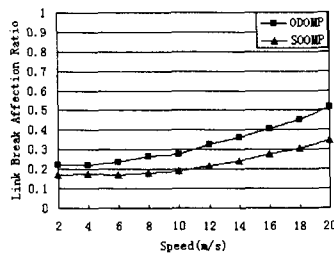
4.2.3 제어 부하: 제어 부하는 송신된 멀티캐스트 데이터 패킷의 개수에 대한 전송된 제어 패킷의 개수 비율로 정의한다.

4.3 시뮬레이션 결과

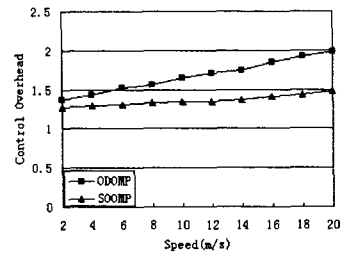
4.3.1 이동성: 우리는 이동성이 SOOMP와 ODOMP의 성능에 미치는 영향을 측정하기 위해 최고 속도를 2m/s에서 20m/s까지 변화시켰다. 시뮬레이션 결과는 그림 2에 나타났다. 그림 2-1은 속도가 증가함에 따라 두 프로토콜의 패킷 전송률이 조금씩 감소함을 보여준다. 이것은 두 프로토콜 모두에서 패킷 손실을 야기하는 링크 단절의 가능성이 증가했기 때문이다. ODOMP와 비교했을 때, SOOMP는 안정된 멤버 노드들이 하위 트리를 가질 확률을 높여주는 방법으로 더 안정된 오버레이 트리를 구성한다. 이는 불안정한 멤버 노드들이 일반적으로 트리의 가장자리에 위치한다는 것과 상대적으로 하위 트리에 멤버 노드들을 적게 가진다는 것을 의미한다. 따라서 SOOMP에서 이런 불안정한 멤버들의 링크 단절에 의해 발생하는 패킷 손실의 개수가 ODOMP에서의 그것보다 더 적다. 그림 2-2는 속도가 증가함에 따라 두 프로토콜의 링크 단절에 의한 손실률이 조금씩 증가함을 보여준다. 이것 역시 링크 단절의 가능성이 증가하기 때문이다. 그러나 SOOMP에서는 불안정한 멤버 노드들이 일반적으로 트리의 가장자리에 위치하고, 이것은 이런 불안정한 멤버 노드들이 상대적으로 하위 멤버들을 적게 가진다는 것을 의미한다. 따라서 불안정한 멤버 노드들의 링크 단절이 망의 전송 실패에 주는 영향이 ODOMP에 비해 더 적다. 제어 부하는 라우팅 경로 검색 부하와 라우팅 경



(1) 패킷 전송률

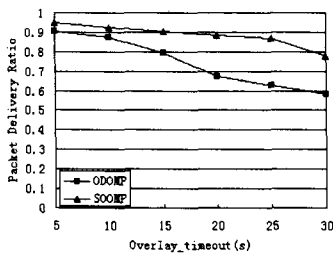


(2) 링크 단절에 의한 손실

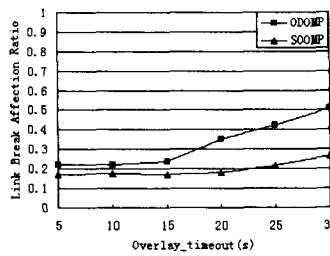


(3) 제어 부하

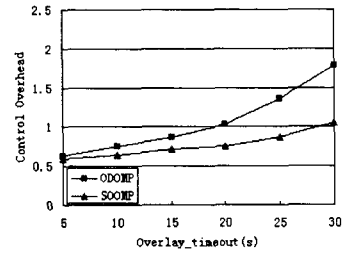
그림 2. 이동성이 SOOMP와 ODOMP의 성능에 미치는 영향



(1) 패킷 전송률



(2) 링크 단절에 의한 손실



(3) 제어 부하

그림 3. OVERLAY_TIMEOUT이 SOOMP와 ODOMP의 성능에 미치는 영향

로 유지 부하로 구성된다. SOOMP와 ODOMP는 둘 다 라우팅 경로 검색을 위해 주기적인 플래딩 메시지를 사용하기 때문에, OVERLAY_TIMEOUT이 같다면 같은 라우팅 경로 검색 부하를 가진다. 라우팅 경로 유지 부하는 속도가 증가함에 따라 두 프로토콜 모두 조금씩 증가한다. 이것 역시 링크 단절의 가능성이 증가하기 때문이다. ODOMP는 링크 단절을 다루기 위해 AODV와 같은 유니캐스트 프로토콜을 사용하는 반면, SOOMP는 링크 단절을 다루기 위해 ERS를 사용한다. 알다시피 ERS는 특히 짧은 거리의 라우팅 경로 검색을 위한 효과적이고 범위 한정적인 라우팅 경로 검색 방안이다. 그림 2-3은 이동성 요소에 따른 두 프로토콜의 제어 부하 비교를 보여준다.

이 증가함에 따라 링크 단절의 횟수 또한 증가한다. 반면 SOOMP는 일반적으로 링크 단절이 트리의 가장자리에서 발생하므로 OVERLAY_TIMEOUT이 패킷 전송률에 미치는 영향이 적다. 그림 4-2는 OVERLAY_TIMEOUT이 두 프로토콜의 링크 단절에 의한 손실률에 미치는 영향을 보여준다. 앞서 언급한 것과 같은 이유로 인해 OVERLAY_TIMEOUT이 증가함에 따라 ODOMP의 링크 단절에 의한 손실률이 SOOMP보다 급격하게 증가한다. 따라서 ODOMP는 링크의 재구성을 위해 더 많은 제어 부하가 필요하다. 그림 4-3은 제어 부하에 관한 시뮬레이션 결과를 보여준다.

5. 결 론

4.3.2 OVERLAY_TIMEOUT: 우리는 OVERLAY_TIMEOUT이 SOOMP와 ODOMP의 성능에 미치는 영향을 측정하기 위해 OVERLAY_TIMEOUT을 5초에서 30초까지 변화시켰다. 시뮬레이션 결과를 그림 3에 나타냈다. OVERLAY_TIMEOUT은 ODOMP와 SOOMP의 성능에 두드러지게 큰 영향을 미쳤다. 그림 4-1에서 보여주듯이 OVERLAY_TIMEOUT을 15초 이상으로 설정하면 ODOMP의 패킷 전송률은 심각하게 감소한다. ODOMP는 데이터 전달을 위한 안정된 오버레이의 구성에 초점을 두지 않기 때문이다. 따라서 생성된 오버레이 트리는 오랜 기간 최적화된 상태로 지속되지 못하고, OVERLAY_TIMEOUT

이 논문에서 우리는 안정성 기반 오버레이 멀티캐스트라는 이동 애드 혹 망을 위한 리액티브 멀티캐스트 프로토콜을 제안했다. SOOMP의 목표는 애드 혹 망에서 멀티캐스트를 위한 안정된 오버레이 트리를 구성하는 것이다. SOOMP에서 각각의 멤버 노드는 지역 이동성과 잔여 에너지를 통해 자신의 안정성을 미리 계산한다. ODOMP와 비슷하게, SOOMP는 오버레이를 구성하기 위해 지연 전달 방안을 사용한다. 하지만 상위 멤버 노드까지의 홉 수를 기반으로 지연 시간을 설정하는 ODOMP와 달리, SOOMP는 상위 멤버 노드의 안정성을 기반으로 지연 시간을 설정한다. 다시 말해, SOOMP에서는 더

안정된 멤버 노드가 불안정한 멤버 노드에 비해 하위 트리를 가질 가능성이 높으며, 이런 방법으로 멀티캐스트 라우팅 안정성을 향상시킨다. 시뮬레이션 결과는 SOOMP가 ODOMP에 비해 높은 성능을 가짐을 보여준다.

참고 문헌

- [1] O. Stanze and M. Zitterbart. On-Demand Overlay Multicast in MobileAd hoc Networks. In Proceedings of IEEE WCNC, Vol. 4, Mar. 2005, pp. 2155 - 2161.
- [2] S. Basagni, I Chlamtac, V.R. Syrotiuk and R. Talebi. On-demand location aware multicast (OLAM) for ad hoc networks. In Proceedings of IEEE WCNC, Vol.3, Sep. 2000 pp.1323 - 1328.
- [3] C.W.Wu and Y. C. Tay. AMRIS: Amulticast protocol for ad hoc wireless networks. In Proceeding of IEEE MILCOM, Atlantic City, NJ, Nov. 1999, pp. 25-29.
- [4] E. M. Royer and C. E. Perkins. Multicast ad hoc on-demand distance vector (MAODV) routing. IETF Internet Draft, Jul. 2000.
- [5] J. Xie, R. Taipade, A. Mcauley, and M. Liu. AMRoute: Ad hoc Multicast Routing Protocol. In Proceedings of Mobile Networks and Applications, vol. 7, Dec. 2002, pp.429-439.
- [6] K. Chen and K. Nahrstedt. Effective Location-Guided Tree Construction Algorithms for Small Group Multicast in MANET. In Proceedings of IEEE INFOCOM, Jun. 2002, pp. 1180-1189.
- [7] P. Basu, N. Khan and T.D.C.Little. A mobility based metric for clustering in mobile ad hoc networks. In proceedings of IEEE 2001, April 2001, pp.413 - 418.