제19회 한국정보처리학회 춘계학술발표대회 논문집 제10권 제1호 (2003.5)

Ad-Hoc 네트워크에서의 간단한 CBT 기반의 Overlay 멀티캐스트 프로토콜

양기선*, 이원준**
*삼성전자, **고려대학교 컴퓨터학과
e-mail:kisyang@samsung.com

A Simple CBT-Based Overlay Multicast Protocol for Ad-Hoc Networks

Ki-Seon Yang*, Won-Jun Lee**
*Samsung Electronics Co., LTD
**Dept. of Computer Science & Engineering, Korea University

요 약

멀티캐스트 그룹의 멤버들만으로 트리를 구성하는 Overlay 멀티캐스트 프로토콜인 AMRoute은 이동이 빈번한 Ad hoc 네트워크 환경에서 중간 라우터의 이동에 대한 트리의 견고함으로 높은 데이터 전송을 제공하는 장점이 있으나, 네트워크의 변동으로 인하여 중복되는 전송 경로가 발생하여 대역폭이 낭비되는 효율성 문제를 가지고 있다. 본 논문에서는 기존의 AMRoute 제어 메시지에 각 멤버의 메쉬 링크 상태 정보가 추가된 제어 메시지를 통하여 멤버 간에 링크 상태를 교환함으로써 AMRoute에서의 중복되는 전송 경로를 제거하여 대역폭의 낭비를 줄이는 방안을 제안한다. 본 논문에서 제안한 방안은 기존의 제어 메시지를 활용함으로써 적은 오버헤드로 새로운 메쉬 링크를 발견하여 기존의 AMRoute의 중복되는 전송 경로로 인한 대역폭의 낭비를 줄여 멀티캐스트의 성능을 향상시키는 효과를 기대할 수 있다.

1. 서론

대역폭과 전원 등의 자원이 제한된 Ad hoc 네트워크[1] 환경을 위한 많은 멀티캐스트 라우팅 프로토콜들이 제안되고 있으나 중간 라우터들의 잦은 이동으로 인하여 멀티캐스트 분배 트리의 견고성이 저하되는 문제를 안고 있다. Overlay 멀티캐스트 프로토콜인 AMRoute[2]는 Ad hoc 네트워크 환경에서의 견고한 데이터 전송을 제공하기 위하여 제안되었다. 그러나 AMRoute에서는 네트워크의 변동으로 전송 경로가 중복되는 비효율적인 경로가 설정되어 대역폭의 낭비가 발생하는 효율성 문제가 있다.

본 논문은 메쉬 링크의 상태 정보가 추가된 AMRoute의 제어 메시지를 멤버 간에 교환하여 새로운 메쉬 링크를 발견하여 트리를 재구성함으로써 AMRoute의 대역폭 낭비를 줄여 멀티캐스트 성능을 향상시키는 것을 목표로 한다.

2. 관련 연구

2.1 Ad Hoc 네트워크에서의 멀티캐스트

고정 네트워크에서 제안된 멀티캐스트 프로토콜들은 Ad hoc 네트워크에서는 빈번한 노드들의 이동으로 인하여 멀티캐스트 트리의 잦은 재구성으로 과도한 제어 트래픽과 잦은 패킷 분실을 유발시킬 수 있다. 이에 AODVM[3], ODMRP[4], LBM[5], ABAM[6] 등 Ad Hoc 네트워크 환경을 위한 많은 멀티캐스트 라우팅 프로토콜들이 제안되고 있다. 이러한 프로토콜들은 멀티캐스트 세션을 지원하기 위하여 멀티캐스트 그룹 멤버 노드들과 라우터 역할을 하는 멤버가 아닌 노드들까지 모두 포함하여 트리를 구성한다. 이에 따라 멀티캐스트 트리에 포함된 비멤버 노드들의 이동에 대해서도 트리를 재구성함으로써 멀티캐스트 트리의 견고성이 저하된다. 또한 멀티캐스트에 관심이 없는 노드들도 멀티캐스트 라우팅을 지원해야 하고, 멀티캐스트 트리와 관련된 정보를 관리, 유지해야하는 문제점이 있다.

2.2 Overlay 멀티캐스트

Overlay 멀티캐스트는 멀티캐스트 라우터들로 구

성된 네트워크에서 멀티캐스트를 수행하는 대신 그림 1과 같이 멀티캐스트 그룹의 멤버의 응용 계층에서 그룹 멤버들만으로 멀티캐스트 트리를 구성하여 멤버간의 유니캐스트 터널을 통하여 이웃 멤버로 패킷을 중계(복사한 후 전송)하는 유니캐스트 기반의 멀티캐스트이다. 비록 네트워크에서 멀티캐스트를 수행하는 것보다 지연 시간이나 패킷 수가 다소 증가하나, 중간 라우터들에게 멀티캐스트 라우팅을 요구하지 않는 장점이 있다. 특히 호스트와 중간 라우터의 이동이 빈번한 Ad hoc 네트워크 환경에서 Overlay 멀티캐스트는 다음과 같은 특징을 가진다.

첫째, 멤버가 아닌 중간 라우터 역할을 하는 노드들의 이동으로 인한 트리의 변동이 없다.

둘째, 하위 유니캐스트 라우팅 프로토콜에 의존적이지 않다. 즉, 중간 라우터의 이동에 의한 네트워크 변동은 하위 유니캐스트 라우팅 프로토콜에서 해결하므로 멀티캐스트 트리의 변동이 없어 트리의 견고성과 높은 데이터 전송을 제공하고, 중간 노드들이 멀티캐스트 관련 정보를 관리 및 유지하지 않아 자원을 절약하는 장점이 있다.

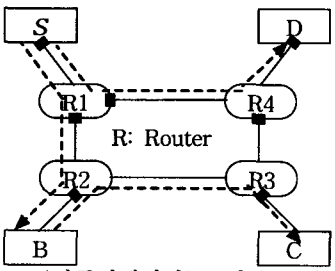


그림 1. Overlav 멀티캐스트

Ad hoc 네트워크에서의 Overlay 멀티캐스트 관련 연구로는 송신자 중심 트리 기반의 PAST-DM[7]과 공유 트리 기반의 AMRoute 등이 있다. 송신자 중심 트리는 공유 트리에 비하여 각 그룹 구성 정보를 흘려보내는 작업이 상당한 비용을 요구하여 오버헤드가 크고, 구조 알고리즘이 복잡한 단점이 있다.

2.2 AMRoute (Ad hoc Multicast Routing Protocol)

AMRoute는 그림 2와 같이 인접한 그룹 멤버들과 양방향 유니캐스트 터널들로 연결된 메쉬를 계속해서 형성하고, 이 메쉬를 기반으로 멀티캐스트 분배 트리를 주기적으로 형성한다. 오직 논리적 코어 노드들이 메쉬와 트리의 생성을 시작한다. 그러나 기존의 CBT[8] 프로토콜과는 달리, 논리적 코어 노드가 변할 수 있다.

1) 메쉬 생성

각 멤버는 JOIN 메시지를 통하여 그룹의 다른 멤버들을 발견하여 메쉬를 형성하며, 메쉬에서는 루프가 생길 수 있다. 그룹을 탈퇴할 경우는 JOIN_NAK 메시지를 메쉬 링크로 통하여 이웃 멤버들에게 송신한다.

2) 트리 생성

메쉬가 형성되면 데이터의 효율적인 전달을 위하여 루프가 존재하지 않는 트리가 필요하다. 코어에서 주기적으로 전송되는 TREE_CREATE 메시지를 수신한 멤버들은 메쉬 링크 중에서 효율적인 하나의 링크를 선택하여 ACK나 NAK 메시지로 응답하여 루프가 존재하지 않는 분배 트리를 생성한다.

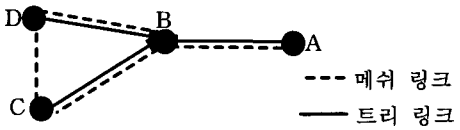


그림 2. Overlay 멀티캐스트 메쉬와 분배 트리

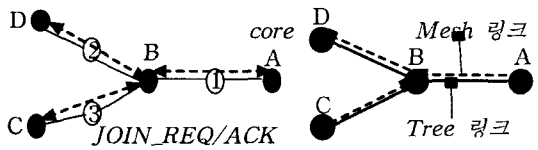
3) 트리 재구성

멤버가 탈퇴하거나 이상이 발생한 경우는 다른 메쉬 링크를 이용하여 해결할 수 있다. 만약 일정 시간동안 TREE_CREATE 메시지를 수신하지 못하면, 멤버는 자신을 코어 노드로 설정하고, JOIN_REQ를 전송하여 다른 멤버들을 찾는 초기 가입 과정을 수행한다.

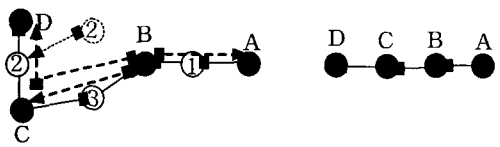
2.3 AMRoute의 효율성 문제(Efficiency Problem)

AMRoute는 견고한 트리 구성과 높은 데이터 전송의 장점을 가진다. 그러나 그룹 멤버들만으로 트리를 구성하기 때문에 트리 링크가 그림 3의 예와 같이 중복되는 경로가 발생하는 효율상의 문제점이 있다. 그림 3의 예에서 (a)의 네트워크 형상에 대하여 각 멤버들은 JOIN 메시지와 TREE_CREATE 메시지를 통하여 (b)와 같은 Overlay 트리를 구성하여 멀티캐스트 데이터를 각 멤버로 전송한다. 이 때 (a)에서 그룹 멤버가 아닌 중간 노드 2가 이동하여 (c)와 같은 네트워크를 형성하였을 경우, AMRoute에서는 멤버의 변동이 없으므로 (d)와 같은 Overlay 트리를 재구성하지 않고 기존의 (b)의 Overlay 트리를 이용하여 그룹 멤버들로 데이터를 전달한다. 이 결과, (c)의 노드 B와 노드 3, 노드 3과 노드 C의

경로가 중복되는 비효율적인 경로를 통하여 데이터를 전달하게 된다. 이와 같이 AMRoute는 견고한 데이터 전송을 제공하나 비 멤버 노드가 트리에 포함되지 않기 때문에 비효율적인 경로가 설정될 수 있으며 이에 따른 대역폭의 낭비와 패킷 지연을 유발시킬 수 있는 문제점이 있다.



(a) 네트워크 예 (b) 메쉬와 트리 구성



(c) 노드 2의 이동 (d) Overlay 트리

그림 3. AMRoute의 효율성 문제 예

3. Ah hoc 네트워크에서의 새로운 CBT-기반 Overlay 멀티캐스트

2.3에서의 AMRoute에서의 효율성의 문제는 그림 3.(c)와 같이 네트워크 형상에 변화가 생겼어도 이에 대한 반응을 하지 못한 것에 기인한다. 즉, 중간 라우터의 이동으로 멤버 C와 D 노드 사이에 새로운 메쉬 링크를 생성하는 네트워크 형상으로 변화였으나, AMRoute는 멤버들의 변동 여부만을 조사하므로 멤버 C와 D가 여전히 서로를 인식하지 못하여 메쉬 링크를 형성하지 못한 것이다. 이를 해결하기 위해서는, 네트워크의 변화에 대하여 동적으로 새로운 멤버를 발견하고 멤버 간에 새로운 메쉬 링크의 존재 여부를 조사하는 것이 필요하다.

3.1 새로운 멤버 발견

본 논문에서는 송신하는 메시지의 IP 헤더의 초기 TTL 값을 지정할 수 있고, 수신한 메시지의 IP 헤더의 TTL 정보를 읽을 수 있다고 가정한다.

1) 추가된 메쉬 링크의 상태 정보

기존의 AMRoute 제어 메시지에 메쉬 링크로 연결된 이웃 멤버들의 정보를 알리기 위한 다음과 같은 필드를 추가한다.

① Option 필드 표시 : 이웃 멤버의 정보를 알리는 필드가 추가됨을 표시하는 플래그
② Option 부분 길이 : Option 부분의 전체 길이
③ 초기 TTL 값 : 해당 제어 메시지의 초기 TTL 값 정보를 기록
④ 코어 노드와의 거리 : 해당 제어 메시지를 생성한 멤버와 코어 노드와의 거리
⑤ 이웃 멤버 노드 수(메쉬 링크의 수) : 해당 메시지를 생성한 멤버의 메쉬 링크로 연결되어 있는 이웃 멤버의 수
⑥ 이웃 멤버 IP 주소 : 메쉬 링크로 연결된 이웃 멤버의 IP 주소
⑦ 이웃 멤버와의 거리 : 이웃 멤버로부터 수신한 제어 메시지를 분석하여 계산한 이웃 멤버와의 거리 기록
⑧ 이웃 멤버와 코어 노드와의 거리 : 주기적으로 코어 노드로부터 수신한 제어 메시지의 TTL 값을 분석한 이웃 멤버의 코어 노드와의 거리

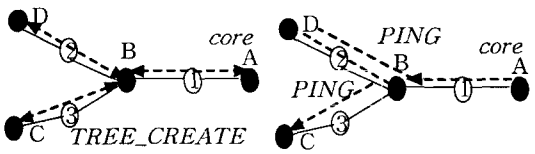
2) 추가된 새로운 제어 메시지

새로운 멤버 또는 직접 메쉬 링크로 연결되어 있지 않는 멤버와의 거리 변화를 조사하기 위하여 주기적으로 교환하는 PING 제어 메시지를 추가한다. PING 메시지의 주기는 다른 제어 메시지의 주기보다 2배 이상 크게 설정한다.

3) 멤버간의 거리

멤버간의 거리는 라우팅 과정에서 통과한 노드들의 홉 수를 의미한다. 이 홉 수는 수신한 제어 메시지의 IP 헤더에서 수집한 TTL 정보와 제어 메시지의 추가된 필드 중 송신 측에서 기록한 초기 TTL 정보를 비교함으로써 계산한다.

4) 새로운 멤버 발견



(a) 새로운 멤버 발견 (b) 발견한 멤버와의 거리

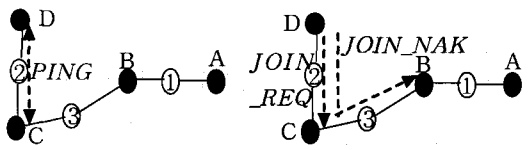
그림 4. 새로운 멤버 발견 및 거리 조사

그림 4.(a)와 같이 이웃 멤버로부터 수신한 제어 메시지에서 수집한 이웃 멤버 노드의 메쉬 링크 상태 정보를 자신의 메쉬 정보와 비교 분석하여 새로

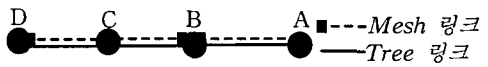
운 멤버를 IP 주소와 코어 노드와의 거리 그리고 이웃 멤버와의 거리등의 정보와 함께 발견할 수 있다. 새로운 멤버를 발견하면 후보 메쉬 멤버 테이블에 추가하고, 그림 4.(b)와 같이, 추가된 새로운 제어 메시지인 PING 메시지를 송수신하여 서로를 알리고 서로간의 거리를 분석하여 새로운 메쉬 링크를 생성할 수 있는지 조사한다.

3.2 새로운 메쉬 링크 발견

그림 3.(c)의 노드 2와 같이 멤버가 아닌 중간 라우터의 이동으로 네트워크의 변동이 발생하면, 멤버 D에서는 멤버 B로부터 수신하는 데이터나 제어 메시지의 TTL 값을 관찰하여 네트워크의 변동을 감지한다. 네트워크의 변동을 감지한 멤버는 그림 5.(a)처럼 후보 메쉬 멤버들과 PING 메시지를 이용하여 새로운 메쉬 링크의 존재 여부를 조사한다. 그림 5.(b)와 같이 멤버 C, D 사이의 메쉬 링크를 새로이 발견하면 멤버 D는 JOIN_REQ 메시지를 멤버 C로 유니캐스트 전송하여 새로운 메쉬 링크를 생성하고, 멤버 B로는 JOIN_NAK 메시지를 전송함으로써 메쉬를 재설정하고 주기적인 TREE_C REATE 메시지의 통하여 변화된 네트워크에 적합한 그림 5.(c)와 같은 트리를 형성하여 그림 3.(c)와 같은 AMRoute의 대역폭 낭비를 줄일 수 있다.



(a) 새로운 메쉬 링크 발견 (b) 메쉬 재설정



(c) 새로운 메쉬와 트리

그림 5. 새로운 메쉬 링크 발견

4. 결론 및 향후 과제

본 논문에서는 Ad hoc 네트워크에서의 Overlay 멀티캐스트 프로토콜인 AMRoute의 대역폭 낭비를 줄이는 방안을 제안하였다. 기존의 AMRoute의 제어 메시지에 메쉬 링크 상태 정보를 추가시켜 교환하여, 적은 오버헤드로 동적으로 새로운 이웃 멤버와 메쉬 링크를 발견하여 중복된 전송 경로를 피하는 메쉬와 트리를 재설정함으로써 멀티캐스트 성능 향상을 기대할 수 있다. 현재 전체 그룹 멤버들로 데이터 전달하는 과정에서 통과하는 라우터의 홉 수와 최대 지연 시간을 성능 평가 기준으로 NS2를 이용하여 실험을 진행하고 있다. 향후 서로 겹치지 않는 라우팅 경로를 가지는 링크들로 메쉬를 형성함으로써 트리의 견고함을 향상시킬 수 있는 연구를 진행할 것이다.

참고문헌

[1] M. Gerla and J. T. Tsai, "Multicluster, Mobile, Multimedia Radio Network," ACM-Blatzer Wireless Networks Vol.1, pp.255-265, 1995.

[2] E. Bommaiah and M. Liu, A. MvAuley and R. Talpa de, "AMRoute: Ad hoc Multicast Routing Protocol," Internet Draft, draft-manet-amroute-00.txt, 2002.

[3] E. M. Royer and C. E. Perkings, "Multicast Operations of the Ad-hoc On-Demand Distance Vector Routing Protocol," In Proceedings of ACM MOBICOM, 1999.

[4] S. J. Lee, M. Gerla and C. C. Chiang, "On Demand Multicast Routing Protocol," In Proceedings of IEEE WCNC'99, pp.1298-1302, September 1999.

[5] Y. B. Ko and N. Vaidya, "Location-based Multicast in Mobile Ad Hoc Networks," In Technical Report, Texas A&M, October 1998.

[6] C. K. Toh, G. Guichala and S. Bunchua, "ABAM: On-Demand Associativity-Based Multica st Routing for Ad hoc Mobile Networks," In Proceedings of IEEE VTC 2000, pp.987-993, Boston, MA., 2000.

[7] Chao Gui and Prasant Mohapatra, "Efficient Overlay Multicast for Mobile Ad Hoc Networks," Wireless Communications and Networking Conference (WCNC), 2003.

[8] A. Ballardie, "Core Based Trees (CBT) Multicast Routing Architecture", RFC 2201, September, 1997.

[9] Y. H. Chu, S. G. Rao and H. Zhang, "A Case for End System Multicast", In Proceedings of ACM SIGMETRICS, June 2000.

[10] Suman Banerjee, Bobby Bhattacharjee and Christopher Kommareddy, "Scalable Application Layer Multicast", ACM SIGCOMM, August 2002.