

Proactive Approach 기반 오버레이 멀티캐스트 트리 구성 및 복구방안

전진한*, 곽용완*, 남지승*

*전남대학교 컴퓨터정보통신공학과

e-mail: jhjeon23@naver.com

Overlay Multicast Tree Construction and Recovery Scheme based on Proactive Approach

Jin-Han Jeon*, Yong-Wan Kwag*, Ji-Seung Nam

*Dept of Computer Engineering, Chonnam National University

요 약

오버레이 멀티캐스트는 기존의 IP 멀티캐스트와 달리 Infrastructure의 도움 없이 멀티캐스트 기능을 수행하지만 갑작스런 노드의 이탈이나 장애 발생 시 트리의 복구가 필요하다. 본 논문에서는 이를 위해 각 노드의 Out-degree 자원을 미리 예약하여 트리 복구 시 예약된 노드에게 서비스를 요청하는 방안을 제시한다. 제시된 방안은 백업 노드로 인해 신속한 트리의 복구가 가능하다. 시뮬레이션을 통해 제시한 방안이 기존의 방안에 비해 보다 작은 복구 시간을 필요로 하며 부모 노드의 이탈로 인해 많은 수의 노드가 영향을 받는 상황에서 더욱 효과적으로 동작함을 보여주고 있다.

1. 서론

오버레이 멀티캐스트(Overlay Multicast)는 기존의 IP 멀티캐스트와 달리 Infrastructure의 도움 없이 말단의 호스트(End host)들로 별도의 가상 트리를 구성하여 멀티캐스트 기능을 제공한다. 그러나 오버레이 멀티캐스트는 트리를 구성하고 있는 중간 노드가 장애와 같은 사건으로 인해 멀티캐스트 그룹을 이탈 할 경우 트리를 재구성해야 하는 문제점을 가지고 있다.

기존의 오버레이 멀티캐스트에 관한 연구는 효율적인 멀티캐스트 트리의 구성을 위해 대부분 라우팅 프로토콜의 설계에 집중되었다[1-5]. 반면 오버레이 멀티캐스트 트리를 재구성 하는데 중점을 둔 연구는 상대적으로 드물게 진행되고 있다. 멀티캐스트를 이용하여 데이터를 전송하는 경우 최적의 트리 구성을 통해 데이터가 효율적으로 전송되는 것이 최고의 목표지만 실시간 방송 서비스나 화상회의와 같은 어플리케이션의 경우 중간 노드들의 이탈로 인한 트리 구성의 변화에도 말단에 존재하는 호스트의 영향을 최소화하도록 QoS를 보장하는 것이 최고의 목표가 된다. 이를 위해서는 어느 정도의 패킷 손실이 발생하더라도 신속한 트리 복구 방안이 필요하다.

오버레이 멀티캐스트 환경에서 트리를 복구하기 위한 기법은 크게 Reactive Approach와 Proactive Approach로 구분 할 수 있다. Reactive Approach는 노드가 트리를 이탈한 후에 복구를 시작하며 Proactive Approach는 사전에 복구계획을 세우고 노드가 이탈하면 세워진 계획에 의해 트리를 재구성하게 된다.

본 논문에서는 노드 간 최소 RTT(Round Trip Time)를 이용하여 멀티캐스트 트리를 구성하고 트리의 빠른 복구를 위해 Proactive Approach를 사용하여 실시간 방송과 같은 서비스에 적합한 오버레이 멀티캐스트 트리 구성 및 복구 알고리즘을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 기술하고 3장에서는 제안된 방안의 개요 및 동작을 설명한다. 4장에서는 시뮬레이션을 통한 기존 방안과의 비교 및 성능을 평가하고 5장에서 결론을 맺는다.

2. 관련 연구

Reactive Approach는 중간 노드 이탈 시 해당 노드의 특정한 조상 노드나 루트 노드에 재가입 하는 방법으로 트리를 복구하는 반면에 Proactive Approach는 각 호스트가 백업 루트를 가지고 있어서 부모 노드의 이탈에 적절히 대비한다.

PRM(Probabilistic Resilient Multicast)[11] 는 각 호스트가 무작위로 선별한 일정한 수의 다른 호스트들에게 작은 확률로 데이터를 재 전달하는 방법을 사용한다.

Okada의 기법[6]은 우선 최소 RTT를 이용하여 멀티캐스트 트리를 구성한 후 두 번째 최소 RTT값을 가지는 노드를 후보 부모 노드로 저장하여 백업 경로를 저장해 두는 방법을 사용한다.

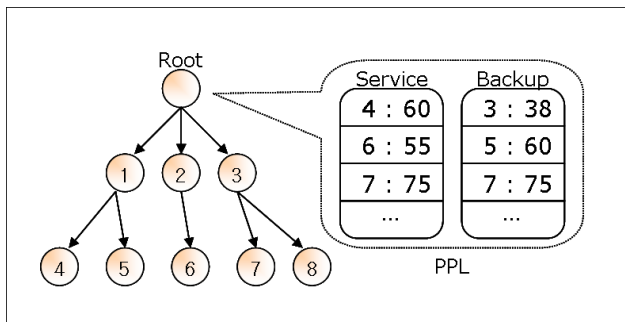
Yang이 제시한 기법[7] 은 중간 노드가 자신이 가지고 있는 자식 노드들을 위한 백업 경로를 사전에 선별하는 방법을 사용한다.

Kusumoto의 제안 기법[8]은 모든 노드들이 항상 1 이상의 여유 Out-degree 값을 가지면서 자식 노드와 조부모 노드 사이의 연결로 백업 경로를 설정한다.

3. 제안 방안

3.1. 개요

제안 모델에서는 루트 노드를 제외한 모든 노드는 서비스를 위한 Out-degree 외에 백업을 위한 Out-degree를 가진다. 한 노드가 각 노드가 Backup Out Degree를 별도로 가지고 있는 이유는 관련 연구에서 살펴보았던 Proactive approach의 또 다른 방안으로서 어느 한 노드의 부모 노드가 이탈 하게 되면 즉시 백업 노드에게 서비스를 요청하게 되고 백업 Out-degree로 예약해 두었던 자원을 서비스용으로 전환 하여 빠른 시간 안에 다시 서비스를 받게 할 수 있게 하기 위해서이다. 하지만 각각의 노드들이 백업 Out-degree 값을 가지게 되어 멀티캐스트 트리를 구성하게 되면 노드의 자원을 모두 활용하지 못해 전체적인 트리의 깊이가 길어지게 되고 이 트리가 가지게 되는 평균 RTT 값이 증가되게 되는 결과로 이어진다. 때문에 최상위 노드인 루트 노드의 Out-degree 만은 모두 서비스용으로 사용함으로써 이러한 단점을 완화시키게 되었다.

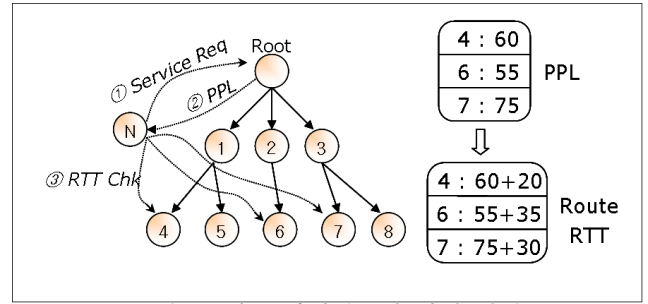


(그림 1) 루트노드의 PPL 구조

멀티캐스트 트리에서 루트 노드는 새로운 호스트가 멀티캐스트 그룹에 가입하고자 할 때 일차적으로 접속하게 되는 랑데부 포인트(Rendezvous Point)의 기능을 수행한다. 이때 루트 노드는 새로운 호스트에게 현재 그룹에 가입할 수 있는 정보를 주기위해 PPL (Portential Parent List)을 전송해 주는데 이 PPL의 구조는 그림 1과 같다. PPL은 서비스를 제공 해 줄 수 있는 노드의 IP 주소와 그 노드와 루트 노드까지의 RTT 값으로 구성된다. 일반적으로 PPL은 서비스를 제공 해 줄 수 있는 노드들에 대한 정보만을 가지고 있지만 제안 기법에서는 이 정보 외에 백업 Out-degree의 정보까지 가지고 있음으로서 노드들이 자신의 백업 노드를 선정할 수 있도록 해주고 루트 노드는 멀티캐스트 그룹에 새로운 노드가 가입하고 떠나거나 노드들이 가지고 있는 정보가 변경될 때 마다 이 정

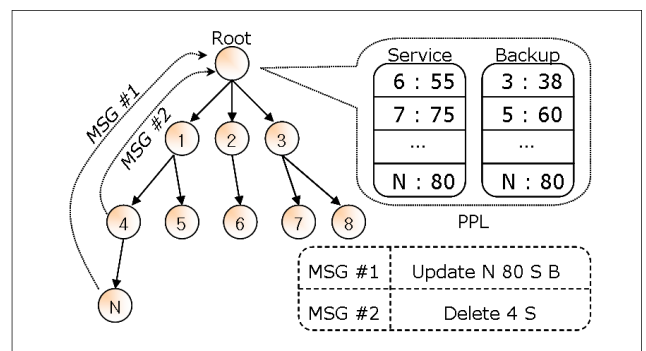
보를 갱신하여 항상 최신의 트리 정보를 유지하게 된다.

3.2. 새로운 노드 가입



(그림 2) 새 클라이언트의 가입 과정

그림 2는 새로운 호스트가 멀티캐스트 그룹에 가입하는 과정을 보여주고 있다. 새로운 호스트 N은 먼저 루트 노드에게 서비스를 요청하고 루트 노드는 자신의 Out-degree 정보를 확인한다. 만약 자신이 직접 서비스를 제공할 수 있는 상태라면 바로 서비스 요청을 수락하는 Ack 메시지를 전송하여 서비스를 제공하게 되지만 그렇지 못한 경우에는 그룹 멤버들로부터 수집한 PPL을 대신 전송하게 된다. 이를 수신한 노드 N은 이 리스트 내에 있는 노드들과의 RTT 값을 체크하는 작업을 수행하게 되고 이 때 얻어진 값을 PPL 내의 각 노드의 값과 합산하여 전체적인 경로의 RTT 값을 계산한다. 이렇게 얻어진 값은 적은 순서대로 정렬이 되어 순차적으로 서비스를 요청하게 되는데 만약 하나 이상의 노드들이 동시에 그룹에 가입하는 작업을 하고 있었다면 성능이 좋은 특정한 노드가 동시에 다수의 노드로부터 서비스 요청을 받게 되는 상황이 발생할 수 있다. 이 경우 가장 먼저 서비스를 요청한 노드 순으로 서비스를 제공하고 더 이상 서비스 요청을 받아들일 수 없는 상태가 되면 Nack 메시지를 통해 서비스가 불가능함을 알린다. 이를 수신한 노드는 정렬된 PPL로부터 그 다음 노드를 선택하여 다시 서비스를 요청하는 작업을 수행하게 된다.



(그림 3) PPL 업데이트 과정

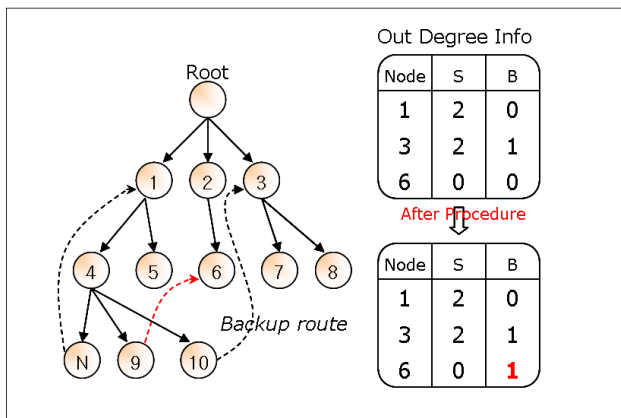
노드 N이 자신의 부모 노드를 찾아 서비스를 받게 되면 그림 3과 같이 새로 멀티캐스트 그룹에 가입한 노드 N과 Out-degree 정보가 변경이 된 노드 4는 루트 노드에게 자신들의 새롭게 변경된 정보를 알려준다. 새롭게 가

입한 노드 N 은 자신이 가지고 있는 전체 RTT 값과 서비스와 백업 Out-degree 값에 여유가 있음을 알려주는 메시지를 전송하게 되고 새로운 자식 노드를 가지게 된 노드 4는 더 이상 자식 노드를 수용할 수 없게 될 경우 PPL로부터 자신을 삭제하라는 메시지를 전송한다. 이러한 방법으로 루트 노드는 항상 그룹 멤버들의 최신의 정보를 가지고 멀티캐스트 트리를 구성할 수 있는 준비를 할 수 있게 된다.

3.3. 트리 복구

멀티캐스트 그룹에 새로 가입한 호스트는 부모 노드의 이탈에 대비하여 백업 노드를 찾는 작업을 수행한다. 이때 각 노드들에 할당된 백업 Out-degree가 이용된다. 백업노드를 찾는 과정은 다음과 같다.

- i) 자신의 Grandparent 노드와 RTT를 측정
- ii) 자신이 부모 노드의 유일한 자식 노드이면 백업 노드는 Grandparent 노드로 선정
- iii) 새로 부모 노드에 접속한 노드 또한 Grandparent 노드와 RTT를 측정하여 기존 노드의 RTT와 비교
- iv) Grandparent 노드와 작은 RTT를 가진 노드가 Grandparent 노드를 백업 노드로 설정
- v) 큰 RTT를 가진 노드는 루트 노드의 PPL 백업 정보를 받아서 부모 노드를 찾는 과정과 동일하게 최적의 경로를 가지는 노드를 선택하여 백업 노드 요청
- vi) 백업을 수락한 노드는 백업 Out-degree를 증가시키고 더 이상 백업 요청을 받을 수 없을 경우 루트 노드로 자신을 백업 PPL에서 삭제 요청하는 메시지 전송



(그림 4) 백업 노드 선정 과정

그러나 백업 노드 선정 시 제약사항이 필요하며 자세한 제약 조건은 다음과 같다.

- i) 부모 노드와의 백업 연결
- ii) 형제 노드들과의 백업 연결
- iii) 자신의 하위 노드들과의 백업 연결
- iv) 백업 노드들 간의 루핑(looping)이 발생하는 백업 연결

4. 성능 평가

4.1. 시뮬레이션 환경

제안 방안의 성능을 평가하기 위해 50개의 노드를 생성하여 각 노드들의 지연 (Delay)시간을 30ms 에서 90 ms 사이의 임의의 값으로 주었다. 한 개의 노드가 가지는 최대 Out-degree 값은 100Mbps의 네트워크 환경에서 25Mbps 정도의 대역폭이 필요한 HD 급 영상을 서비스한다고 가정하여 4로 정의 하였다.

시뮬레이션의 성능 비교를 위해 관련 연구에서 살펴보았던 Yang의 제안 기법과 Kusumoto의 제안 기법을 동일 환경 하에서 테스트 하였다. 단 복구 알고리즘 상의 비교를 위해 멀티캐스트 트리의 구성 알고리즘은 본 논문에서 제안한 각 경로가 가지는 최소 RTT 값을 이용하여 트리를 구성하는 방법으로 동일하였다.

4.2. 시뮬레이션 결과

트리 구성 단계에서는 트리를 구성하는 노드의 수를 10개부터 50개까지 증가시켜가며 이 때 구성이 된 트리의 평균 RTT 값을 측정하였고 이렇게 구성이 된 트리에서 특정 한 노드를 이탈 시켰을 때 변화된 평균 RTT 값과 트리의 복구 시간을 측정하였다.

그림 5에서는 제안된 기법과 Yang의 기법이 적은 수의 노드로 트리가 구성이 될 때에는 값의 차이가 없다가 노드의 수가 증가해 가면서 그 차이가 생기는 반면 Kusumoto의 기법은 초기부터 성능이 비교적 좋지 않음을 보여주고 있는데 이는 트리 구성 시 사용할 수 있는 Out-degree 값의 제한 때문이다. Yang의 기법에서는 한 노드가 사용할 수 있는 Out-degree 값을 모두 서비스용으로 사용하는 반면 Kusumoto의 기법에서는 모든 노드가 항상 1이상의 여분의 Out-degree 값을 가지고 있어야 한다. 이러한 이유로 같은 수의 노드로 트리를 구성하게 되면 Out Degree 값을 적게 사용하는 쪽이 상대적으로 트리의 깊이가 증가하여 전체적인 RTT 값을 증가 시키는 결과로 이어진다. 반면 제안된 기법에서는 Kusumoto의 기법과 같이 백업 노드를 위해 1이상의 여분의 값이 필요하지만 최상위 노드인 Root 노드의 Out Degree 값만은 모두 서비스용으로 사용하게 함으로서 평균 RTT 값의 차이를 줄일 수 있게 되었다.

그림 6과 그림 7은 트리가 구성될 때 2번째로 가입한 노드를 이탈시켜 복구 후 변화된 평균 RTT 값과 복구에 소요된 시간을 측정한 것이다. 이 시뮬레이션에서는 상위 노드를 이탈시킴으로서 그 영향을 받는 노드의 수가 변화할 때 미치는 영향을 알아보려고 하였다. 적은 수의 노드로 트리가 구성이 되었을 때에는 제안된 기법과 Yang의 기법 사이에 차이는 근소하지만 노드의 수가 증가 할수록 제안된 기법의 성능이 더 좋을 수 있는데 이는 복구가 필요한 노드의 백업 경로가 제안된 기법에서

는 비교적 상위 노드에 분포하고 있지만 Yang의 기법에서는 부모 노드의 자리를 차지한 한 개의 자식 노드를 제외 하면 모두 하위에 위치한 노드에게서 다시 서비스를 받기 되는 확률이 크기 때문이고 이 때 복구 작업이 하위 노드로 계속 내려가면서 발생하게 되는 시간 때문에 복구 시간 또한 길어지게 된 결과이다.

백업 Out-degree를 이용하여 트리를 재구성하는 방안을 제안하였다. 제안된 방안은 중간 노드의 이탈 시 Proactive Approach를 적용하여 새로운 부모 노드를 찾는 과정 없이 바로 서비스를 제공이 가능하다. 그러나 백업을 위해 각 노드들의 Out-degree를 예약하기 때문에 트리 구성 시 성능이 저하되지만 루트 노드의 Out-degree를 모두 서비스를 위해 할당하여 성능 저하를 최소화 하였다.

또한 시뮬레이션을 통해 기존의 방안들과 비교한 결과 제안 방안이 트리의 복구에 가장 작은 시간이 소요되었고 이탈이 빈번한 노드들로 멀티캐스트 트리를 구성해야 하는 상황에서 효율적임을 보여주고 있다.

참고문헌

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

[2] D. Pendarakis, S. Shi, D. Verma, M. Waldvogel, "ALMI: An Application Level Multicast Infrastructure", 3rd USENIX Symposium on Internet Technologies and Systems, Mar 2001.

[3] Y. Chawathe, S. McCanne, E. Brewer, "Scattercast: An Architecture for Internet Broadcast Distribution as an Infrastructure Service" PhD Thesis, University of California, Berkeley, 2000

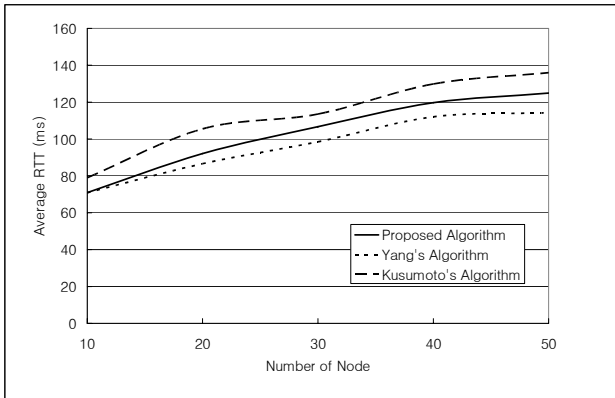
[4] P. Francis, "Yoid: Extending the Internet Multicast Architecture", <http://www.icir.org/yoid/> 1999

[5] J. Jannotti, D. Gifford, K. Johnson, M. Kaashoek, J. O'Toole, "Overcast: Reliable Multicasting with an Overlay Network", 4th Symposium on Operating Systems Design & Implementation, Oct. 2000

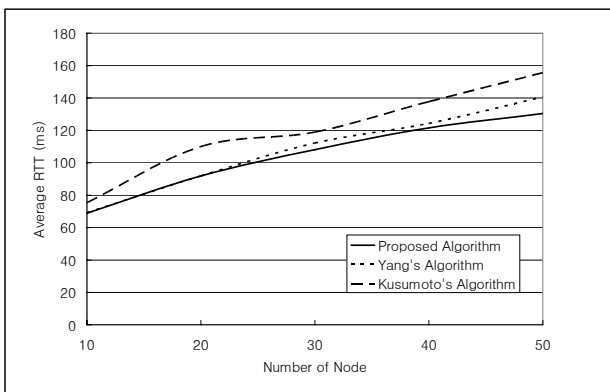
[6] Y. Okada, M. Oguro, J. Katto, S. Okubo, "A New Approach for the Construction of ALM Trees using Layered Video Coding", in Proceedings of ACM Multimedia 2005

[7] M. Yang, Z. Fei. "A Proactive Approach to Reconstructing Overlay Multicast Trees", in proceedings of INFOCOM 2004, March

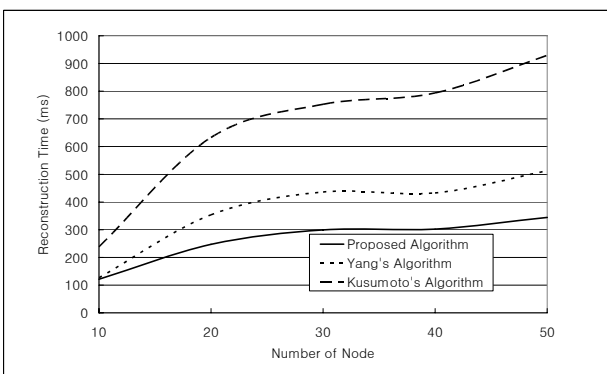
[8] T. Kusumoto, Y. Kunichika, J. Katto, S. Okubo, "Proactive Route Maintenance and Overhead Reduction for Application Layer Multicast", ICAS/ICNS 2005



(그림 5) RTT 변화



(그림 6) 트리 재구성 후 RTT 변화



(그림 7) 트리 재구성 시간의 변화

5. 결론

본 논문에서는 오버레이 멀티캐스트에서 말단 노드가 아닌 중간 노드의 이탈 시 발생하는 문제점 해결을 위해