

PIM-DM 멀티캐스트에서의 가입 지연시간의 개선

김한수* 장주욱
 서강대학교 전자공학과
 {kutestar, jjang}@ecca1.sogang.ac.kr

Improvement of Join Latency Time in PIM-DM Multicast

Han-Soo Kim* Ju-Wook Jang
 Dept. of Electronic Engineering, Sogang Univ.

Abstract

One of the remarkable problems in PIM-DM(Protocol Independent Multicast-Dense Mode) is the latency time, increasing for some periods. The reason of this problem is proved to the confusion of flooding prune message and leave prune message. We propose a convincing solution to this confusion and we prove this by network simulation.

요 약

PIM-DM을 이용하여 멀티캐스트를 하였을 때 그룹에 가입하기 위해서 대기하는 시간이 일정시간마다 길어지게 되는 문제점이 있다. 본 논문에서는 이러한 문제점을 해결하기 위해서 그 원인을 분석하고, 이에 대한 해결방법을 제시하였다. 제한된 기법의 성능 향상을 네트워크 시뮬레이션을 통해서 확인하였다.

1. 서 론

네트워크 자원이 풍부하고 구성원들이 특정 장소에 밀집되어 있는 LAN 환경에서 주로 이루어지는 PIM-DM(Protocol Independent Multicast-Dense Mode)[1]은, 수신자가 그룹에 가입하여 멀티캐스트 서비스를 받고자 할 때 일정시간마다 그 가입이 지연되는 문제가 있다. 이는 멀티캐스트 그룹의 트리 구성을 위해 일정시간마다 실시하는 플러딩(flooding)에 대한 응답으로 가입자 및 하위 라우터가 송신하는 플러딩 프룬(prune)과, 가입자가 그룹 탈퇴를 위해 송신하는 리브 프룬(leave prune)을 상위 라우터가 구분하지 못하여 생기는 것이다.

본 논문에서는, 플러딩이 주기적으로 이루어지므로 플러딩 프룬도 주기적이지만 리브 프룬은 주기적이지 않다는 점에 착안하여 두 프룬 메시지를 구분하는 알고리즘을 제시하였다. 이로 인해 가입자의 가입 지연을 없앨 수 있었고, 이를 네트워크 시뮬레이션을 통하여 확인하였다.

2. 문제점 및 원인분석

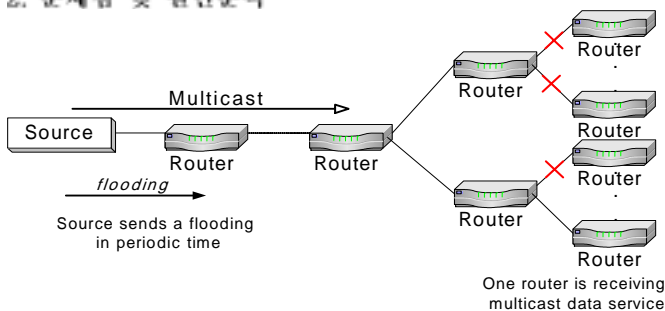


그림 2-1 PIM-DM 플러딩과 트리 구성

그림 2-1에서, 근원지는 멀티캐스트를 시작할 때와 정해져 있는 프룬 시간이 되면 네트워크에 있는 모든 라우터들에게 플러딩(flooding)을 실시해서 멀티캐스트 트리를 유지하거나 재구성하게 된다[2]. 그림 2-2와 같이, 만약 모든 인터페이스가 프룬 메시지를 받았을 때는 자신도 근원지 방향 인터페이스로 프룬을 돌려보낸다. 또, 자신이 서비스를 받지 않고 있을 경우 멀티캐스트 그룹에 가입하고 싶을 경우에는 그래프트(graft) 메시지를 근원지까지의 최적경로 인터페이스로 보내서 멀티캐

스트 데이터를 수신한다. 그리고 서비스를 받다가 중지하고 싶을 경우에도 프룬 메시지를 돌려보낸다.[3]

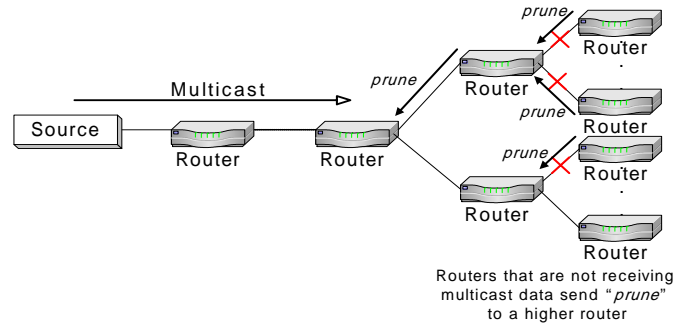


그림 2-2 하위 라우터들의 플러딩에 대한 프룬 메시지 전송

이 과정에서, 그래프트 메시지를 보냈을 때 그룹에 가입하여 서비스를 받기 위해 대기하는 시간이 일정 간격으로 길어지는 문제가 발견되었다[4]. 이는 멀티캐스트 정보를 보존하기 위해서, 즉 트리 구성을 위해서 일정시간마다 실시하는 플러딩에 대한 프룬 메시지와 멀티캐스트 그룹에서 탈퇴하기 위한 프룬 메시지가 구별되지 않고 같은 메시지로 간주되기 때문이라고 볼 수 있다. 이렇게 되면 각 라우터들은 프룬을 받은 인터페이스 이하로 멀티캐스트 서비스를 중단하여, 그 이하에 있는 사용자들의 가입 메시지를 다음 플러딩이 있을 때까지 전달하지 못하게 되는 것이다.

각 라우터의 전달 지연 시간을 10ms로 동일하게 놓고, 플러딩의 주기를 1초(1000ms)로 설정하여 시간에 따른 변화를 살펴보면 그림 2-3과 같다. 0ms에서 근원지에서 플러딩을 실시하여, 40ms가 될 때 최하위 라우터에 도달한다. 이 때 최하위 라우터에서 서비스를 받고 있던 수신자가 탈퇴하여, 상위 라우터로 프룬 메시지를 송신하게 된다. 이 직후에 가입하고자 하는 수신자는, 60ms 이후와 같이 가입이 이루어지지 못하고 1040ms가 지난 후, 즉 다음 플러딩 때 비로소 가입하게 되는 것이다.

3. 문제점의 해결방안

이 문제를 해결하기 위한 방법으로, 우선 플러딩 프룬과 리브 프룬이 수신되는 특성에 대하여 살펴보았다. 그 특징은, 플러딩이 일정시간의 주기성을 가지고 실시되므로, 이에 대한 응

답인 플러딩 프론트도 주기성을 가지고 수신된다는 것이다. 이에 비해 리브 프론트는 주기성을 갖고 있지 않다. 즉, 일정치 않은 순간에 가입자의 의도대로 이루어진다는 것이다. 따라서 주기적이지 않은, 즉 플러딩 프론트가 오지 않던 인터페이스에서 최초로 발생하는 프론트 메시지는 리브 프론트로 생각할 수 있다. 이에 기초하여, 라우터는 주기적으로 수신되는 프론트 메시지는 플러딩 프론트로, 그렇지 않은 메시지는 리브 프론트로 간주하여 데이터를 처리하면 될 것이다. 만약 그 이후에 같은 인터페이스에서 플러딩의 주기와 같은 시간경과 후에 프론트가 수신된다면, 그때부터는 플러딩 프론트로 인지하여 멀티캐스트 데이터의 송신을 멈춘다.

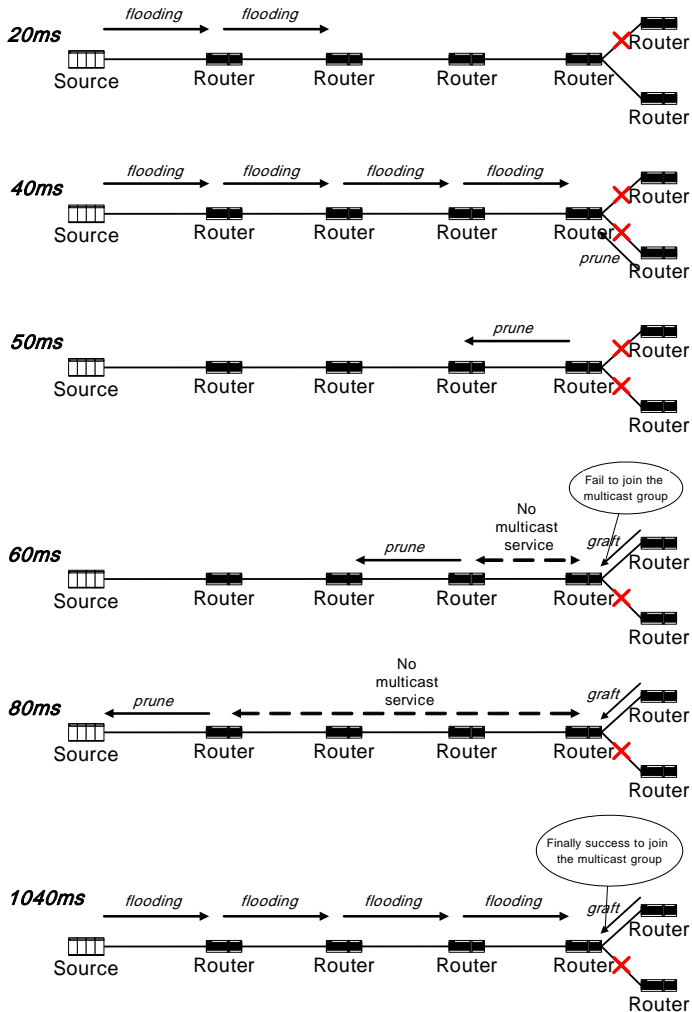


그림 2-3 시간에 따른 그룹 가입 지연의 발생모습

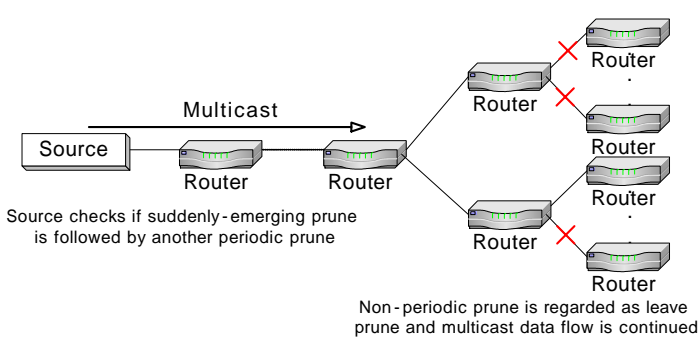


그림 3-1 최초의 프론트 메시지를 리브 프론트로 간주, 프리를 제거하지 않아 가입자의 대기 시간 없이 멀티캐스트 데이터 전송

그림 3-1은 이와 같은 알고리즘으로 동작하는 네트워크이다. 프론트 메시지가 상위 라우터로 전송된 후에 그래프트 메시지가 수신되었을 때, 그 상위 라우터는 이 프론트 메시지를 리브 프론트로 간주하고 멀티캐스트 데이터를 계속 송신한다. 따라서 뒤늦게 가입한 수신자는 대기시간 없이 데이터를 수신할 수 있는 것이다. 그림 2-3에서, 이와 같은 알고리즘을 도입하였을 때 얻을 수 있는 변화는 그림 3-3과 같다. 40ms가 되었을 때 최하위 수신자가 프론트 메시지를 라우터로 보내도, 그 라우터는 주기적이지 않고 갑자기 수신되는 프론트를 리브 프론트로 간주하여 상위 라우터로 프론트 메시지를 보내지 않는다. 따라서, 그림의 50ms에서 이 라우터는 계속 멀티캐스트 서비스를 받고 있는 것이다. 따라서 60ms에 다른 수신자가 그룹에 가입하고자 하는 메시지를 보냈을 때, 가입 지연 없이 바로 서비스를 받을 수 있게 되는 것이다.

```

InitFlooding()
    S = 0 // interface to the multicast source
    I = 0 // interface of received multicast packet
    G = 0 // interface of joined multicast group
    A = 0 // all interfaces

    F = FLOODINGFRQ // Flooding period of the source
    p = 0 // index of whether first prune is received

MakeTree(I)
// function of handling tree messages

    if (I == prune) and (G > 1)
        RemoveGroup(I from G)
    if (I == prune) and (G == 1)
        if (p == 1) and (ReceiveTimeGap(I) == F)
            p = 0
            RemoveGroup(I from G)
            SendPrune(S)
        if (p == 0)
            p = 1

    if (I == Flooding)
        if (G == EMPTY)
            SendPrune(S)
        if (G != EMPTY)
            SendFlooding(G)

    if (I == graft)
        AddGroup(I to G)
        if (G == EMPTY)
            SendGraft(S)
    
```

그림 3-2 제안된 알고리즘의 pseudo code

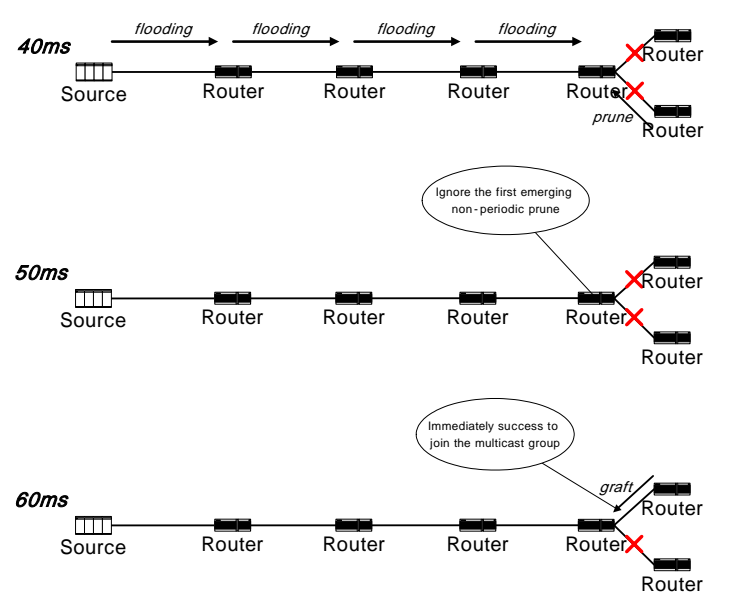


그림 3-3 지연시간 없이 그룹에 가입하는 수신자의 시간에 따른 모습

4. 실험 및 결과

기존에 구현되어 있는 PIM-DM 프로토콜을 분석하여, 프룬이 수신되었을 때 동작하게 되는 클래스의 부분에 위에 제안된 방법을 삽입하여 개선된 프로토콜을 적용하게 되면 제시된 문제점이 개선되고 수신자가 그룹에 가입하는 대기시간이 현저하게 줄어들게 된다.

제안한 개선 방법의 성능을 확인하기 위해서 버클리 대학의 ns-2 시뮬레이터를 사용하였다[5]. 이 시뮬레이터에 구현되어 있는 PIM-DM 프로토콜을 사용하여 실험하였고, 실험에 사용한 토폴로지는 그림 4-1과 같다. 라우터는 N0부터 N10, 각 링크는 모두 1.5Mb, 10ms, droptail로, 근원지는 N5, 수신단은 N10으로 설정하였다.

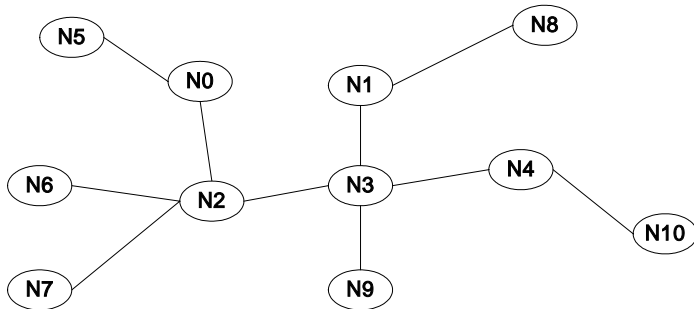


그림 4-1 실험에 쓰인 네트워크 구성의 위상도

실험은 N10에서 각각 평균 0.05초와 0.1초 간격으로 2개의 가입자가 가입과 탈퇴를 반복하는 것으로 설정하였다. 그림 4-2, 4-3, 4-4는 위와 같은 설정에 따른 실험 결과를 그래프로 나타낸 것이다.

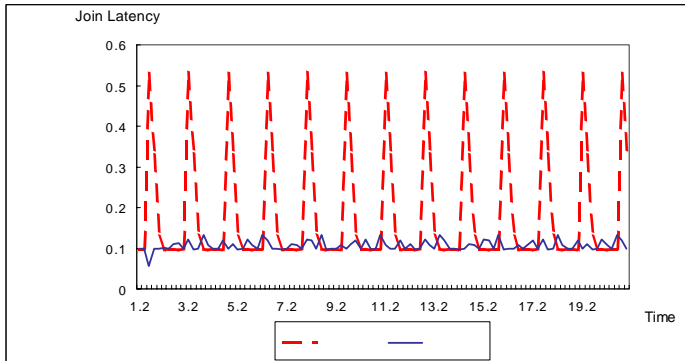


그림 4-2 Prune time이 0.5초일 때의 비교 그래프

실험은 prune time이 0.5초, 1.0초, 2.0초일 때로 실행하였다. 그림 4-2에서 x축은 시뮬레이션이 진행된 시간이고 y축은 N10에서 그룹에 가입하겠다는 메시지를 전송한 후에 처음으로 멀티캐스트 데이터를 받을 때까지의 시간을 측정한 join delay를 나타낸 것이다. 굵은 점선은 현재 시행되고 있는 멀티캐스트 PIM-DM 방식을 그대로 적용한 것이고, 실선은 본 논문에서 제안된 개선 방법을 적용했을 때의 실험결과이다.

그림 4-3, 4-4에서 아래로 내려간 부분의 경우에는 그룹에 가입하기 위해서 그래프 메시지를 전송하는 순간에 플러딩이 발생해서 빠른 시간에 가입된 경우이다.

그림들을 보면 일정주기마다 시간이 늘어가는 것은 완전하게 개선하지 못하였다. 하지만 기존의 방법은 평균 0.16초의 시간 지연을 갖는 반면에 개선실험은 평균 0.11초로 약 37.44%의 개

선 효과를 얻었다. 또한, 플러딩 시에 그래프 메시지에 대한 대기 시간은 그래프에서 알 수 있는 바와 같이 현저하게 감소되었음을 알 수 있다.

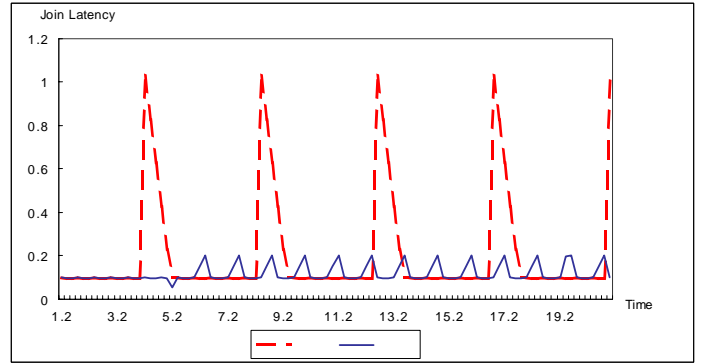


그림 4-3 Prune time이 1.0초일 때의 비교 그래프

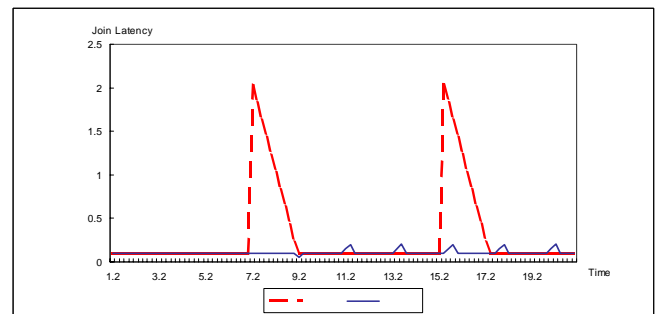


그림 4-4 Prune time이 2.0초일 때의 비교 그래프

5. 결론 및 추후과제

본 논문에서는 PIM-DM 프로토콜에서 수신자가 그룹에 가입할 때 지연되는 시간에 대한 문제점과 원인을 살펴보고, 그것을 보완하기 위한 방법을 제안하였다. 제안된 방법으로 시뮬레이션한 결과, 이 방법은 기존의 방법보다 약 37.44%의 개선 효과를 보였다. 차후, 실험 데이터에 기초한 그래프에서 볼 수 있는 소량의 대기시간의 증가에 대한 연구가 있어야 하겠다. 또, 프룬 메시지가 주기적으로 수신된다고 가정하였는데, 이는 PIM-DM 프로토콜의 특성상 자원이 풍부한 근거리 네트워크를 기반으로 동작한다고 가정하여 이론 수립 및 실험을 통한 결과 측정이 가능했지만, 네트워크의 급작스런 변화 등에 대한 고려가 없어 WAN 환경에서의 적용에는 어려울 것이라 생각된다. 또한, 플러딩에 대한 프룬과 그룹 탈퇴를 위한 리브 프룬을 헤더에 각각 다르게 표시해 줌으로써 플러딩 프룬과 충돌되는 것을 막는 방법이 가장 정확하고 확실한 해결 방법이라 생각된다.

6. 참고 문헌

- [1] Williamson, Beau, *Developing IP Multicast Networks, Volume 1*, Cisco Press, 2000.
- [2] S. Deering, *Protocol Independent Multicast Version 2 Dense Mode Specification*, IETF, June 7, 1999
- [3] A. Adams, J. Nicholas and W. Sisdak, *Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)*, IETF, February 15, 2002
- [4] J. Kim and J. Jang, *Analysis and Improvement for latency time of PIM-DM*, KISS, 2001
- [5] Network Simulator Document by U.C Berkely <http://www.isi.edu/nsnam/ns/>