

# 다이내믹한 터널을 이용한 REUNITE 개선방안

정철<sup>1)</sup> 김중권  
서울대학교 컴퓨터 공학부  
(chjeong, ckim)@popeye.snu.ac.kr

## Improved REUNITE using dynamic tunnel

Chul Jeong<sup>1)</sup> Chong-Kwon Kim  
School of Computer Science and Engineering, Seoul National University

### 요 약

최근에 제안된 멀티캐스트 프로토콜인 REUNITE는 규모 확장성(scalability) 문제를 해결하고 새로운 주소 배정(addressing) 방식을 도입하였다 [1]. 그러나 REUNITE는 많은 IP Lookup 작업을 필요로 하는 문제를 갖는다. REUNITE에서는 라우터로 들어오는 패킷이 멀티캐스트 패킷인지 유니캐스트 패킷인지를 구별하지 않는다. 따라서 어떤 패킷이라도 REUNITE를 지원하는 라우터에 들어오게 되면 부가적인 테이블 Lookup 과정을 수행하게 된다. 본 논문에서는 라우터에서 멀티캐스트 패킷을 구별하여 기존의 유니캐스트 패킷의 전송속도에는 영향을 미치지 않게 하였고, 멀티캐스트 패킷의 경우에도 branching 노드간에 형성된 터널을 통해 branching 노드에서만 테이블 Lookup 작업을 수행하도록 하였다.

### 1. 서론

IP 멀티캐스트(Multicast)가 제안된 지 10여년이 지나고 수많은 연구와 발전이 있었으나 현재의 Internet 망에 도입하기에는 여전히 여러 문제점이 남아있다. 특히 현재의 라우팅 프로토콜은 각각의 라우터가 모든 그룹에 대한 멀티캐스트 포워딩 테이블 엔트리를 저장하고 있어야 한다. 그러므로 그룹의 수가 증가하게 되면 자연히 멀티캐스트 포워딩 테이블도 증가하게 되는 규모 확장성(Scalability) 문제가 발생하게 된다.

본 논문에서 근간으로 하고 있는 프로토콜은 REUNITE[1]로 이러한 규모 확장성 문제를 해결하는데 중점을 두고 있다. REUNITE에서는 컨트롤 메시지를 이용하여 데이터 필드와 컨트롤 필드를 나누어 그룹 내의 라우터의 위치에 따라 다른 방식으로 해당 정보를 저장하여 어떤 패킷이 라우터에 도달하면 이 정보를 이용, 가입된 수신자에게 패킷을 전달한다. 여기서 그룹의 수신자 정보를 담고 있는 MFT 필드는 멀티캐스트 그룹의 branching 노드에서만 저장함으로써 규모 확장성 문제를 해결하였다. 그러나 REUNITE를 지원하는 라우터는 자신을 지나가는 대부분의 패킷인 유니캐스트 패킷 역시 자신의 MFT 필드를 검색한 후에 IP Lookup을 수행하기 때문에 패킷 전송 성능이 저하된다.

본 논문에서는 패킷 중에 멀티캐스트 패킷만을 구별하여 branching 노드에서만 MFT 필드를 확인하도록 하는 방법을 제시한다. 이러한 방법을 구현하기 위해 기존의 JOIN 메시지를 수정하여, 그룹에 가입하려는 새로운 수신자가 있을 때마다 생기는 branching 노드를 상위

branching 노드에게 알려주는 방식을 이용함으로써 branching 노드간에 다이내믹한 터널이 생성되도록 하였다.

본 논문의 나머지 부분은 다음과 같이 구성된다. 2장에선 REUNITE의 동작 방식에 대해서 간략하게 설명한다. 3장에서는 REUNITE의 문제점들을 제안하고 본 논문에서 제안한 개선된 REUNITE와 터널의 생성방법을 알아본다. 4장에서는 결론과 향후 진행될 연구 방향을 제시한다.

### 2. REUNITE

REUNITE가 기존의 멀티캐스트 라우팅 프로토콜과 다른 가장 큰 특징은 데이터 포워딩과 그룹 확인을 위해 class D 주소를 사용하는 것이 아니라, 일반적으로 사용하고 있는 유니캐스트 IP 주소를 사용한다는 것과 branching 노드의 개념을 도입하여 branching 노드에서만 group state를 저장하도록 하는 것이다. 이렇게 함으로써 전체 라우터에서 저장해야 하는 포워딩 상태 정보를 줄임으로써 규모 확장성을 지원하고 class D IP 주소를 사용하지 않음으로써 쉽게 유일한 그룹 주소를 부여할 수 있다. 또한 모든 라우터에서 REUNITE를 지원할 필요도 없으며, 송신자를 확인함으로써 액세스 컨트롤(access control)을 지원한다.

위 방법을 사용하여 멀티캐스트 방식을 구현할 수 있는 이유는 다음과 같다. 멀티캐스트의 그룹 수가 증가하면 그룹의 대부분이 매우 희박하게 분산된다. 이때 그룹의 데이터 전송 트리(data delivery tree)는

본 연구는 한국과학재단 목적기초연구(1999-1-302-005-3)지원으로 수행되었음.

대부분이 non-branching 라우터들로 구성된다 [1]. 따라서 branching 되는 라우터에서만 포워딩 정보를 가지게 함으로써 규모 확장성 문제를 해결할 수 있는 것이다.

2.1 주소 배정과 포워딩 알고리즘

A. 주소 배정

앞에서도 말했듯이 REUNITE에서는 주소를 배정할 때 기존의 IP 주소를 그대로 사용한다. 각 그룹에는 관련된 루트 노드가 존재한다. 멀티캐스트 그룹은 루트의 주소와 포트를 이용하여 구별된다.

(<root\_IP\_address, root\_port\_number>)

이 경우 진역적으로 유일한 그룹을 만드는 것은 단순히 루트가 지역적으로 유일한 포트 번호를 배정해 주는 것이다.

B. 포워딩 알고리즘

각 멀티캐스트 그룹에 있어서 REUNITE는 루트 노드를 근간으로 전송 트리(delivery tree)를 생성한다. 각각의 REUNITE를 지원하는 branching 라우터는 MFT(Multicast Forwarding Table)를 저장하고 있다.

MFT Format

<root_address,root_port>	<dst, stale>
<rcv1, alive>, <rcv2, alive>, .....	

- dst : 모든 수신자중에서 그룹에 가입한 첫 번째 수신자의 IP 주소
- rcvi : 라우터가 dst로 지정된 그룹으로부터 멀티캐스트 패킷을 받았을 때 복사하여 전달할 수신자들의 주소
- stale , alive : MFT state는 soft state이기 때문에 각 수신자의 가입여부를 알려주는 boolean 값

2.2 REUNITE 트리 구성, 유지 및 데이터 전송

각 branching 라우터는 멀티캐스트 포워딩 트리를 정의하기 위해 MFT 필드에 그룹당 state를 가지고 있다. Non-branching인 라우터에서는 MCT(Multicast Control Table)필드에 자신이 멀티캐스트 그룹의 한 부분이라는 것을 알리기 위해 값을 저장하고 있다. 이러한 값들은 컨트롤 메시지에 의해 생성되거나 삭제된다.

REUNITE에서 멀티캐스트 그룹 트리를 구성하는데 사용하는 컨트롤 메시지는 JOIN 메시지와 TREE 메시지가 있다. JOIN의 경우 주기적으로 각 수신측에서 루트를 향해 유니캐스트 되는 반면 TREE 메시지는 멀티캐스트 전송 트리를 따라 루트에서 멀티캐스트 된다.

- JOIN : MFT 안의 수신자 엔트리를 생성 혹은 갱신(refresh)한다.
- TREE: MCT 엔트리를 생성 또는 갱신, MFT 안의 그룹 엔트리를 갱신한다.

MCT Format

<root_address,root_port>	<dst>
--------------------------	-------

데이터 패킷의 전송 과정은 다음과 같다. 소스에서 데이터가 <dst>를 향해서 유니캐스트로 전송되면 REUNITE를 지원하는 각 라우터는 패킷 헤더를 검사하여 MFT 필드에서 관련된 그룹 엔트리가 있는가를 확인한다. 엔트리가 존재하면 rev에 따라 데이터를 복사하여 해당 주소로 전송하고, 엔트리가 존재하지 않으면 IP Lookup 과정을 통해 지정 주소로 downstream된다.

3. REUNITE의 개선

앞장에서는 REUNITE에 대해서 간략하나마 알아보았다. REUNITE의 장점 중 하나는 현재의 Internet 망에 큰 변화를 주지 않는 상태에서 바로 사용할 수 있다는 점이다-REUNITE를 지원하지 않는 라우터에서는 IP Lookup을 통해 데이터를 전송하면 된다. 그러나 여기서 생각할 점은 현재의 인터넷 전송방식은 유니캐스트로 이루어지고 있다는 것이다. 즉 멀티캐스트 그룹의 일부분으로 형성된 라우터를 지나가는 대부분의 패킷들은 유니캐스트 패킷이다. 따라서 대부분의 유니캐스트 패킷들은 MFT 필드를 검색할 이유가 없다. 그러나 REUNITE에서는 기존의 유니캐스트 패킷과 멀티캐스트 패킷을 구별할 수 없기 때문에 부가적인 테이블 검색과정과 IP Lookup 과정을 수행하게 된다. 만약 소스와 목적지 간의 거리가 멀다면 MFT 필드의 검색과정이 Exact match방식을 따른다고 할 지라도 심각한 성능 하락을 유발할 것이다. 따라서 본 장에서는 branching 노드들 간의 터널을 다이나믹하게 만들어 non-branching 노드에서의 불필요한 IP Lookup을 제거한 방법을 제시할 것이다.

3.1 변경 사항

우리가 제안한 방식은 JOIN메시지와 MFT 필드의 구성을 기존의 것과 변경하였다. JOIN 메시지는 기존의 정보 이외에 추가로 branching 노드에 대한 정보를 담았다. 상위로 올라갈 수 있는 JOIN 메시지는 기존의 방식과 달리 나중에 가입한 수신측의 JOIN 메시지가 상위로 올라 갈 수 있도록 함으로써 branching 노드가 생성되는 즉시 branching 노드에 대한 정보를 상위 branching 노드에서 알 수 있도록 하였다. MFT 필드에는 이러한 JOIN 메시지의 정보를 이용하여 기존의 수신자 리스트 이외에 해당 수신자와 연관된 하위 branching 노드에 대한 정보를 리스트로 연결하였다. 즉 branching 노드로 들어온 멀티캐스트 패킷은 수신자와 관련된 branching 노드에 대한 정보를 검색함으로써 다음 branching 노드까지 가는 도중의 라우터에서 불필요한 테이블 검색을 하지 않아도 된다. 이렇게 함으로써 유니캐스트 패킷들은 생성된 터널로 들어올 수 없기 때문에 불필요한 테이블 검색 없이 바로 IP Lookup을 통해 라우터를 빠져나간다.

3.2 트리 구성 방식

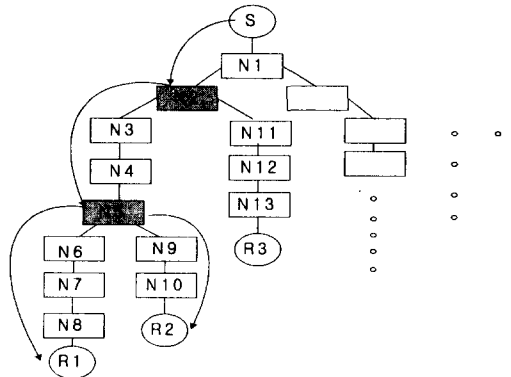
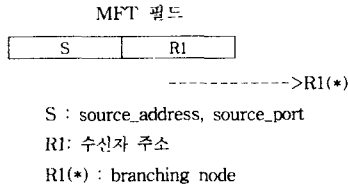


그림 3-1 멀티캐스트 라우팅 트리

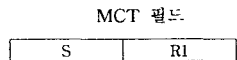
그림 [3-1]을 바탕으로 변경된 알고리즘의 예는 다음과 같다.  
색이 칠해져 있는 N2 와 N5가 branching node 이다.

**A. R1 이 JOIN (initial)**

R1이 그룹에 가입하려고 하면 다음과 같은 JOIN 메시지를 S를 향해서 전송하게 된다. JOIN 메시지 구성은 다음과 같다.  
(소스에 대한 정보 (address, port), 자신에 대한 정보 (address), branching 노드에 대한 정보 (초기에는 자신의 주소가 정보가 된다.))  
JOIN메시지는 S로 거슬러 올라가면서 MFT->MCT를 확인하면서 기존의 group이 존재하는가를 확인한다. S에서 JOIN 메시지를 받으면 MFT 에 JOIN 메시지의 정보를 저장한다.

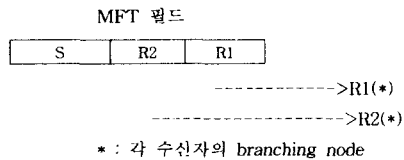


S는 이 MFT 필드를 이용해서 데이터 패킷과 TREE 패킷을 전송한다. 데이터 패킷의 경우 branching이 R1이기 때문에 R1까지 MFT 필드의 검색 작업 없이 바로 R1으로 전송된다. TREE 메시지의 경우 R1까지 지나가면서 REUNITE를 지원하는 라우터를 만나면 라우터의 MCT 필드에 해당 정보를 기록한다. (이때 라우터는 자신이 multicast 그룹의 일 부분임을 알게 된다. 그림 [3-1]에서는 1,2,3,4,5,6,7,8번의 라우터)

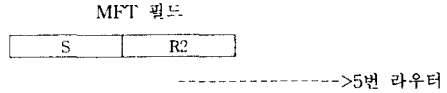


**B. R2에서 group에 가입**

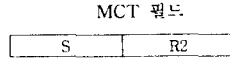
마찬가지로 R2에서 해당 정보를 가진 JOIN 메시지를 S를 향해서 전달한다. JOIN 메시지는 5번 라우터에서 MCT 필드를 검색함으로써 기존의 group이 있다는 것을 알게 되고, MFT 필드에 자신의 정보 역시 기록하고, 기존의 MCT 필드를 삭제한 후 그 정보를 MFT 필드로 복사한다.



R2에서 전달된 JOIN 메시지는 기존의 방식과 달리 intercept되지 않고 다음의 정보를 가지고 S로 간다.  
(소스의 정보(주소, 포트), 자신의 정보(주소), branching 노드의 정보(5번))  
반면 R1에서 전달된 JOIN 메시지는 intercept되어 S로 보내지지 않는다. S가 R2에서 전송된 JOIN 메시지를 받으면 자신의 MFT 필드값을 수정한다.



S는 MFT 필드 정보를 이용 5번 라우터까지 중간 라우터에서의 MFT 확인작업 없이 데이터 패킷을 전달한다. TREE 메시지는 R2로 향해 가면서 MCT 필드를 바꾼다.



5번 라우터에서 데이터 패킷을 받으면 MFT 필드를 이용해서 패킷을 복사하여 R1으로 바로 전달하고 Original 패킷 역시 R2로 전달한다.  
이 후의 과정은 B의 과정을 반복하여 그룹을 생성하게 된다.

**4.결론 및 연구 방향**

본 논문에서는 REUNITE의 장점을 살리면서 불필요한 테이블 검색을 최대한 줄이는 방법을 제안하였다. Branching 노드 간에 다이내믹하게 터널을 만들어 멀티캐스트 패킷의 경우 각 branching 노드에서만 MFT 필드를 검색하도록 하였다. REUNITE를 지원하는 라우터에서는 자신을 지나가는 모든 패킷들에 대한 MFT 테이블 검색할 필요없이 자신을 목적지로 한 패킷들만 MFT 검색을 해 주면 된다.

향후에는 제안된 방식의 성능분석과 그룹 탈퇴 시에도 터널을 이용할 수 있는 방법을 고안할 예정이다.

**5.참고문헌**

- [1] Ion Stoica, T.S. Eugene Ng, Hui Zhang, "REUNITE: A Recursive Unicast Approach to Multicast", in Proceedings of INFOCOM 00, Mar 2000
- [2] J. Tian and G. Neufeld, "Forwarding state reduction for sparse mode multicast communication", in Proceedings of INFOCOM'98, Mar.1998.
- [3] Chuck Semeria, Tom Maufer, "Introduction to IP Multicast Routing",Internet-Draft, July 1997.
- [4] M. Handley, "Session directories and internet multicast address allocation", in Proceedings of ACM SIGCOMM'98, Sept.1998
- [5] R. Perlamn, C. Lee, T. Ballardie, J. Crowcroft, Z. Wang, T. Maufer, C. Diot, J. Thoo, and M. Green, "Simple multicast: A design for simple, low-overhead multicast", Internet Draft, Internet Engineering Task Force, Mar.1999.