

# 효율적인 호스트 기반 멀티캐스트 트리 알고리즘을 적용한 호스트 기반 멀티캐스트 설계

이승철, 전진한, 광용완, 남지승  
전남대학교 정보통신협동과정  
e-mail :leesch@mdclab.chonnam.ac.kr

## Design of Host-base Multicast with HMTA

Seung-Cheol Lee, Yong-Wan Kwang, Jin-Han Jun, Ji-Seung Nam  
Interdisciplinary Program Of Information And Telecommunication, Chonnam  
National University

### 요 약

전통적인 멀티캐스트는 유니캐스트 방식에 비해 네트워크자원 이용 및 전송 측면에서 매우 효율적이지만 라우터에서 각 멀티캐스트 그룹의 상태 유지, 복잡한 네트워크 운영, 멀티캐스트를 위한 준비된 IP 주소 공간의 부족, 네트워크 변경의 제한 등의 문제점이 제기되어 왔다. 본 연구에서는 전통적인 멀티캐스트의 문제점을 최소화하기 위하여 효율적인 호스트 기반의 멀티캐스트 트리 알고리즘을 적용하여 호스트 기반 멀티캐스트 시스템을 설계 한다.

### 1. 서론

네트워크의 보급 확대와 성능 향상으로 네트워크 상에서 여러 사용자들 간에 데이터 교환이 이루어지는 다중 사용자 응용 프로그램들이 주목을 받고 있다. 다자간 데이터 교환을 지원하기 위한 대표적인 기술로 IP 멀티캐스트가 고려되어 왔으나, 실제 인터넷상에서 보편적인 서비스로서 사용되지 못하고 있다.

전통적인 멀티캐스트 프로토콜(IP 멀티캐스트)은 멀티캐스트를 지원하는 라우터에 필요한 모든 기능을 집중시키는 방법을 기반으로 개발되었다. 즉 라우터에서 각 멀티캐스트 그룹들에 대한 상태를 관리하고 임의의 Source 에서 임의의 다른 그룹으로 자료전송을 허용 함으로써 유니캐스트 방식에 비해 네트워크 자원 이용 및 전송 측면에서 매우 효율적으로 사용된다. 하지만 라우터 설계 시 기본 원칙인 Stateless 를 위반하고, 네트워크 확장 시 중대한 제한이 되며 라우터 관리의 복잡성을 증가시킨다. 또한 악의적인 침해에 의해 네트워크 취약성이 증가하며, 네트워크 관리 및

환경설정의 복잡성이 증가하게 된다.[3, 4, 5]

이러한 문제점을 최소화하기 위해 호스트 기반 멀티캐스트 프로토콜을 이용하는 방안이 대두되고 있다. 호스트 기반 멀티캐스트는 기존의 IP 멀티캐스트에서 라우터가 수행하던 기능을 호스트가 대신하게 함으로써 IP 멀티캐스트 환경에서 발생하는 제한을 없앤다. 하지만 호스트 기반 멀티캐스트는 라우터의 기능을 호스트에서 대신하기 때문에 IP 멀티캐스트에 비해 자료의 전송횟수가 증가한다.[2]

본 논문에서는 IP 멀티캐스트와 호스트 기반 멀티캐스트의 문제점을 최소화하기 위하여 호스트 기반 멀티캐스트 시스템을 설계한다.

### 2. 관련 연구

#### 2.1 호스트 기반 멀티캐스트

호스트 기반 멀티캐스트는 자료를 전송하기 위해 구성된 라우팅 트리에 의해 송신자가 자료를 전송하고 자료를 받은 수신자가 라우팅 트리에 의해 다음 수신자에게 자료를 전송하게 된다.[그림 1] 즉 IP 멀티캐스트에서 라우터가 하는 기능을 호스트에서 대신 함으로써 네트워크 운영이나 변경 시 IP 멀티캐스트

\* 본연구는 한국과학재단지정 전남대학교 고품질 전기전자부품 및 시스템연구센터의 연구비 지원에 의해 연구되었음

환경에서 발생하는 제한을 없애고 에리제어, 흐름제어, 혼잡제어 등의 기능을 간단히 적용 할 수 있다.[1]

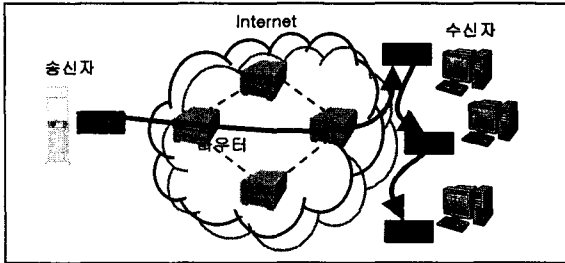


그림 1. 호스트 기반 멀티캐스트

### 3. 제안된 호스트 기반 멀티캐스트

#### 3.1 구성요소

제안하는 호스트 기반 멀티캐스트는 MCA (Multicast Control Agent)와 MRA(Multicast Routing Agent)로 구성된다.

MCA 는 멀티캐스트 그룹, MCA 와 MRA 간의 컨트롤 정보 및 트리 구조에 대한 정보를 관리한다. 즉 MRA 의 Join, Leave, Create, Rejoin, Failure 등의 컨트롤 데이터를 관리하며, 각 MRA 에 관한 라우팅 정보 및 MRA 들의 최적의 트리를 관리 한다.

MRA 는 이웃 노드들에 대한 라우팅 정보를 가진다. 실제적으로 각각의 호스트는 MRA 가 되며 MRA 는 데이터를 수신 받기 위해 처음 MCA 로 접속하여 멀티캐스트 그룹에 대한 상태정보를 받은 다음 그 상태정보를 통해서 자신의 이웃한 노드들을 정하고 정해진 이웃노드들로부터 데이터를 수신하거나 이웃한 노드들에게 데이터를 송신하게 된다.[그림 2]

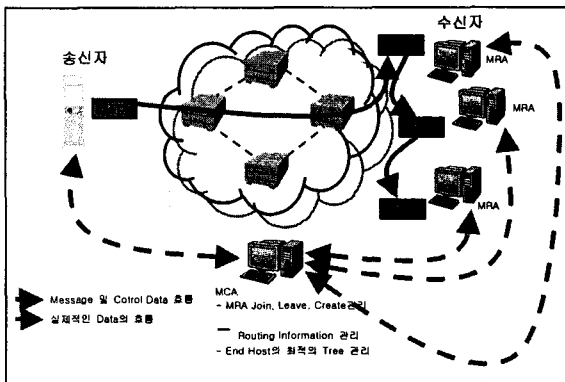


그림 2. 제안된 호스트 기반 멀티캐스트

#### 3.2 호스트 기반 멀티캐스트 트리 구성 알고리즘 (HMTA: Host-base Multicast Tree Algorithm)

1. Sub\_Root가 1개 이상의 자손을 가지면 모든 자손 노드들의 정보(Root 와의 Cost)를 변경
2. End /\* End of Procedure \*/

```

Procedure Tree_Build(Root, New_Node)
1. Root 와 New_node가 동일 네트워크에 존재하지 않으면
   Call Procedure Tree_adjust(Root, New_node)
   Return; /* End of Procedure */
2. New_node = Peer(Root)
3. Call Procedure Renew_Tree(New_Node)
4. End /* End of Procedure */
    
```

```

Procedure Tree_Adjust(Sub_Root, New_Node)
1. Sub_Root 의 모든 Children 노드들과 New_Node가 동일 네트워크에 존재하는지 비교
2. 동일 네트워크에 존재하지 않으면 go to 5
3. New_Node = Peer(Children(Sub_Root)[i])
4. Call Procedure Renew_Tree(New_node)
5. Return /* End of Procedure */
6. If available_degree(Sub_Root) > Current_Degree(Sub_Root) then
7.   Sub_root = Parent(New_Node)
8.   Call Procedure Renew_Tree(New_Node)
9. Return /* End of Procedure */
10. New_Node_RTT = Measured_RTT(Sub_Root, New_Node)
11. Max_Cost = Sub_Root와 그 Children들과의 최대 RTT
12. Switch_Node = Sub_Root와 최대 RTT를 가진 Sub_Root의 Child
13. If Max_node > New_node_RTT then
14.   Sub_Root = Parent(New_node)
15.   Call Procedure Tree_Renew(New_node)
16.   Call Procedure Tree_Adjust(Sub_Root, Switch_Node)
17. Return /* End of Procedure */
18. M_RTT[i] = Sub_Root의 Children들과 New_Node와의 RTT 측정
19. New_Sub_Root = Minimum(M_RTT[i]+ Sub_Root의 Children의 Cost)을 만족하는 Sub_Root의 Child
20. Call Tree_Adjust(New_Sub_Root, New_Node)
21. End /* End of Procedure */
    
```

Procedure Renew\_Tree(Sub\_Root)

제안된 트리 구성 알고리즘은 트리의 전체 Depth 를

최소화하여 데이터 전송 시 발생하는 코스트를 최소화 한다. 이를 위해 루트와의 코스트(RTT)가 작은 MRA 가 전체 트리의 상위 Level 에 위치하며 루트와의 코스트가 큰 MRA 는 다른 MRA 가 Join 함에 따라 트리의 하위 Level 로 옮겨진다.(Tree\_Adjust 라인 10~17 참조) 트리 알고리즘의 성능은 새로운 멤버의 join 시  $O(\log_2 n)$ 이다. 기존의 MRA 와 동일한 네트워크에 존재하는 MRA 가 Join 할 경우에는 기존 MRA 의 Peer 로 처리하여 데이터 전송 시 LAN 의 Broadcasting 기능을 사용하며, 이를 통해 데이터 전송 성능 및 트리의 성능을 향상하였다.

4. 호스트 기반 멀티캐스트 시스템의 구조

본 논문에서 기술하고자 하는 제안된 호스트 기반 멀티캐스트 시스템의 전체 구성도는 [그림 3]과 같다. MCA 는 메시지 관리자, 트리구성 관리자, 라우팅 상태정보 관리자로 구성된다. 메시지 관리자는 메시지 송수신 모듈을 가지고있다. 메시지 송수신 모듈에서는 시스템의 컨트롤데이터를 송신하고 수신한다. 또한 라우팅 상태정보를 송수신한다. 트리 구성 관리자는 메시지 관리자에 의해 수신된 데이터를 가지고 각 MRA 의 라우팅 트리를 구성한다. 라우팅 상태정보 관리자는 트리 구성 관리자에 의해 구성된 라우팅 트리들을 저장하고 관리한다.

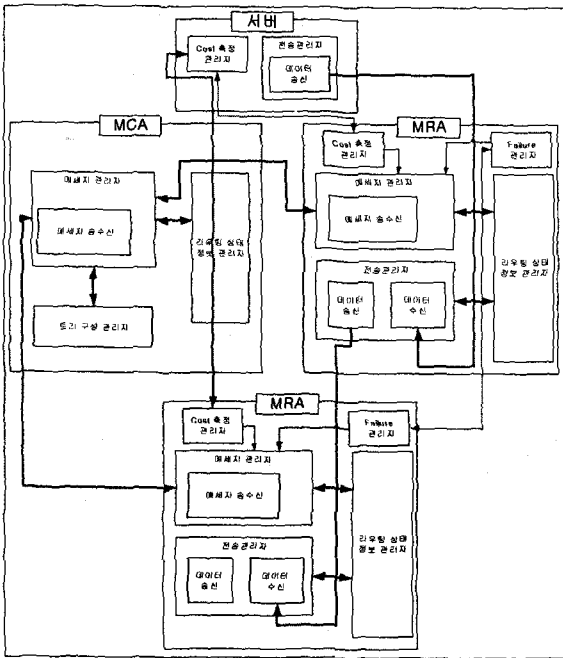


그림 3. 제안된 호스트 기반 멀티캐스트 시스템구성도

MRA 는 Cost 측정 관리자, 메시지 관리자, 라우팅 상태정보 관리자, 전송관리자, Failure 관리자로 구성 된다. Cost 측정 관리자에서는 서버와의 RTT 를 계산 하여 메시지 관리자에게 전달한다. 메시지 관리자는 메시지 송수신 모듈이있다. 메시지 송수신 모듈에서는

시스템의 컨트롤 데이터들과 Cost 를 송수신 하고, 라우팅 상태정보를 송수신 한다. Failure 관리자는 이웃한 노드들이 정상적으로 동작하는지 일정시간마다 확인 을 한다. 라우팅 상태 정보관리자는 MCA 에서 보내진 자신의 이웃한 노드들에 대한 상태정보를 저장하고 관리한다. 전송관리자는 데이터 송신 모듈과 데이터 수신 모듈로 구성된다. 전송관리자는 라우팅 상태정보 관리자에서 이웃한 노드들에 대한 정보를 가지고 데이터를 송수신 한다.

5. 제안된 호스트 기반 멀티캐스트 시스템 동작과정

MRA 는 먼저 Cost 를 결정하기 위하여 서버와의 RTT 를 측정한다. 측정된 RTT 값은 컨트롤 데이터와 함께 MCA 로 전송된다. MCA 는 컨트롤 데이터와 RTT 측정 값을 수신하고 컨트롤 데이터의 상태여부에 따라 호스트 기반 멀티캐스트 트리 알고리즘으로부터 멀티캐스트 라우팅 트리를 결정하고 접속된 모든 MRA 에 대한 라우팅 상태정보를 저장하고 관리한다. 그리고 관련된 MRA 에 대한 라우팅 상태정보를 MRA 에게 전송한다. MRA 는 MCA 에서 전송된 라우팅 상태정보를 수신, 저장하고 관리한다. 그리고 라우팅 상태정보에 따라 이웃한 노드들에 대해 데이터를 송수신 한다.

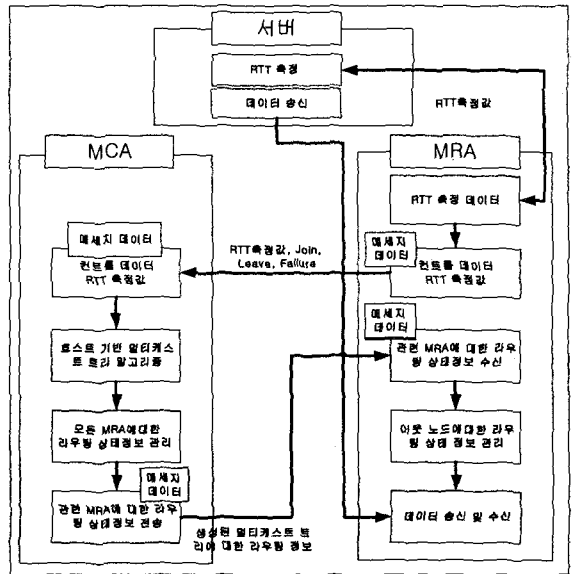


그림 4. 호스트 기반 멀티캐스트 동작 과정

5.1 Join 동작

멀티캐스트 그룹에 Join 하고자 하는 멤버는 MCA 에 Join 메시지를 보낸다. MCA 는 Join 메시지를 받은 후 새로운 멤버가 속한 그룹의 루트에 관한 정보를 보내고 동시에 Tree\_Build(Root, New\_node) 함수를 호출하여 트리 구성 알고리즘을 수행한다. 트리 구성 알고리즘이 수행이 완료되면 MCA 는 새로운 멤버에 인접 MRA 에 대한 정보를 전송하며, 새로운 멤버는 이

를 기초로 하여 데이터의 송/수신을 수행한다.

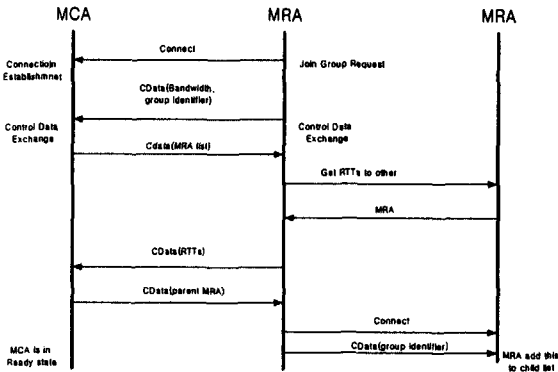


그림 5. Join Group

5.2 MRA의 Leave 동작

멀티캐스트 그룹을 Leave 하고자 하는 멤버는 MCA에 Leave 메시지를 보낸다. MCA는 Leave 메시지를 받은 후 Leave를 요청한 멤버가 트리의 Leaf 노드이면 새로운 그 노드를 트리에서 제거한다. 만약 요청 노드가 Leaf 노드가 아닐 경우, 그 노드의 각각의 Children들에게 Rejoin 요청 메시지를 전송한다. Rejoin 요청 메시지를 받은 노드들은 MCA에 Join 요청을 전송하며, MCA는 새로운 멀티캐스트 트리를 구성한다.

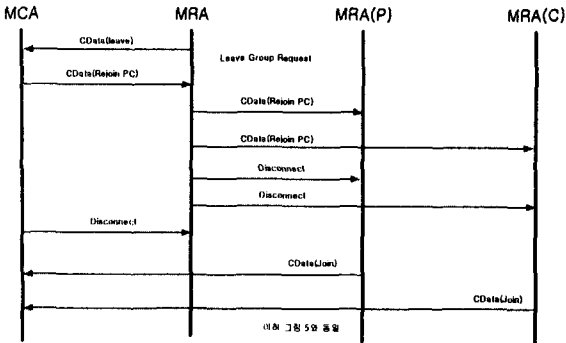


그림 6. Leave Group (Leaf 노드가 아닐 때)

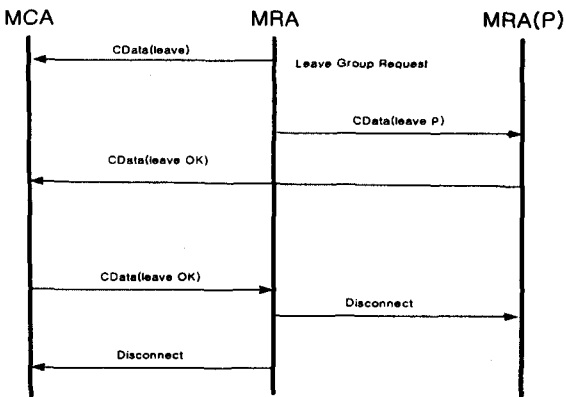


그림 7. Leave Group (Leaf 노드 일 때)

5.3 Failure Detection 동작

인접 노드들과의 Failure가 발생할 경우, 이 Failure로 인해 기존의 멀티캐스트 트리에서 분리된 서브 트리의 루트가 MCA에 Join 요청을 한다. MCA는 Join 요청을 받은 후 새로운 멀티캐스트 트리를 구성한다.

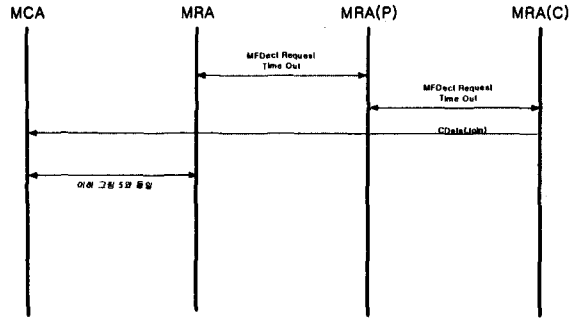


그림 7. Failure Detection

6. 결론 및 향후 연구과제

본 논문에서는 호스트 기반 멀티캐스트 트리 알고리즘을 설명하고 이를 적용하여 호스트 기반 멀티캐스트 시스템을 설계하였다. 호스트 기반 멀티캐스트 시스템을 효율적인 호스트 기반 멀티캐스트 알고리즘을 적용하였기 때문에 기존 멀티캐스트의 단점인 네트워크 관리 및 환경설정의 복잡성을 해결하였다. 또한 호스트 기반 멀티캐스트에서의 호스트가 라우터 역할을 함으로서 호스트에 가중되는 부하를 줄였다.

향후 연구과제로는 더욱더 안정적이고 효율적인 서비스를 제공하기 위해서 에러 제어, 흐름 제어, 혼잡 제어를 적용시킬수 있는 다양한 연구가 진행 되어야 할 것이다.

참고문헌

- [1] Y. Chu, S.G. Rao, and H. Zhang, "A case for end system multicast," Proceeding of ACM SIGMETRICS, pages 1-12, June 2000. Carnegie Mellon University
- [2] B. Zhang, S. Jamin, L. Zhang, "Host Multicast: Framework for Delivering Multicast To End Users," Computer Science Dept., University of California, Los Angeles, CA90095. 2002 IEEE
- [3] H.W. Holbrook and D.R. Cheriton. IP multicast channels: EXPRESS support for large-scale single-source
- [4] R. Perlman, 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, March 1999. Work in progress.
- [5] I. Stoica, T.S.E. Ng, and H. Zhang. REUNITE: A recursive unicast approach to multicast. In Proceedings of IEEE INFOCOM '00, Tel-Aviv, Israel, March 2000.