

# 개인 방송 시스템을 위한 mOBCP 기반의 오버레이 멀티캐스트 트리 구성 방안

정회원 남 지 승\*, 준회원 강 미 영\*, 전 진 한\*, 손 승 철\*

## Personal Broadcasting System Using mOBCP-based Overlay Multicast Tree Construction Method

Ji-Seung Nam\* *Regular Member*,  
Mi-Young Kang\*, Jin-Han Jeon\*, Seung-Chul Son\* *Associate Members*

### 요 약

개인 방송 서비스를 다수의 사용자들에게 제공하기 위해서 본 논문에서는 오버레이 멀티캐스트 기반의 알고리즘을 적용한 효율적이고 향상된 성능을 제공하는 miniOverlay Broadcasting Control Protocol(mOBCP)을 제안하였다. mOBCP는 서비스를 받고자 대기하는 자식 노드들에게 대기시간을 최소한으로 줄이면서 가능한 빠르게 부모 노드를 찾을 수 있는 알고리즘과 서비스를 받던 기존 부모 노드의 경로에 Failure가 발생 되었을 경우 새로운 부모를 찾아서 빠르게 다시 서비스를 가능하게 할 수 있는 성능 효율적인 Mechanism을 본 논문에서는 제안한다.

성능비교는 제한된 트리 구성 시간 변화에 대한 지연(Latency)을 시뮬레이션을 통하여 결과를 얻어내었으며 결과를 통해 제안된 기법의 효율성을 보여주고 있다.

**Key Words** : Overlay Multicast, PPL, mOBCP, PBS, Join session

### ABSTRACT

For better performance and to avoid member service annoyance that results due to joining-clients' waiting durations and time-outs when there are more than one client wanting to join concurrently for Personal Broadcasting System service, there is a need for improving concurrent member joining mechanism.

For a more efficient and better performing, this paper apply Overlay Multicast based mini-Overlay Broadcasting Control Protocol (mOBCP) Algorithm on Personal Broadcasting System. mOBCP proposed is performance-effective mechanism, since it considers the case of how fast will children, concurrently, find and join new parents when paths to existing parents are in Failure.

The performance comparison, in terms of tree construction time variation and Latency are done through simulations and the results conclude in favour of the Proposed mOBCP.

### I. 서 론

전통적인 TBCP(Tree Building Control Protocol)

메커니즘을 적용하여 다수의 사용자들에게 개인 방송 서비스를 제공할 경우 다수 사용자들의 동시적 Join 세션 부분과 서비스 노드 경로에서 발생하는

※ 본 논문은 2006년도 전남대학교 연구년 교수연구비 지원에 의하여 연구되었음(This study was financially supported by Chonnam National University).

\* 전남대학교 컴퓨터공학과 멀티미디어정보통신 연구실 (jsnam@chonnam.ac.kr)

논문번호 : KICS2007-06-269, 접수일자 : 2007년 6월 12일, 최종논문접수일자 : 2007년 7월 20일

다양한 Failure 처리 시 사용자들은 불만족스러운 서비스를 받게 된다.

전통적인 TBCP는 사용자가 Join 요구를 하였을 때, 부모 노드가 선택되어지고 서비스를 받을 수 있는 상태가 될 때까지 다른 사용자들의 Join 처리를 대기시키는 즉 Blocking 상태로 들어가게 된다. 따라서 TBCP 메커니즘을 적용하여 다수 사용자가 거의 동시에 개인 방송 서비스를 받기 위해 Join 을 요구하였을 때 대기하는 시간이 길어짐에 따라 최악의 상태의 경우 Join 세션 연결이 끊어지는 결과를 초래하게 된다.

물론 개인 방송 서비스를 하는데 있어 적용할 수 있는 기법으로서 전통적인 TBCP 기법 뿐 만이 아니라 발표된 다양한 기법들에 대해 연구하고 개인 방송 시스템에 적용해 봄으로써 실제 운용하는데 있어서의 장단점을 비교하였다. 하지만 본 연구에서 개인방송 서비스를 하는데 있어서 우선적으로 고려한 부분인 사용자 대기 시간을 최소한으로 줄이면서 가능한 빠르게 서비스 해 줄 수 있는 부모 노드를 찾는 최적의 알고리즘을 적용하는데 어려움을 겪었다.

본 논문에서 제안하는 mOBCP 기반 오버레이 멀티캐스트 기법은 사용자 Join 대기 시간(Joining-Time-Out)과 빠른 시간 안에 최적의 부모 노드를 찾는 문제를 해결하는데 중점을 두었다.

개인방송 시스템은 서버모듈을 가진 개인방송국(Personal Broadcasting Station)으로 구성되어있으며 서버 모듈인 mOBCP는 New 노드 Join 요청에 대한 응답, 탈퇴, Failure, Broken Paths 그리고 세션에서의 빠른 Join과 Rejoin 등 사용자 관리를 원활하게 할 수 있도록 하는 업무를 담당한다.

또한 mOBCP 메커니즘에서는 개인 방송을 서비스 하고자 하는 노드가 소스 루트(source-root)가 되며, 소스 루트는 서비스가 가능한 상태인 부모 노드 리스트인 PPL(Potential Parents List)을 관리하고 업데이트 정보를 유지하는 업무를 관장한다.

서비스를 받고자 Join하는 New 노드들은 각각 개인방송국에 B\_REQs(Broadcasting Requests) 메시지를 보낸다. Join 요청을 받은 개인방송국은 Event 를 처리하기 위한 서버모듈을 가동시킴으로써, New 노드들과의 세션 설정을 하게 된다.

본 논문의 구성은 1.1에서 제안한 mOBCP와 관련된 연구 논문에 대해 살펴보고 2.1에서는 오버레이 방송 시스템에 대해 알아보고 2.2에서는 제안하는 오버레이 방송 시스템에서의 세션 기술에 대해

서 설명한다. 2.3에서 사용자들의 세션 Join 처리 과정에 대해 자세히 살펴보고 2.4, 2.5에서는 제안하는 mOBCP 기법에 대해 자세히 설명한다. 2.6에서는 시뮬레이션 환경과 결과에 대해 설명하고 III장에서는 결론 및 향후 연구 방향에 관하여 기술한다.

## 1.1 관련 연구

Mathy, Canonico, 그리고 Hutchison은 전통적인 TBCP에 대한 기초를 확립하였으며 분산형 알고리즘 중 Tree-First 기반 특성으로 분류 할 수 있으며 가능한 초기에 가장 좋은 트리를 만드는데 있어서 수렴 시간(convergence time)을 줄이는데 주안점을 두었다<sup>[1]</sup>.

오버레이 기반 멀티캐스트에서 노드는 자료의 수신 및 전송기능 뿐만 아니라 기존 IP 멀티캐스트에서 라우터가 수행하던 멀티캐스트 그룹 내의 자료 전송을 위한 트리의 구성/관리 기능을 수행한다. 따라서 IP 멀티캐스트와 달리 현재 설치된 네트워크의 구성뿐 아니라 네트워크의 확장이나 변경 시에도 영향을 미치지 않는다. 게다가 네트워크 계층의 상위계층에서 제공하는 오류제어, 혼잡제어 및 흐름 제어 등의 기능을 IP 멀티캐스트에 비해 용이하게 제공할 수 있다<sup>[2]</sup>. 그러나 오버레이 기반 멀티캐스트는 자료의 전송 시 각각의 노드들이 물리적인 네트워크의 구성을 파악할 수 없기 때문에 IP 멀티캐스트에 비해 네트워크 회선의 중복 이용과 전송을 담당하는 노드에서 발생하는 오버헤드와 같은 성능 감소 요인이 필연적으로 존재하게 된다.

Farinacci 등은 reverse path 개념을 기반으로 한 컨트롤 트리를 구성하는 기법을 제안하고 발표하였다<sup>[3]</sup>. 그러나 제안되어진 컨트롤 트리는 라우터 정보에 의존함에 따라 오버레이 트리 구성 개념과는 맞지가 않는다.

Pendarakis 등은 트리 확장성의 한계에 대해 주안점을 두고 설명하고 있다<sup>[4]</sup>. 즉, 분산되어진 토폴로지의 스패닝 트리(spanning tree)를 만들고, 세션 컨트롤러를 유지하고, 멤버들의 연결 정보를 유지해야 하므로 그룹 멤버 쉽들의 전체 정보를 알고 있어야 하며 메쉬 토폴로지의 정보를 알고 있어야만 한다. 따라서 분산형 알고리즘 중 Mesh 기반 특성을 가지고 있으며, 또한 Francis 등이 토의한 Yoid<sup>[5]</sup> 그리고 각 노드들이 그룹 멤버쉽들간의 정보를 유지하기 위해 전체 정보를 알고 있는 유명한 Narada 프로토콜도 분산형 알고리즘 중 Mesh 기반 특성으로 분류할 수 있다.

이렇게 모든 프로토콜들은 트리를 구성하기 위한 컨트롤에 대해 계속 연구되어왔으며 앞으로도 계속 될 것이다. 그러나 개인방송시스템에 적합한 프로토콜로는 상당한 한계를 가지고 있었다. 본 논문에서 제안한 모델의 트리 구성 알고리즘은 다양한 메커니즘들 중에서 분산형 알고리즘 중 Tree-First 기반 특성을 가지고 있으며 성능비교는 Tree-First 기반 특성을 가지고 있는 전통적인 TBCP와 HMTP와의 비교를 통해 성능을 측정하였다.

## II. 본론

### 2.1 오버레이 개인 방송 시스템

오버레이 멀티캐스트 개인 방송 시스템은 3가지 주요 기술(오버레이 기술, 멀티캐스트 기술, 개인방송 기술)을 이용함으로써 여러 가지 한계를 극복하도록 구현 되어졌다. 그림 1에서는 오버레이 방송 시스템의 프레임워크를 보여주고 있다.

그림 1에서 오버레이 개인 방송 시스템은 기본적으로 개인 방송 시스템 그리고 mOBCP기반 오버레이 멀티캐스팅 모듈로 구성되어 있다.

New 노드가 방송을 보기 위해 Join요청을 하면 mOBCP 오버레이 멀티캐스트 모듈은 New 노드에 요청에 대한 응답을 하고 New 노드가 최적의 부모 노드를 선택했을 경우 트리를 생성하고 그룹을 컨트롤하고 관리하는 담당을 하게 된다. mOBCP 오버레이 멀티캐스트 모듈과 개인방송국은 서비스 개시를 위해 서로 통신을 하며 mOBCP 오버레이 멀티캐스트 모듈은 New 노드들의 세션을 위한 Join을 처리할 수 있으며, New 노드들이 최적의 부모 노드를 빠른 시간 안에 선택할 수 있게 PPL(Potential Parent List)이라 불리는 정보를 관리하고 있다. New 노드들은 세션 Join 요구를 하였을 때 mOBCP 모듈로부터 응답 받을 시 PPL 정보

를 함께 받으며 PPL 정보를 받은 New 노드들은 PP(Potential Parent)를 각각 RTT 체크루틴을 수행함으로써 적절한 부모를 선택하게 된다.

### 2.2 오버레이 방송 시스템의 세션 구조

오버레이 개인 방송 시스템에서 세션에 대한 구조는 그림 2에서 잘 보여주고 있다.

만약 New 노드가 방송 세션에 참가하려고 하면 먼저 New\_B\_REQ(New Client Broadcast Request) 메시지를 PBS(Personal Broadcast Station)에 전송한다. 그러면 mOBCP기반 오버레이 멀티캐스트 트리 관리 모듈은 PBS에 속한 모듈로서 New\_B\_REQ를 받은 PBS는 mOBCP 모듈을 가동하여 New 노드에게 서비스가 가능한 부모 노드 리스트인 PPL을 응답 메시지와 함께 보내게 된다. PPL 정보를 받은 New 노드는 RTT 체크 루틴을 수행하여 최적의 부모노드를 선택함으로써 성공적으로 Join하게 된다.

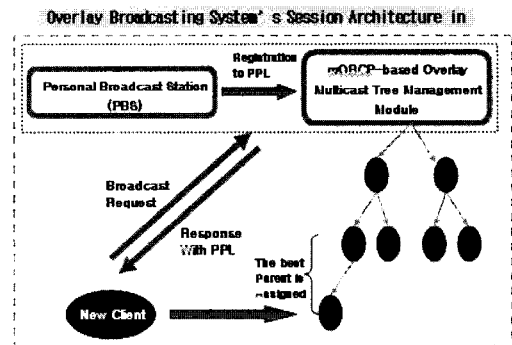


그림 2. 오버레이 방송 시스템의 세션 구조

New 노드는 선택한 부모 노드에 성공적으로 Join 했다는 메시지를 PBS에 보냄으로써 개인방송국의 서버 모듈인 mOBCP기반 오버레이 멀티캐스트 트리 관리 모듈은 PPL 정보를 업데이트하며 항상 최신의 정보를 유지한다.

### 2.3 TBCP의 한계

전통적인 TBCP에서 New 노드가 세션에 Join하기를 원하였을 때 TBCP 수행을 위해 소스루트의 주소와 포트 정보를 이용하여 접근한다. 자식 노드의 수는 소스루트에 의해 조절될 수 있으나 고정되어 있다. 이와 같은 소스루트의 컨트롤은 TBCP 트리에 많은 트래픽 로드를 가져온다. 그림 3은 전통적인 TBCP의 개인방송 시스템에서의 한계를 설명하고 있다.

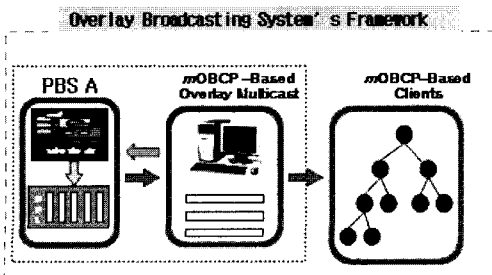


그림 1. 오버레이 개인 방송 시스템 프레임 워크

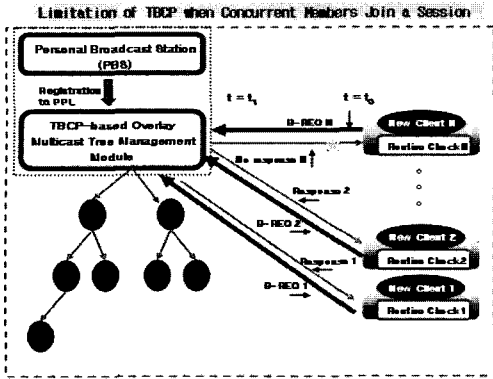


그림 3. 세션 가입시 TBCP 의 한계

오버레이 개인 방송 시스템에서 동일 시간대에 개인방송 서비스 세션에 참가하고자 하는 New 노드가 한 명 이상이었을 때 TBCP에서 수행 되어지는 Join 메커니즘에서 각각의 New 노드는 먼저 Join 요청했던 New 노드가 새로운 최적의 부모노드를 찾을 때까지 대기상태에 들어가게 된다. 그림 3은 TBCP기반 오버레이 기법에 의해 컨트롤 되어 트리가 생성되는 구조로서 N은 t = t0 시간에 Join하기를 원하는 New 노드이며 동일 시간대에 즉, 동시에 B\_REQs 메시지를 PBS에 보낼 수 있다. 그러나 첫 번째 New 노드의 B\_REQ를 처리하는 동안 TBCP 프로세스는 첫 번째 New 노드(t1 - t0)를 처리하는 동안 다른 New 노드들의 B\_REQs 메시지를 받지 않는다. 이와 같이 동일 시간대에 다수의 New 노드들이 Join하기를 원하더라도 알고리즘 특성상 순서적으로 대기하여 처리되어지는 다음과 같은 수식을 얻게 된다.

$$W_i = (i - 1)T \tag{식 1}$$

i는 동일 시간대에 Join 요청을 하는 New 노드들이며 T는 각각의 New 노드들이 TBCP 알고리즘을 수행하면서 연계 되는 처리시간 (Join 요청시간, PPL 응답시간, RTT 체크 수행 시간, 최적의 부모노드 선택시간)이며, Wi는 동일 시간대에 다수의 New 노드들이 Join 요청을 하였을 때 TBCP 알고리즘을 수행하여 첫 번째 New 노드부터 마지막으로 서비스를 받게 되는 New 노드까지 걸리는 New 노드들의 총 대기시간이다.

따라서 TBCP를 수행할 경우 빠른 Join 서비스에 참가 하지 못한 New 노드들은 서비스 참가에 불만족스러워 하거나 아예 서비스 연결을 끊어버릴 수 있다. 본 논문에서는 이러한 기존의 TBCP한계를

극복하기 위해 mOBCP 모듈에서는 동일 시간대에 New노드들이 동시에 Join요청을 할 경우 소스루트가 New 노드들 각각에게 PPL(Potential Parent List) 정보를 보내줌으로써 New노드들 각각은 즉시 최적의 부모 찾아 Join하는 기능을 수행 할 수 있으므로 Multicast Group의 노드 증가에 따른 Join 처리 시간의 증가를 줄일 수 있다. 따라서, 동일 시간대에 New 노드들의 Join요청이 풀리는현상이 발생할 경우 대기시간을 최소한으로 줄이면서 빠른 시간 안에 최적의 부모 노드를 선택할 수 있는 다음과 같은 식을 얻을 수 있다.

$$W_i = \text{Max } T_1, T_2, T_3, \dots, T_{i-1} \tag{식 2}$$

i는 동일 시간대에 Join 요청을 하는 New 노드들이며 T는 각각의 New 노드들이 제안하는 mOBCP알고리즘을 수행하면서 연계되는 처리시간 (Join 요청 시간, PPL 응답시간, RTT 체크 수행 시간, 최적의 부모노드 선택 시간)이며, Wi는 동일 시간대에 다수의 New 노드들이Join 요청을 하였을 때 mOBCP 알고리즘을 수행하여 첫 번째 New 노드부터 마지막으로 서비스를 받게되는 New 노드까지 걸리는 New 노드들의 총 대기시간으로서 본 알고리즘에서는 Join요청에 대해 거의 동시에 처리되는 특성을 가지고 있으므로 동일 시간대에 Join 요청을 한 New 노드들 중 가장 대기 시간이 많이 걸린 노드의 대기시간 안에 다른 New 노드들은 이미 서비스를 소스루트로부터 받고 있는 상태이다.

### 2.4 mOBCP 기법

제안한 mOBCP(mini-Overlay Broadcasting Control Protocol)기법은 New노드들의 대기시간 최소화화 최적의 부모노드를 가능한 빠르게 선택하여 만족스러운 서비스를 받을 수 있는 성능효율적인 트리를 구성하는 것이다. mOBCP기반 오버레이 멀티캐스트 모듈은 서비스가 가능한 부모노드 리스트인 PPL(Potential Parents List)의 정보를 관리하는 기능을 가지고 있으며 세션에 Join하기를 원하는 모든 New노드는 B\_REQ 메시지를 소스 루트에 보냄으로서 소스루트로부터 응답메시지와 함께 PPL 리스트를 받게 된다. PPL 정보를 받은 New노드는 최적의 부모 노드를 찾기 위해 PPL 정보를 이용하여 체크 루틴을 수행 하게 된다. 최적의 부모노드를 선택하게 된 New 노드는 소스루트에 서비스 가능한 자신의 여유 Outdegree를 등록하고 선택 되어진 부모노드의 여유Outdegree를 삭제하는 PPL의 일련의

정보를 업데이트하게 된다.

PPL의 업데이트 작업은 mOBBCP 모듈이 실행되는 동안 아주 중요한 수행작업이며, 이 작업은 모든 New노드들이 부모 노드를 선택하였을 때와 기존의 노드들이 경로가 끊어져서 다시 새로운 부모노드를 찾았을 때 반드시 수행되어야 할 루틴이다.

### 2.5 mOBBCP 구조

그림 4에서는 mOBBCP의 구조를 단순화 시킨 그림을 보여주고 있으며 mOBBCP모듈이 실행되는 동안 동시에 join하기를 원하는 New노드들의 수행 과정을 보여주고 있다.

①에서 방송서비스를 하고자 하는 소스루트는 먼저 자신의 Out-degree 상태를 서비스 가능한 부모노드 리스트 PPL(Potential Parent List)에 등록한다. 다음으로 New노드가 세션에 Join하기를 원할 경우 ②에서 New노드는 Join 요청을 소스 루트에 보낸다.

③에서 소스루트는 Join 요청을 보낸 New 노드에게 현재 가지고 있는 PPL 정보를 보냄으로써 응답하게 된다.

④에서 New 노드는 PPL 정보를 받고 최적의 부모 노드를 선택하기 위해 PPL에 등록되어진 입시 부모 노드와 자신 사이 각각의 RTT 상태를 체크하게 된다.

⑤에서 New 노드는 가장 최적의 부모 노드를 선택하게 되고, ⑥에서 자신의 여유 Out-degree 상태와 선택어진 부모노드의 변경된 Out-degree 상태 등 변경된 정보를 소스루트에 보냄으로써 PPL 정보를 최신의 것으로 업데이트 하게 된다.

만약 동일 시간대에 다수의 New 노드의 Join 요청을 받았을 때 TBCP에서 수행 되어지는 Join 메커니즘에서는 각각의 New 노드는 먼저 Join 요청했던 New 노드가 새로운 최적의 부모노드를 찾을 때까지 대기하였다가 순서적으로 처리되는 기법으로서 시간에 민감한 개인방송 시스템에는 적합하지 않다.

본 모델의 Join 메커니즘에서는 만약 동일 시간대에 다수의 New 노드의 Join 요청을 받았을 때

⑦과 같이 소스루트는 다수의 New 노드 Join 요청을 동시에 받아들인다.

⑧에서 소스루트는 Join 요청을 보낸 각각의 New 노드들에게 현재 가지고 있는 PPL 정보를 보냄으로써 응답하게 된다.

⑨에서 각각의 New 노드들은 PPL 정보를 받고

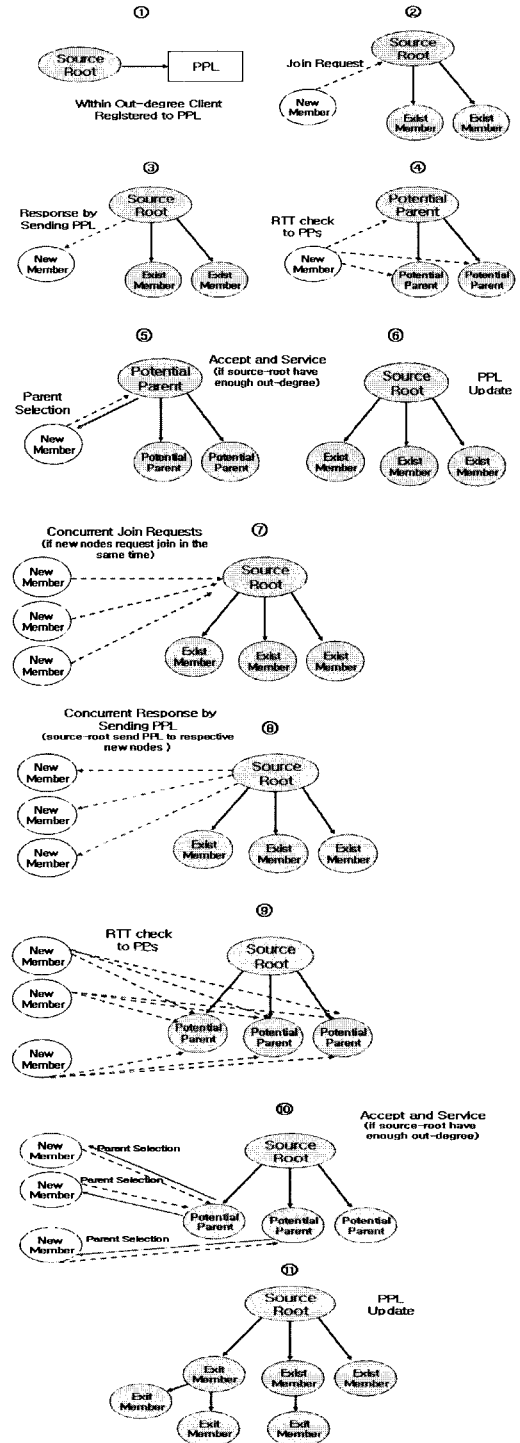


그림 4. mOBBCP 의 구조

최적의 부모 노드를 선택하기 위해 PPL에 등록되어진 입시 부모 노드와 자신 사이 각각의 RTT 체크를 수행하게 된다.

⑩에서 각각의 New 노드들은 가장 최적의 부모 노드를 선택하게 되고,

⑪에서 자신의 여유 Out-degree 상태와 선택 되어진 부모노드의 변경된 Out-degree 상태 등 변경된 정보를 소스루트에 보냄으로써 PPL 정보를 최신의 것으로 업데이트 하게 된다.

이와 같은 알고리즘 수행으로 동일 시간대에 New 노드들의 Join요청이 몰리는 현상이 발생할 경우 New 노드들의 대기시간을 최소한으로 줄이면서 빠른 시간 안에 최적의 부모 노드를 선택할 수 있다.

### 2.6 시뮬레이션 환경

시뮬레이션을 이용한 mOBCP(mini-Overlay Broadcasting Control Protocol) 알고리즘의 성능평가를 위하여 GT-ITM<sup>[9]</sup>의 Transit-Stub 모델을 적용하여 1000개의 노드들로 구성된 네트워크 토폴로지를 생성하였다. 구성된 네트워크 토폴로지를 기반으로 하여 트리를 구성할 수 있는 시간을 점점 증가시키면서 (15초, 75초, 150초) 256개의 노드가 거의 동시에 개인방송 서비스를 요청하였을 때 트리를 구성하는 평균 Latency를 제안한 mOBCP 그리고 전통적인 TBCP 와 HMTP에 대해서 트리를 생성하여 성능평가를 실시하였다.

### 2.7 결과 분석

본 논문에서는 세 가지 기법(mOBCP, TBCP, HMTP)의 트리를 생성하여 시뮬레이션 하였다. 그림 5, 그림 6, 그림 7은 256개의 New 노드가 동시에 Join 요청을 하여 트리를 구성하였을 때 트리 구성 시간을 제한함으로써 평균 트리 Latency의 결과를 분석하였다. 그림 5, 6, 7에서 첫 번째 New노드가 Join 요청을 소스루트에 한 후에 소스루트로부터 응답메시지를 받고 트리를 구성하는 일련의 알고리즘 루틴을 수행하는데 까지 걸리는 평균 Latency를 측정하였을 때의 트리 구성 시간으로서 약간의 차이를 보이면서 트리가 구성됨을 분석 할 수 있다. 시간이 증가할수록 차이를 보이고 있는데 이 중에서 TBCP의 성능이 비교적 좋지 않음을 보여주고 있다. 이는 트리 구성 시 전통적인 TBCP의 특성을 보여주고 있는데 New 노드의 동시 Join 요청 시 각각의 New 노드의 Join 요청을 일정시간 제한 한 다음 순차적으로 서비스를 수행하기 때문이다. 약간 더 나은 성능을 보이고 있는 HMTP는 동시적으로 New 노드들이 Join 요청을 하였을 때

Join 요청을 동시적으로 수행하지만 소스루트(임시 부모노드)의 모든 자식 노드 리스트를 체크한 다음에 최적의 부모노드를 선택함으로써 전통적인 TBCP 보다는 더 나은 성능을 보이고 있으며 또한 트리의 상위 부모 노드의 서비스 가능 여부에도 불구하고 하위 수준에 인접한 위치에 존재하는 호스트들끼리 그룹핑이 발생하므로 HMTP에서 발생하는 그룹핑은 트리의 평균 깊이를 증가시키며 그 결과 제한한 mOBCP알고리즘에 비해 Latency 값을 증가 시키는 결과로 이어진다. 반면 제안한 기법인 mOBCP는 New 노드들이 동시에 서비스 요청을 하여도 각각의 New 노드들에게 최적의 부모노드를 선택할 수 있는 권한을 줌으로써 빠른 시간 안에 트리를 구성하며 또한 밸런스드 트리를 추구함으로써 상위부터 Out-degree를 모두 소진하면서 서비스를 수행하므로 평균 Latency 값의 차이를 줄일 수 있었다. 그림 6 과 그림 7에서도 약간의 차이를 보이면서 본 논문에서 제안한 mOBCP의 개선 된 성능차이를 확인할 수 있다.

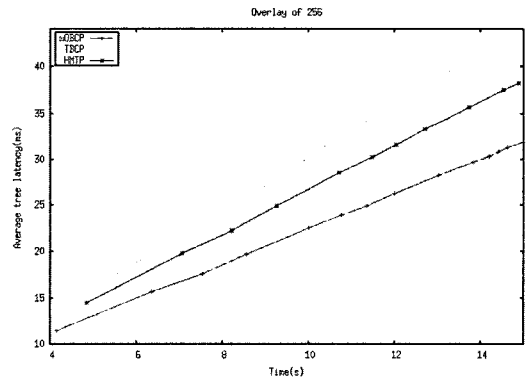


그림 5. 트리 구성 시간 15초

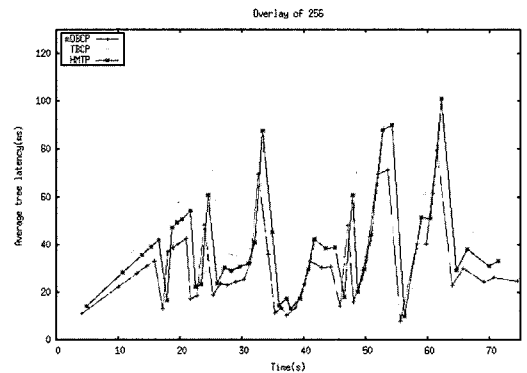


그림 6. 트리 구성 시간 75초

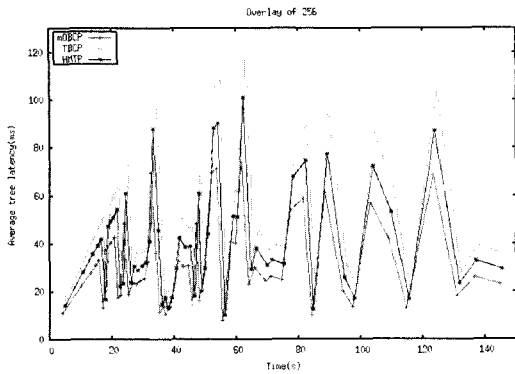


그림 7. 트리 구성 시간 150초

### III. 결론

네트워크 환경에서의 멀티미디어 서비스는 텍스트 기반의 데이터가 대부분 이었을 때부터 끊임없는 관심의 대상이었을 뿐만 아니라 그에 대한 많은 연구가 지속적으로 이루어져 왔다. 인터넷 방송, VOD 등 멀티미디어 서비스는 그 동안 공중파 TV나 VCR 등을 통한 품질을 기대하기 힘들었고 그에 따라 이용자는 많은 불편함을 감수하고 저품질의 데이터를 인내심을 가지고 서비스 받는 수준이었다. 최근 특히 국내의 네트워크 및 시스템 환경은 세계 제일의 수준으로서 HD급 멀티미디어 데이터의 수신 및 처리가 가능할 정도로 발전해 왔다. 그에 따라 방송 통신 융합 환경 및 디지털 가전으로서의 멀티미디어 서비스 수요가 증가하게 되었고 또한 네트워크 서비스 사용자는 보다 능동적이고 참여적인 역할을 하고 있다. 이것은 P2P 서비스나 양방향 개인인터넷 방송 서비스 요구등으로 나타나는 새로운 환경의 사용자의 특성을 잘 보여준다. 하지만 환경변화에도 불구하고 실제로 개인인터넷방송을 서비스 하는데 있어서 많은 난관에 가로막혀 있다.

본 논문에서는 개인 방송 서비스 시스템에 제안한 오버레이 멀티캐스트 기반 mOBCP 기법을 적용하여 서비스를 받고자 대기하는 지식 노드들의 대기 시간을 최소한으로 줄이면서 가능한 빠르게 부모 노드를 찾을 수 있도록 하여 방송 서비스의 품질을 향상시킬 수 있는 트리 구성 기법을 제안하였으며, 시뮬레이션 결과를 통해 제안된 모델이 기존의 기법들 보다 개인방송시스템에 적합한 방안을 보여주고 있다.

### 참고 문헌

[1] Mathy, L., Canonico, R., Hutchison, D.: An

Overlay Tree Building Control Protocol. In: Crowcroft, J., Hofmann, M. (eds.): *NGC. Lecture Notes in Computer Science*, Vol. 2233. Springer-Verlag, Berlin Heidelberg (2001) 7687

- [2] Chu, Y-H., Rao, S., Zhang, H.: A Case for End System Multicast. In *ACM SIGMETRICS 2000*, Santa Clare, CA, USA (June 2000) 1-12
- [3] Farinacci, D., Lin, A., Speakman, T., Tweedly, A.: Pretty Good Multicast (PGM) Transport Protocol Specification. Internet Draft draft-speakman-pgm-spec-00. *IETF* (1998)
- [4] Zhi Li and Prasant Mohapatra, "Hostcast: A New Overlay Multicasting Protocol", *IEEE Int. Communications Conference(ICC)* 2003.
- [5] Francis, P.: Yoid: Extending the Internet Multicast Architecture. Technical Report, *ACIRI*, (Apr 2000)
- [6] Beichuan Zhang, Sugih Jamin and Lixia Zhang, "Host Multicast: A Framework for Delivering Multicast To End Users", In *Proc. of IEEE INFOCOM*, New York, NY, June 2002.
- [7] Minseok Kwon and Sonia Fahmy, "Topology-Aware Overlay Networks for Group Communication", In *Proceedings of the ACM NOSSDAV*, May 2002.
- [8] David A. Helder and Sugih Jamin, "End-host multicast communication using switch-trees protocols", In *Proc. GP2PC*, Berlin, Germany, May 2002.
- [9] NS-2 Network Simulator. [www.isi.edu/nsnam/ns](http://www.isi.edu/nsnam/ns)

남 지 승 (Ji-Seung Nam)

정회원

1992년 : Univ. of Arizona, 전자공학과(공학박사)

1992년~1995년 : 한국전자통신연구소 선임연구원

1999년~2001년 : 전남대학교 정보통신특성화 센터장

2001년~2005년 : 전남대학교 인



터넷창업보육 센터장

1995년~현재 : 전남대학교 컴퓨터공학과 교수

<관심분야> 통신프로토콜, 인터넷 실시간 서비스, 라우팅

강 미 영 (Mi-Young Kang)

준회원



2001년 : 전남대학교 컴퓨터공학과(공학석사)

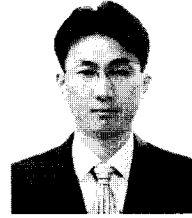
2001년~2002년 : 한국전자통신연구소 위촉연구원

2003~현재 : 전남대학교 컴퓨터공학과 박사과정

<관심분야> 통신프로토콜, 인터넷 실시간 서비스, 라우팅

손 승 철 (Seung-Chul Son)

준회원



2002년 : 전남대학교 컴퓨터공학과 졸업

2003년~2004년 : (주)금영 음향연구소

2007년 : 전남대학교 컴퓨터공학과(공학석사)

현 재 전남대학교 컴퓨터공학과 박사과정

<관심분야> 컴퓨터 네트워크, 통신프로토콜

전 진 한 (Jin-Han Jeon)

준회원



1994 : 전남대학교 전산학과

2005 : 전남대학교 컴퓨터 공학과(공학석사)

현 재 전남대학교 컴퓨터공학과 박사과정

<관심분야> 컴퓨터 네트워크, 통신 프로토콜