

# 오버레이 멀티캐스트를 적용한 확장성 있는 VoD 서비스 모델

정희원 김 경 훈\*, 손 승 철\*, 남 지 승\*

## A Scalable VoD Service Scheme Based on Overlay Multicast Approach

Kyung-Hoon Kim\*, Seung-Chul Son\*, Ji-Seung Nam\* *Regular Members*

### 요 약

인터넷 환경에서 VoD 서비스를 제공하기 위해서는 시스템 자원과 네트워크 대역폭을 효율적으로 사용해야 한다. 이러한 효율성을 높이기 위한 대안으로 오버레이 멀티캐스트를 생각 할 수 있지만 실제로 On-Demand 서비스에 적용하기 위해서는 많은 결점의 보완이 필요하다. 본 논문에서는 이러한 결점을 보완하고 기존 시스템 환경의 간단한 확장으로 VoD 서버의 자원과 대역폭을 효율적으로 사용할 수 있는 오버레이 멀티캐스트 모델을 제안 한다. 제안된 모델은 호스트의 공유 버퍼와 패칭 채널 그리고 트리 구성 알고리즘을 이용하여 임의적인 사용자 데이터 요구시점은 물론 다양한 재생을 지원 등 제한사항이 없는 On-Demand 서비스 수행이 가능하다. 시뮬레이션 결과는 제안된 모델이 기존의 유니캐스트를 이용한 경우보다 월등히 많은 사용자를 수용할 뿐 아니라 네트워크의 성능 또한 향상시켜줄 것을 보여주고 있다.

Key Words : VoD Service, Overlay Multicast, Unicast, Patching join, Overaly join

### ABSTRACT

To provide VoD Service on the Internet VoD server is required to have a large amount of system resource and network bandwidth. Therefore, overlay Multicast schemes are considered as suitable alternatives but they also have some drawbacks to support on-demand media services. In this paper, we propose an overlay multicast based on-demand media service scheme which could exploit server's system resource and its network bandwidth efficiently. Proposed scheme uses shared buffer of clients involving the relay of traffic and Patching while it gives no restrictions to this scheme compared with unicast. Our simulation results show that proposed scheme can support more user than unicast and improve the network performance at the same time.

### I. 서 론

네트워크와 사용자 PC의 성능 향상으로 인하여 멀티미디어 데이터는 더욱 고화질, 대용량화 되어 가고 있으며 이러한 상황 하에 인터넷상에서의

video on demand(VoD)서비스가 크게 주목을 받고 있다. 그러나 인터넷을 통한 VoD 서비스는 서버 측의 대역폭 문제와 많은 수의 클라이언트를 지원 하기 위한 확장성 문제 등 많은 제약을 가지고 있다<sup>[1]</sup>. 서버와 클라이언트를 유니캐스트로 연결한 전

\* 전남대학교 공과대학 컴퓨터공학과 멀티미디어 데이터통신 연구실(rodem94@empal.com)

논문번호 : KICS2005-07-284, 접수일자 : 2005년 7월 12일

※ 본 논문은 한국전자통신연구원의 “광가입자망(FTTH) 서비스개발 실험사업”에서 지원받아 수행 되었습니다.

형적인 VoD 서비스 방식을 예로 들면, 인기가 높은 콘텐츠는 필연적으로 서버 측의 부하를 가져오게 된다. 이러한 문제들을 해결하기 위해 개별적으로 많은 연구들이 수행되어 오고 있으며, 이에 대한 해결책으로 수년 전까지 IP 멀티캐스트가 적절한 메커니즘으로 여겨져 왔다. 그러나 IP 멀티캐스트는 라우터의 구현, 혼잡 제어와 신뢰성 있는 전송에서 여러 가지 문제점에 직면하게 되었으며, 그에 대한 대안으로 오버레이(overlay) 멀티캐스트 방법이 제안 되었다. 본 논문에서는 오버레이 멀티캐스트 환경 하에서 VoD 서비스를 수행하기 위해 수신 호스트가 재생 데이터를 일정시간 동안 다른 호스트를 위해 공유 버퍼에 보관 하게 하고 새로운 호스트의 서비스 요청 시 호스트 노드들의 서비스 상태를 검토하여 가장 효율적인 오버레이 멀티캐스트 전송 트리가 형성되도록 하는 기법을 제시한다. 이러한 방법은 기존의 유니캐스트 서비스를 사용하는 시스템에 비해 특별한 기능적 제약사항이 없이 멀티캐스트 서비스를 수행하여 VOD 서비스의 성능을 향상시킬 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 기술하고 3장에서는 제안된 모델의 개요 및 전체적인 동작 절차를 설명한다. 4장에서는 시뮬레이션을 통하여 기존 방법과의 성능을 비교, 분석하고 5장에서 결론을 맺는다.

## II. 관련 연구

기본적으로 구성된 네트워크 장비들을 교체하지 않고 VoD서비스를 제공하기 위해서는 오버레이 멀티캐스트 기법과 패칭기법의 조합이 필요하다. 멀티캐스트 에서 새로운 클라이언트가 접속 하였을 때, 실시간 라이브 방송에서는 오직 접속한 시간의 미디어 스트림만이 필요한 반면 On-Demand서비스에서는 미디어의 시작부분부터 접속한 시간까지의 모든 스트림이 필요하다. 따라서 미디어의 시작 부분을 전송해줄 수 있는 특별한 기법이 필요하며, 이를 패칭기법 이라고 한다<sup>5)</sup>. 오버레이 멀티캐스트는 라우터의 구현, 혼잡 제어, 신뢰성 있는 전송 등의 문제점을 지니는 IP 멀티캐스트의 대안으로 제안된 기법이며 전형적인 패칭기법 또한 이러한 IP 멀티캐스트 상에서 연구되어온 기법이다<sup>5)</sup>. 이러한 패칭기법을 오버레이 멀티캐스트 상에 적용하여 제안된 대표적인 관련 연구들을 살펴보면 다음과 같다.

P2Cast<sup>1)</sup>는 P2P네트워크와 IP 멀티캐스트의 패칭

채널, 그리고 ALM(Application Layer Multicast)을 조합하여 고안된 방법이다. 멀티캐스트 그룹에 새로운 클라이언트가 가입 하였을 때, 그 클라이언트는 미디어의 첫 부분을 VoD 서버 또는 미디어의 시작 부분을 캐쉬에 저장하고 있는 다른 클라이언트들로부터 수신 받게 된다. P2Cast 세션내의 클라이언트들은 수신된 스트림을 모두 캐쉬에 저장하고 있는 반면, 본 논문에서 제안된 기법에서는 캐쉬가 아닌 특정크기의 공유버퍼를 이용한다. 또한 P2Cast 에서는 클라이언트의 세션 이탈을 고려해서 데이터 손실을 복구하는 방법을 제시하고 있지만 클라이언트들의 이탈이 활발한 경우 한 클라이언트가 다수의 클라이언트들에게 스트림을 전송해 주어야만 서비스가 가능하다는 문제점을 가지고 있다.

P2Vod<sup>2)</sup>는 IP 주소의 부족, 보안 및 보급상의 문제로 실현 불가능한 IP 멀티캐스트에 대한 대안으로 제안된 오버레이 멀티캐스트를 이용하여 VoD 서비스를 수행하는 P2Cast 기반의 또 다른 방법이다. 비디오 스트림을 시간단위의 데이터로 블록화하여 서비스를 수행하고 P2Vod 세션에 가입한 모든 클라이언트들은 이러한 데이터 블록 중 가장 최근의 블록을 저장 하기위한 특정 크기의 버퍼를 구성하게 된다. 같은 데이터 블록을 보유하고 있는 클라이언트들을 동일 차수로 그룹화 하여, 새로운 요청이 있는 클라이언트는 자신보다 상위 차수에 있는 모든 클라이언트들을 검사하고, 요청한 스트림 블록을 보유하고 있는 클라이언트에게서 서비스를 받게 된다. 이러한 그룹화는 장애가 발생 했을 때 복구에는 효과적일 수 있지만 메커니즘의 구현이 복잡해질 수 있는 단점이 있다.

## III. On-Demand Overlay Multicast 모델

### 3.1 개요

제안 모델에서 VoD 서버는 클라이언트의 ID, 세션 ID, 클라이언트의 서비스 요청 시간, 클라이언트의 상태 필드를 유지하고 있는 세션 테이블을 이용하여 서비스를 수행한다. 멀티캐스트 그룹을 구성하는 모든 클라이언트들은 특정 길이의 공유 버퍼를 가지고 있으며, 이 공유 버퍼는 부모 노드나 최상위 VoD 서버로부터 수신된 스트림을 저장하고 그 스트림을 자식 노드에게 전달하는데 이용된다. 제안 모델에서는 오버레이 멀티캐스트와 패칭기법을 동시에 이용한다. 오버레이 멀티캐스트 기법은 효율적으

로 VoD서비스를 수행하기 위한 오버레이 트리를 구성하는데 사용되며, 패칭 기법은 미디어의 시작부분을 부모 노드로부터 수신 받지 못할 경우 VoD 서버로부터 일정기간 데이터를 수신, 부모노드의 멀티캐스트 세션에 합류할 수 있도록 한다. 이는 공유 버퍼 크기에 의해 제한된 시스템의 성능을 높이기 위해 고려되었다.

아래 그림 1은 앞에서 설명한 트리의 형성 과정을 개괄적으로 도식화한 예이다. 클라이언트의 공유 버퍼의 크기가 5주기 일 때, 15주기 요청시점에 새로운 클라이언트 CL5가 가입을 요청하면 오버레이 멀티캐스트 최적화 알고리즘에 의해 CL4를 부모노드로 하여 멀티캐스트 그룹에 가입하게 된다. 이때 모든 클라이언트들은 미디어의 시작 부분을 공유 버퍼에 소유하고 있지 않기 때문에 CL5는 VoD 서버와 직접 연결되어 일정시간 미디어의 시작 부분을 수신 받게 된다.

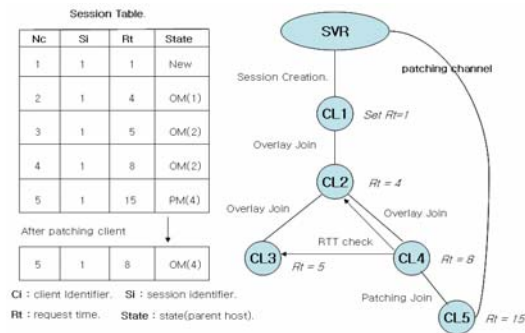


그림 1. 세션 테이블과 멀티캐스트 전송 트리의 구성

### 3.2 클라이언트의 가입

VoD세션에 가입을 원하는 클라이언트는 최초로 VoD서버에 가입 요청을 한다. 새로운 클라이언트로부터 가입 요청을 수신한 VoD서버는 세션 테이블을 검사한 후 새로운 클라이언트의 가입 타입을 결정하고 요청한 클라이언트에게 가입 타입과 부모 노드가 될 수 있는 클라이언트 리스트를 송신한다. 새로운 클라이언트는 수신된 이러한 정보들을 이용해 부모 노드를 선택하고 결과적으로 VoD서비스 세션에 가입되거나 새로운 VoD 서비스 세션을 만든다.

클라이언트의 가입 타입은 overlay join, patching join, session creation과 같은 세 가지 종류의 가입 타입 중 하나가 될 수 있다.

#### 1) Overlay Join(OM join)

Overlay join 형식으로 가입된 클라이언트는 오직

다른 클라이언트에 의해서만 스트림을 수신 받는다. overlay join 형식으로 세션에 가입되기 위해서는 미디어의 시작부분을 공유 버퍼에 저장하고 있으면서 부모 노드가 될 수 있는 클라이언트가 요구 미디어 세션 내에 하나 이상 존재해야 된다. 이러한 조건을 만족하는 상태를 식으로 표현하면 아래와 같다. 여기서  $Rt(New)$ 는 새롭게 가입한 클라이언트의 요청 시간을 의미하고,  $Rt(i)$ 는 부모노드가 될 수 있는 현재 서비스 중인 노드들의 요청시간을 의미하며,  $M$ 은 클라이언트들의 공유 버퍼 크기를 의미한다. 위의 정보들은 서버의 세션 테이블을 통해 얻을 수 있다.

$$Rt(new) < Rt(i) + M \tag{1}$$

식(1)의 조건에 만족하는 클라이언트는 현재 자신의 공유버퍼에 미디어의 시작부분을 지니고 있다는 것을 의미한다. 식(1)을 만족하는 클라이언트들 중 다음의 조건을 만족하는 클라이언트가 새롭게 가입한 클라이언트의 부모 노드로 선택된다. 먼저 비디오 스트림을 서비스 해줄 수 있는 충분한 대역폭을 가지고 있어야 된다. 만약 충분한 대역폭을 가지지 못했다면 그 클라이언트는 부모 노드 후보에서 제외된다. 두번째로 대역폭이 충분한 노드들 가운데 새롭게 가입한 클라이언트와의 RTT가 최소 값인 클라이언트가 부모 노드로 선택 되어진다. 만일 위의 조건들을 만족하는 호스트가 없어 Overlay join 형식으로 세션에 가입될 수 없다면 Patching join 형식을 검사 하게 된다.

#### 2) Patching Join(PM join)

Patching join 형식으로 가입된 클라이언트는 다른 클라이언트와 VoD서버에게서 동시에 스트림을 수신 받는다. 이러한 patching join 상태는 VoD서버로부터 패칭 스트림의 수신이 완료 되면 서버와의 연결은 종료되고 가입된 클라이언트는 overlay join 상태로 변경 된다. patching join이 되기 위한 조건을 식으로 표현하면 아래와 같다.

$$Rt(new) < Rt(i) + 2M - 1 \tag{2}$$

식(2)를 만족하는 클라이언트는 공유버퍼의 재생 시간 이하의 일정 시간동안 서버로부터 미디어의 시작부분을 수신한 후 서버와의 연결을 해제하고 부모 노드의 세션에 합류될 수 있는 클라이언트들을 의미한다.

patching join 역시 Overlay join과 마찬가지로 노드의 대역폭을 고려하고 다수의 부모노드 후보가 존재 할 때 RTT가 최소인 클라이언트를 부모노드로 결정한다.

이러한 patching join의 조건을 만족하는 클라이언트가 세션 내에 존재 하지 않는다면 새로운 VoD 서비스 세션이 만들어지게 된다.

### 3) Session creation

새로운 클라이언트가 overlay join과 patching join의 조건을 만족하는 부모노드를 찾지 못한다면 그 클라이언트는 새로운 VoD서비스 세션을 생성한다. 새로운 클라이언트는 VoD서버와 직접 연결을 생성하고 VoD 서비스 세션의 루트 역할을 담당하게 되며 VoD 서버로부터 비디오 스트림을 수신 받게 된다.

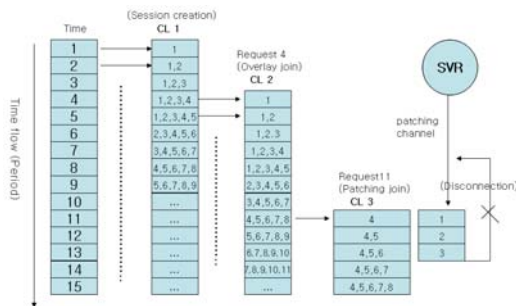


그림 2. 공유버퍼 Size가 5주기일 때 클라이언트의 버퍼상태

그림 2는 각각의 클라이언트가 가지고 있는 공유 버퍼의 상태를 도식화 한 것이다. 그림에서 시간 단위는 전체 미디어의 서비스 시간을 일정 단위로 나누는 주기를 의미 하며, 모든 클라이언트는 다른 사용자들 위해 재생 후 보관하는 사용자 공유 버퍼를 5 주기의 크기로 보유하고 있다. 처음 연결되어 새로운 세션을 설정한 CL1의 공유 버퍼는 그림과 같이 미디어 주기 블록을 저장 할 것이다. CL2는 4번째 주기에 서비스를 요청했고 서버는 메모리에 서비스 미디어의 시작 데이터를 보관하고 있는 CL1의 정보를 전송하고 이 정보에 의하여 CL2는 CL1에 overlay join이 되고 자신의 연결 정보를 서버에 전송한다. 새로운 클라이언트 CL3이 가입 요청을 했을 때 상위 노드들의 공유 버퍼에는 미디어의 시작 부분이 존재하지 않는다. 이때 CL3은 CL2에 patching join 되고 CL2와 VoD서버에 동시에 연결되어 미디어의 시작부분을 서버로부터 수신 받게 되고 시작부분의 수신이 완료되면 VoD서버와의 연

결은 종료되고 CL2로부터 다음 스트림을 전송 받게 된다. 이 경우 서버로부터의 패칭 수신이 종료되는 시점으로부터 세션 테이블의 가입형식은 overlay join으로 변경된다.

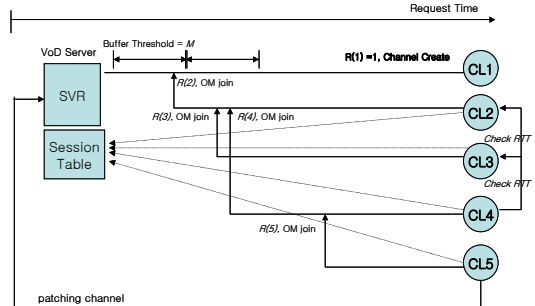


그림 3. 호스트수가 5일 때 join 과정

그림 3은 다섯 개의 호스트의 join 과정을 나타낸 것이다. CL4의 경우 CL2와 CL3, 두 개의 클라이언트에 overlay join될 수 있어 RTT 체크를 통하여 물리적인 네트워크 상황이 더 좋은 CL2에 overlay join 되었다. overlay join이나 patching join이 될 수 있는 호스트가 다수 검색 되었을 경우 제안 모델에서는 기본적으로 On-Demand 서비스에 맞게 트리의 깊이를 고려하지 않고 RTT 체크를 통하여 최적의 호스트 노드를 결정한다.

아래 그림 4는 크기가 5주기인 공유 버퍼를 갖는 클라이언트의 버퍼 구조를 나타낸 것이다. 새로운 채널 설정이나 overlay join이 이루어진 경우 General Queue는 자신의 서비스 재생 시점에서 M 만큼의 데이터를 다른 호스트를 위해 보관한다. 만일 호스트가 overlay join되지 못하고 일정기간 서버와의 패칭을 통해 데이터를 전송 받게 될 때는 패칭을 위한 Queue를 임시로 설정하고 이 Queue의 데이터를 먼저 재생한 후 패칭 채널이 해제 되었을 때 General Queue의 데이터를 재생한다.

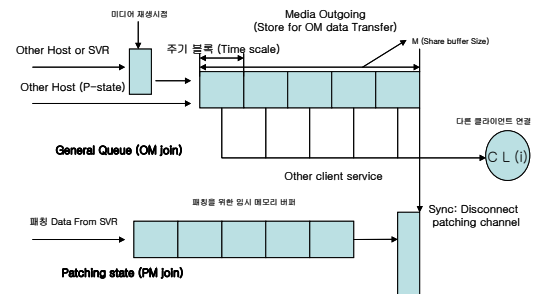


그림 4. 호스트의 Queue Structure

### 3.3 장애 복구

제안된 모델에서 클라이언트의 장애나 인위적인 세션 이탈 등은 오버레이 트리를 분할 시켜 버리는 심각한 상황을 초래 할 수 있다. 다시 말해 이탈된 클라이언트로부터 서비스를 받고 있던 모든 자식 클라이언트들이 이탈된 시점으로부터 서비스를 받지 못하는 결과가 생길 수 있다. 따라서 이러한 장애가 발생했을 때 세션을 빠르게 복구하기 위한 메커니즘이 필요하다.

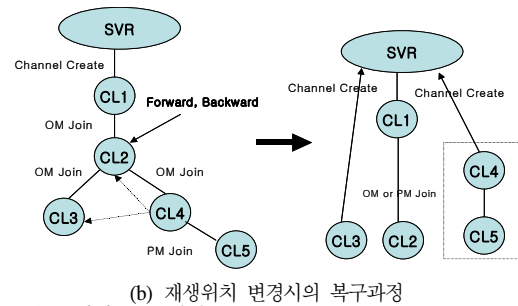
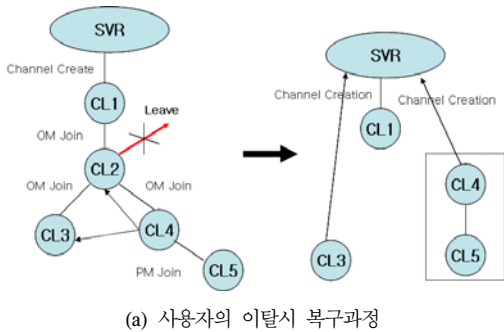


그림 5. 장애 복구 방법

그림 5(a)는 오버레이 트리가 구성된 상태에서 CL2가 세션을 이탈할 때의 복구 과정을 보여주고 있다. 장애가 감지되었을 때 장애노드 CL2의 자식 노드인 CL3과 CL4는 새로운 세션을 생성하게 된다. 간단하면서도 장애 복구에 소요되는 시간이 적고 오버헤드를 최소화 시킬 수 있는 효율적인 방법이라고 할 수 있다.

재생 위치를 변경한 호스트의 경우 호스트 탈락 후 요구 위치주기를 기준으로 가입 알고리즘을 통하여 재가입 시키고 그 자식노드들은 호스트 이탈 시와 같이 복구한다. 그림 5(b)는 이 같은 경우에 복구된 트리를 나타낸 것이다. 일반적으로 많은 사용자가 동일트리에 가입할수록 시스템의 성능은 증가한다. 하지만 위와 같은 장애 상황은 트리 구성 노드가 증가할수록 많이 발생 할 것이므로 세션의

크기를 적절하게 유지하기 위하여 서버가 적절한 시간을 기준으로 사용자 요청을 새로운 세션으로 분기 시키는 것이 복구 시 호스트들의 서비스 동기화와 전체 시스템 성능에 도움이 될 수 있다.

## IV. 성능 평가

### 4.1 시뮬레이션 환경

제안된 모델의 성능을 평가하기 위해 GT-ITM을 이용하여 생성한 서로 다른 다섯 개의 Transit-Stub 네트워크 토폴로지를 사용하였다. Transit-Stub 모델은 상위 레벨인 Transit영역과 하위 레벨인 Stub영역이 연결된 2 계층으로 이루어져 있다. Transit영역끼리의 연결 대역폭을 50Mbps부터 100Mbps까지, Transit영역과 Stub영역의 연결 대역폭을 10Mbps부터 50Mbps까지, Stub영역끼리의 연결 대역폭을 5Mbps부터 10Mbps까지 변화 시켜보았고 LAN상의 호스트 수에는 제한을 두지 않고 성능을 평가 하였다.

시뮬레이션을 위해 각각 다른 재생률을 갖는 1시간 30분(5400초) 길이의 10가지의 비디오 타입을 사용하였다. 각 호스트는 비디오 ID, 서비스 요청 시간을 포함하고 있으며, 이러한 호스트의 수를 100 명부터 2000명까지의 증가 시켜가며 성능을 평가 하였다. 비디오 ID는 Zipf's Rule<sup>[3]</sup>을 사용, 인기도를 반영하였으며 사용자들의 서비스 요청은 Poisson 분포에 의해 1초부터 10800초까지 생성되도록 구성 하였다. 이렇게 10800초 즉, 3시간까지 서비스 요청이 생성되도록 한 이유는 최초 서비스의 시작 시점으로부터 1시간 30분 길이의 미디어가 종료될 때 접속한 사용자들은 다시 1시간 30분 동안 서비스를 받게 될 것이기 때문이다. 또한 VoD서버는 Transit영역과 Stub영역에 관계없이 어떤 곳에도 위치할 수 있다고 가정 하였고 호스트들의 버퍼링 주기를 1분에서 5분까지 변화시켜 가면서 성능을 평가 하였다.

### 4.2 결과

아래의 그래프들은 제안된 모델과 유니캐스트를 이용한 방법을 비교 분석한 결과이다. 각 그래프 상에서 m은 클라이언트들의 공유 버퍼 크기를, HOD는 제안된 모델을 나타낸다. 공유 버퍼의 크기를 1부터 5분까지 변경시켜가면서 전체적인 트래픽과 대역폭을 조사하였다.

#### 1) Total Network Traffic

그림 6은 클라이언트의 수에 따라 전체 네트워크

트래픽이 변화되는 모습을 보여준다. 전체적으로, HOD를 사용한 방법이 유니캐스트를 사용한 방법에 비해 큰 성능 향상이 없는 이유는 서비스를 요청한 클라이언트들의 불규칙적인 위치 및 시간과 관련이 있다. HOD를 사용한 방법에서 공유 버퍼 크기가 증가할수록 트래픽은 감소하고 있다는 사실을 알 수 있는데 이는, 공유 버퍼가 클수록 클라이언트간의 연결이 patching join 보다는 overlay join 형태를 취하기 때문에 나타날 수 있는 현상이다. 그러나 공유 버퍼의 크기에 제한을 두지 않는다면 트래픽은 현저히 감소될 수 있지만, 악의를 지닌 사용자에 의한 보안상의 문제, 호스트의 자원 공유문제 등을 고려하여 공유 버퍼의 크기를 결정해야 한다.

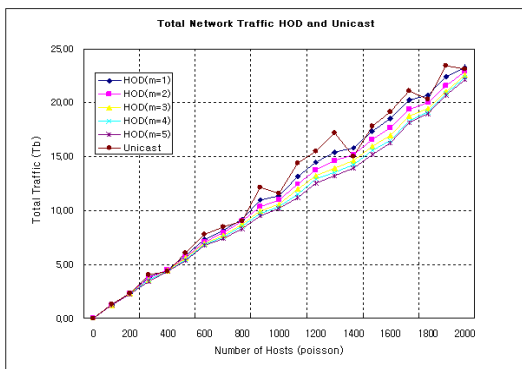


그림 6. Total Network Traffic-I

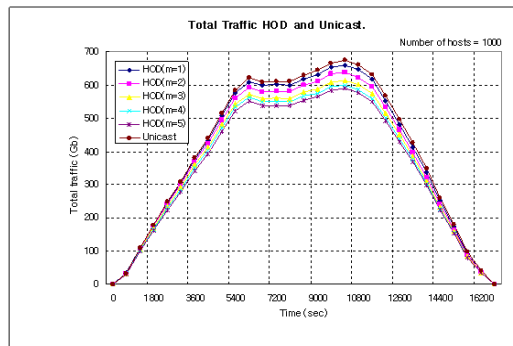


그림 7. Total Network Traffic-II

2) VoD 서버의 대역폭

아래 그림 8은 클라이언트의 수에 따라 VoD 서버상의 네트워크 트래픽이 변화되는 모습을 보여주고 있다. 유니캐스트를 사용한 방법의 경우에는 HOD를 사용한 방법들에 비해 VoD 서버상의 트래픽이 크게 증가되는 모습을 볼 수 있다. 유니캐스트를 사용한 경우, 모든 사용자들이 서버에 직접 연결

되기 때문에 서버측의 트래픽은 사용자와 비례하여 증가한다. 따라서 유니캐스트를 이용하여 수많은 사용자에게 서비스를 수행하는 VoD 시스템의 구성은 비용 면에서 현실적으로 거의 불가능한 일이다. HOD를 사용한 경우에는 스트림이 호스트간 릴레이 전송 되고 미디어의 시작부분을 공유버퍼에 보유한 호스트가 고루 분포되어 있기 때문에 실제 VoD 서버상의 트래픽은 유니캐스트에 비해 현저하게 감소된다. 공유버퍼의 크기에 따라 서버상의 트래픽에는 큰 차이가 없지만 새로운 세션의 생성이나 patching join 형태의 가입은 서버측의 부하를 증가시키는 요인으로 작용할 수 있다. 반면 overlay join 형태의 가입은 서버측의 부하를 감소시킨다. 공유버퍼의 크기가 클수록 그러한 overlay join 형태의 가입이 증가하게 되어 결과적으로 서버측의 부하는 감소하게 된다. 하지만 공유버퍼의 크기에 제한을 두지 않을 수 없으며 앞서 언급한 것과 같이 보안이나 자원공유문제 등을 고려하여 그 크기를 결정해야 할 것이다.

그림 9는 시간의 흐름에 따른 VoD 서버상의 네트워크 트래픽 변화를 보여주고 있다. 유니캐스트를 사용한 방법에서는 모든 사용자들이 서버에 직접

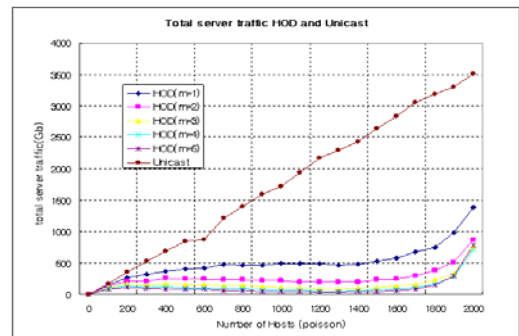


그림 8. Server Bandwidth-I

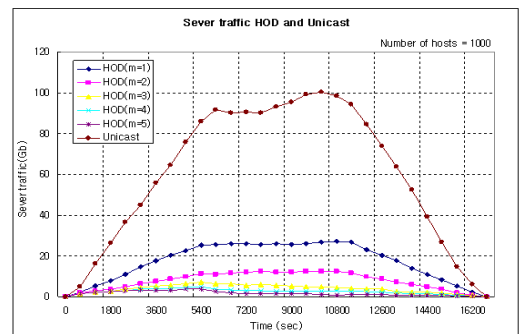


그림 9. Server Bandwidth-II

연결되어 서비스를 받기 때문에 서버측의 트래픽은 그림 7의 전체 네트워크 트래픽과 유사한 모습을 보이고 있다. HOD를 사용한 방법의 경우, 공유버퍼의 크기에 따라 다소 차이는 있지만 대체적으로 유니캐스트를 사용한 방법에 비해 서버측의 트래픽이 상당히 감소된 모습이다. 그림 8과 그림 9의 결과로 볼 때, HOD를 이용한 방법이 유니캐스트를 이용한 경우보다 월등히 많은 사용자를 수용하며 네트워크 또한 보다 효율적으로 사용할 수 있음을 알 수 있으며 HOD방법에서 공유버퍼의 크기를 3분 이상으로 설정하는 것은 큰 성능향상의 요소가 되지 않는다는 것도 알 수 있다.

### V. 결론

본 논문에서는 오버레이 멀티캐스트와 패칭기법을 이용하여 기존의 네트워크 장비를 교체하지 않고 확장이 용이한 VoD서비스 모델을 제안 하였다. 제안된 모델에서 VoD 서버는 클라이언트의 상태를 유지하고 있는 세션 테이블을 이용하여 서비스를 수행한다. 이러한 방법은 기존의 유니캐스트 서비스 방법보다 특별히 많은 제어 정보가 필요하지 않으며 호스트의 관리 및 성능 제어, 복구 등에 있어서 기존의 방법들보다 더 효율적이며 간단하게 구현, 적용 될 수 있다.

모의실험을 통해서 기존의 유니캐스트 방법과 제안된 HOD방법의 전체 네트워크와 서버측의 트래픽을 비교한 결과 HOD를 이용한 방법이 네트워크의 효율적 사용, 서버 부하 감소 등의 멀티캐스트의 이익을 얻을 수 있으며 이를 통하여 최소한의 서버 자원으로 다수의 클라이언트를 지원할 수 있음을 확인 하였다. 따라서 제안된 시스템은 일정 규모 이상의 사용자를 지원하는 실제 VoD 서비스환경의 상용성을 그대로 유지하여 실용적으로 적용될 수 있음을 알 수 있다.

### 참고 문헌

[1] Yang Guo, Kyoungwon Suh, Jim Kurose, Don Towsley, "P2Cast: P2P Patching Scheme for VoD Service", in WWW 12th, 2003.  
 [2] Tai Do, Kien A. Hua, and Mounir Tantaoui, "P2VoD: Providing Fault Tolerant Video-on-Demand Streaming in Peer-to-Peer Environ-

ment", in Proc. of the IEEE International Conference on Communications (ICC 2004), June 2 0-24 2004, Paris, France  
 [3] A.Chervenak. D. Patterson, and R. Katz, "Choosing the Best Storage System for Video Service," in proc. Of ACM Multimedia 95, pp. 109-119, August. 1995  
 [4] Yang-Hua Chu, Sanjay G. Rao, and Hui Zang, "A Case for End System Multicast", in Proc. of ACM SIGMETRICS 2000, 2000, pp. 1-12.  
 [5] K. Hua, Y. Cai, and S. Sheu, "Patching: A multicast technique for true video-on-demand services," in Proc. ACM Multimedia, September 1998.  
 [6] Duc A. Tran, Kien A. Hua, Simon Sheu, "A New Caching Architecture for Efficient Video-on-Demand Services on the Internet", in IEEE Symposium on Applications and the Internet (SAINT 2003), 2003.  
 [7] P. Francis. Yoid: Extending the Multicast Internet Architecture, 1999. White paper <http://www.aciri.org/yoid/>.  
 [8] Y.-H. Chu, S. G. Rao, S. Seshan, and H. Zhang. Enabling Conferencing Applications on the Internet using an Overlay Multicast Architecture. In Proceedings of ACM SIGCOMM, August 2001.  
 [9] Y.-H. Chu, S. G. Rao, and H. Zhang. A Case for End System Multicast. In Proceedings of ACM SIGMETRICS, June 2000.  
 [10] B. Zhang, S. Jamin, and L. Zhang. Host multicast: A framework for delivering multicast to end users. In Proceedings of IEEE Infocom, June 2002.  
 [11] S. Banerjee and B. Bhattacharjee. Analysis of the NICE Application Layer Multicast Protocol. Technical report, UMIACSTR 2002-60 and CS-TR 4380, Department of Computer Science, University of Maryland, College Park, June 2002.  
 [12] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable application layer multicast. In Proceedings of ACM Sigcomm, August 2002.  
 [13] M. Castro, P. Druschel, A-M. Kermarrec, and A. Rowstron. SCRIBE: A large-scale

and decentralized application-level multicast infrastructure. IEEE Journal on Selected Areas in communications (JSAC), 2002. To appear.

김 경 훈 (Kyung-Hoon Kim)   정회원  
2000년~2003년 (주)포스트립 개발팀  
2002년 2월 전남대 컴퓨터공학과 (박사수료)  
2000년 2월 전남대 컴퓨터공학과(공학석사)  
2004년 1월~현재 (주)포스트립대표  
<관심분야> 실시간통신 프로토콜, 분산처리, P2P

손 승 철 (Seung-Chul Son)   정회원  
2002년 2월 전남대학교 컴퓨터공학과 졸업  
2003년~2004년 (주) 금영 음향연구소  
현재 전남대학교 컴퓨터공학과 석사과정  
<관심분야> 컴퓨터 네트워크, 통신 프로토콜

남 지 승 (Ji-Seung Nam)   정회원  
1992년 Univ. of Arizona, 전자공학과(공학박사)  
1992년~1995년 한국전자통신연구소 선임연구원  
1999년~2001년 전남대학교 정보통신특성화 센터장  
2001년~현재 전남대학교 인터넷창업보육 센터장  
1995년~현재 전남대학교 컴퓨터공학과 교수  
<관심분야> 통신 프로토콜, 인터넷 실시간서비스, 라우팅