

오버레이 멀티캐스트 기반의 확장성 있는 VoD 서비스 모델

이형옥*, 손승철*, 광용완*, 하연정*, 남지승*

*전남대학교 컴퓨터정보통신공학과

e-mail:narcis@freechal.com

A scalable VoD Service Model based on Overlay Multicast

Hyung-Ok Lee*, Seung-Chul Son*, Yong-Wan Kwag*,
Youn-Jung Ha*, Ji-Seung Nam*

*Dept of Computer Engineering, Chon-nam University

요 약

인터넷 상에서 VoD 서비스를 제공하기 위해서는 시스템 자원과 네트워크 대역폭을 효율적으로 사용해야 한다. 이러한 효율성을 높이기 위해서 오버레이 멀티캐스트를 생각 할 수 있지만 실제로 On-Demand 서비스에 적용하기 위해서는 많은 결점의 보완이 필요하다. 본 논문에서는 이러한 결점을 보완하고 기존 시스템의 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 사용자 데이터의 요구시점은 물론 다양한 재생을 지원 등 제한이 없는 On-Demand 서비스 수행이 가능하다.

1. 서론

최근 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 클라이언트를 지원하기 위한 확장성 문제 등 많은 제약을 가지고 있다[1]. 예로 들어, 서버와 클라이언트를 유니캐스트로 연결한 전형적인 VoD 서비스 방식은 인기가 높은 콘텐츠는 반드시 서버 측의 부하를 가져오게 된다. 이러한 문제들을 해결하기 위해 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 환경에서 VoD 서비스를 수행하기 위해 수신 호스트가 재생 데이터를 일정시간 동안 다른 호스트를 위해 공유 버퍼에 보관 하게 하고, 새로운 호스트 요청 시 호스트 노드들의 서비스 상태를 검토하여 가장 효율적인 멀티캐스트 트리가 형성되도록 하는 기법을 제시한다. 이러한 방법은 기존의 유니캐스트 서비스를 사용하는 시스템에 비해 특별한 제약사항 없이 멀티캐스트의 성능을 향상시킬 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 기술하고 3장에서는 제안된 모델의 개요 및 전체적인 동작 절차를 설명한다. 4장에서는 추후 연구과제와 함께 결론을 맺는다.

2. 관련연구

기본적으로 구성된 네트워크 장비들을 교체하지 않고 VoD 서비스를 제공하기 위해서는 오버레이 멀티캐스트 기법과 패칭 기법의 조합이 필요하다. 새로운 클라이언트가 접속 하였을 때, 실시간 방송에서는 오직 접속한 시간의 미디어 스트림만이 필요한 반면 On-Demand 서비스에서는 미디어의 시작부분부터 접속한 시간까지의 모든 스트림이 필요하다. 따라서 미디어의 시작 부분을 전송해줄 수 있는 특별한 기법이 필요하며, 이를 패칭기법 이라고 한다. 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. 이러한 패칭기법을 오버레이 멀티캐스트 상에 적용하여 제안된 대표적인 관련 연구들을 살펴보면 다음과 같다.

P2Cast[1]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 부분

을 VoD 서버 또는 미디어의 시작부분을 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 P2Cast에서는 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[2]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 블록화 하여 서비스를 수행하고 P2Vod 세션에 가입한 모든 클라이언트들은 이러한 데이터 블록 중 가장 최근의 블록을 저장 하기위한 특정 크기의 버퍼를 구성하게 된다. 같은 데이터 블록을 보유하는 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.

3. On-Demand Overlay Multicast 모델

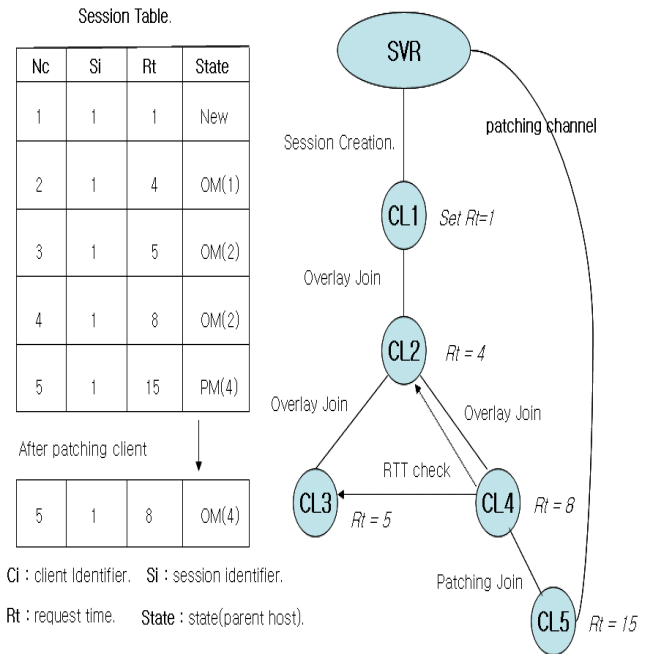
1. 개요

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 효율적으로 VoD서비스를 수행하기 위한 오버레이 트리를 구성하는데 사용되며, 패칭 기법은 미디어의 시작부분을 부모 노드로부터 수신 받지 못했을 경우 VoD 서버로부터 데이터를 수신, 멀티캐스트 세션에 합류시켜 공유버퍼 크기에 의해 제한된 시스템의 성능을 높이기 위해 사용된다.

아래 그림 1은 앞서 설명한 과정을 도식화한 예이다. 클라이언트의 공유버퍼의 크기가 5주기 일 때, 15주기 요청 시점에 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 VoD서버와 직접 연결되어 일정시간 미디어의 시작 부분을 수신 받게 된다.

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.



(그림 1) CL5가 멀티캐스트 그룹에 가입되는 과정

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 join형식으로 가입된 클라이언트는 오직 다른 클라이언트에게 스트림을 수신 받을 수 있다. overlay join형식으로 세션에 가입되기 위해서는 미디어의 시작부분을 공유 버퍼에 저장하고 있으면서 부모 노드가 될 수 있는 클라이언트가 VoD세션 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 노드들의 요청시간을 의미하며, M 은 클라이언트들의 공유 버퍼 사이즈를 의미한다.

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

식(1)을 만족하는 클라이언트들 중 다음 조건을 만족하는 클라이언트가 새롭게 가입한 클라이언트의 부모 노드

로 선택된다. 먼저 VoD 스트림을 서비스 해줄 수 있는 충분한 대역폭을 가지고 있어야 된다. 만약 충분한 대역폭을 가지지 못한다면 그 노드는 부모 노드가 될 수 없다. 두 번째로 부모 노드가 될 수 있는 클라이언트들 중 새롭게 가입한 클라이언트와의 RTT가 최소 값인 클라이언트가 부모 노드로 선택 될 수 있다. 위의 조건을 만족하지 못해서 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 VoD 서버에게서 동시에 스트림을 수신 받는다. 이러한 patching join 상태는 VoD 서버로부터 패칭 스트림의 수신이 완료 되면 VoD 서버와의 연결은 종료되고 가입된 클라이언트는 overlay join 상태로 변경 된다. 이러한 patching join 조건을 만족하는 상태를 식으로 표현하면 아래와 같다. 여기서 $Rt(New)$ 는 새롭게 가입한 클라이언트의 요청 시간을 의미하고, $Rt(i)$ 는 부모노드가 될 수 있는 노드들의 요청시간을 의미하며, M 은 공유 퍼버 사이즈를 의미한다.

$$Rt(new) < Rt(i) + 2M - 1 \quad (2)$$

Patching join은 새롭게 가입된 클라이언트의 부모노드가 미디어의 시작 부분을 가지고 있지 않을 때 이루어지는 형식이며 이러한 patching join의 조건을 만족하는 클라이언트가 세션 내에 존재 하지 않는다면 새로운 VoD 서비스 세션이 만들어지게 된다.

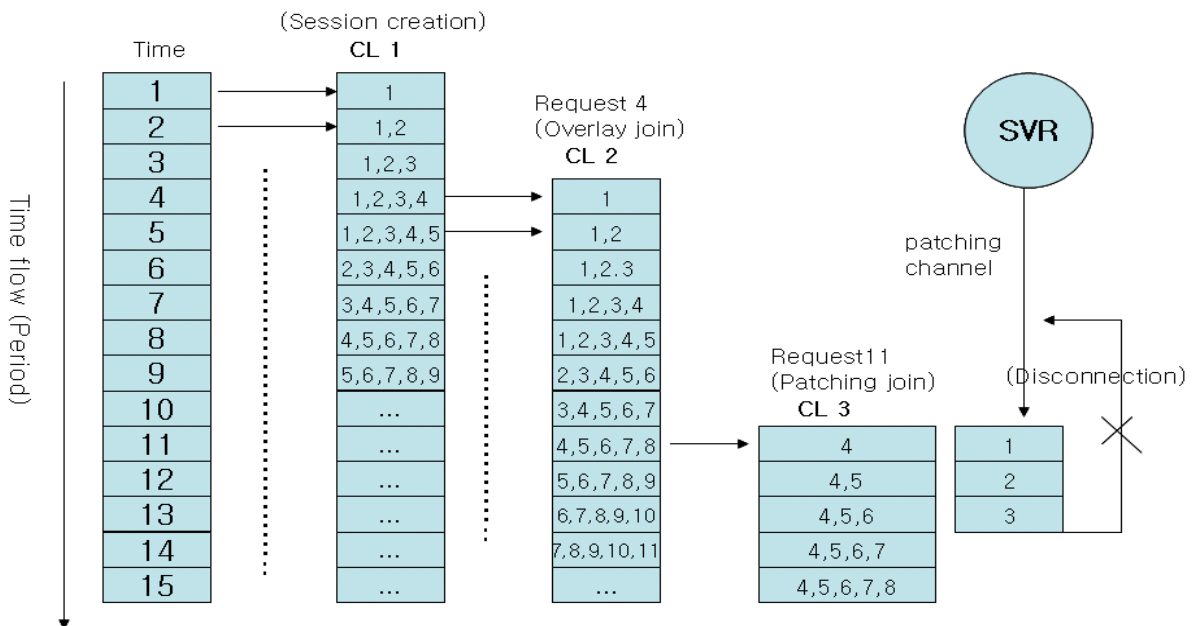
3) Session creation

새로운 클라이언트가 overlay join과 patching join의 조건을 만족하는 부모노드를 찾지 못한다면 그 클라이언트는 새로운 VoD 서비스 세션을 생성한다. 새로운 클라이언

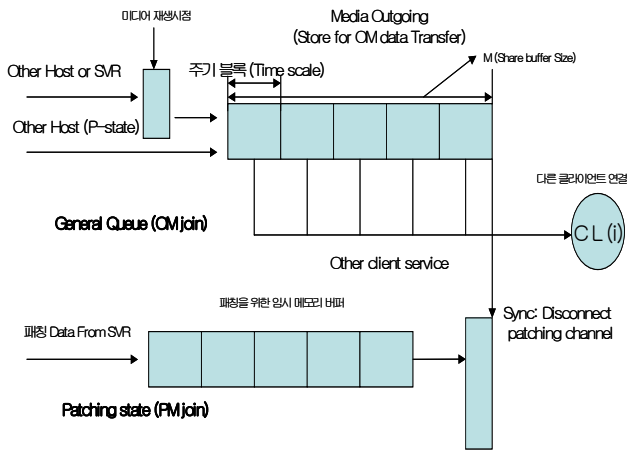
트는 VoD 서버와 직접 연결을 생성하고 VoD 서비스 세션의 루트 역할을 담당하게 되며 VoD 서버로부터 비디오 스트림을 수신 받게 된다.

그림 2는 각각의 클라이언트가 가지고 있는 공유버퍼의 상태를 도식화 한 것이다. 그림에서 시간 단위는 전체 미디어의 서비스 시간을 일정 단위로 나눈 주기를 의미 하며, 모든 클라이언트는 다른 사용자를 위해 재생 후 보관하는 사용자 공유 버퍼를 5주기의 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 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 VoD 서버에 동시에 연결되어 미디어의 시작부분을 서버로부터 수신 받게 되고 시작부분의 수신이 완료되면 VoD 서버와의 연결은 종료되고 CL2로부터 다음 스트림을 전송 받게 된다. 이 경우 가입 형식 또한 overlay join으로 변경된다.

아래 그림 3은 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.



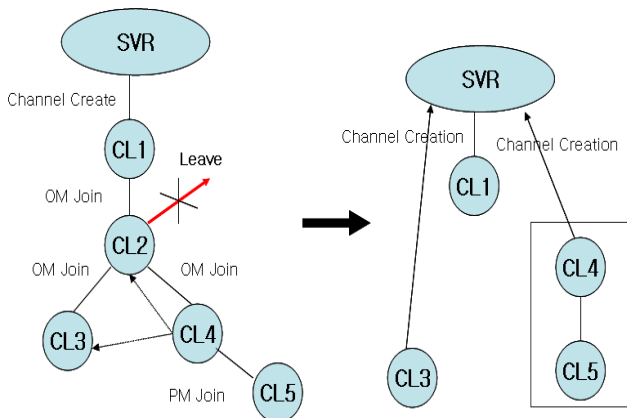
(그림 2) 각 클라이언트의 공유 버퍼



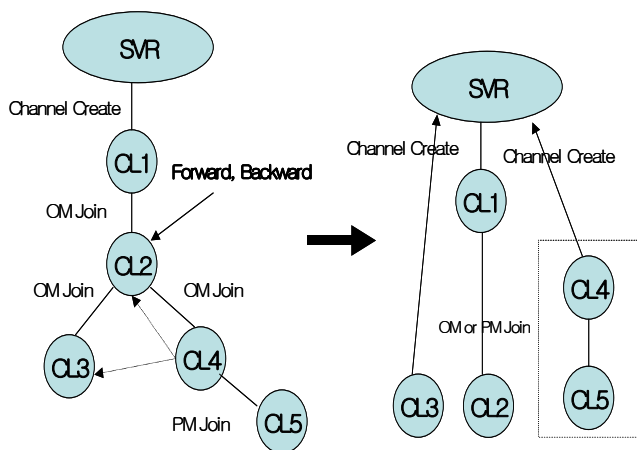
(그림 3) 호스트의 Queue Structure

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 4) 사용자의 이탈시 복구과정



(그림 5) 재생위치 변경시 복구과정

그림 4는 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 재 가입시키고 그 자식노드들은 호스트 이탈 시와 같이 복구한다. 그림 5는 이 같은 경우에 복구된 트리를 나타낸 것이다. 많은 사용자가 트리에 가입되어 질수록 시스템의 성능은 증가한다. 하지만 위와 같은 예외상황에서의 서버 세션의 증가에 따라 서버가 적정한 시간을 기준으로 사용자 요청을 새로운 세션으로 분기 시키는 것이 전체 시스템 성능에 도움이 될 수 있다.

4. 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.

향후 연구과제로는 본 논문에서 제안한 모델을 구현하고 모의실험을 통해 성능을 평가하는 것이다.

참고문헌

- [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 Environment", in Proc. of the IEEE International Conference on Communications (ICC 2004), June 20-24 2004, Paris, France
- [3] 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.
- [4] Duc A. Tran, Kien A. Hua, SimonSheu, "A New Caching Architecture for Efficient Video-on-Demand Services on the Internet", in IEEE Symposium on Applications and the Internet (SAINT2003), 2003.
- [5] P. Francis. Yoid: Extending the Multicast Internet

Architecture, 1999. White paper <http://www.aciri.org/yoid/>.

[6] 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.

[7] Y.-H. Chu, S. G. Rao, and H. Zhang. A Case for End System Multicast. In Proceedings of ACM SIGMETRICS, June 2000.

[8] B. Zhang, S. Jamin, and L. Zhang. Host multicast: A framework for delivering multicast to end users. In Proceedings of IEEE Infocom, June 2002.

[9] 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.

[10] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable application layer multicast. In Proceedings of ACM Sigcomm, August 2002.

[11] 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.