

패칭(Patching) 기법과 호스트 멀티캐스트(Host Multicast)트리 구성을 활용한 효율적인 VOD(Video On Demand) 시스템의 성능 향상 기법 연구

곽용완*, 남지승*, 이형옥*, 양현중*

*전남대학교 전자컴퓨터정보통신공학과

e-mail : yongwan38@empal.com, jsnam@chonnam.ac.kr, narcis@freechal.com, withyunjong@naver.com

A Research of effective Video on Demand System performance using Patching and Host Multicast Tree construction

Yong-Wan Gwak*, Ji-Seung Nam*, Hyeong Ok Lee*, Hyun-Jong Yang*

*Dept of Computer Science, Chonnam University

요 약

최근 그룹 통신 분야에서 전통적으로 대표되는 IP 멀티캐스트의 대안기술로서 호스트기반 멀티캐스트 및 P2P시스템은 기존의 IP 멀티캐스트의 단점을 극복하고자 하는 연구이다. 이러한 호스트기반 멀티캐스트는 사용자 중심의 서비스를 제공함으로써 다양한 네트워크 환경에서 누구나 사용자들이 효율적으로 데이터를 공유하게 해주고 있다. 본 논문에서는 이러한 협동 기반의 호스트 멀티캐스트를 사용하여 미디어 서비스를 제공하는 VOD 시스템 전반에 관한 연구를 수행하고 이러한 VOD의 서비스를 고도화 하고자 한다. 보다 구체적으로는 패칭 기법을 사용하여 사용자간의 전달 지연을 최소화 할 수 있는 효율적인 자료 전송 경로를 설정 방법과 주문형 비디오 서버를 위한 현실적인 호스트 기반 멀티캐스트 트리 구성 모델을 제시하고자 한다.

1. 서론

네트워크 환경의 발전속도는 가히 경이적이라 할 수 있을 만큼 빠르게 변화를 하고 있다. 이에 따른 사용자 환경도 계속 발전하고 있으며 고품질의 미디어 콘텐츠가 네트워크 서비스에서 가장 중요한 이용 목적이 되고 있다. 인터넷을 통한 멀티미디어 콘텐츠의 서비스는 그간 네트워크 자원의 효율적 활용 및 IP 네트워크 본연의 구조에 합당한 성능개선이 목표였다.

최근 그룹 통신 분야에서 새롭게 급부상한 호스트 기반 멀티캐스트 및 P2P는 전통적인 IP 멀티캐스트의 대안 기술로써 뿐만 아니라 다양하고 어느(Any) 사용자들의 자원공유를 통한 고품질의 멀티미디어 콘텐츠 서비스를 가능하게 해주는 가능성을 보여주고 있다.

일반적인 VOD서비스의 경우 사용빈도가 높은 콘텐츠는 필연적으로 서버 측의 부하를 가져오게 된다. 이에 대한 해결책으로 수년 전까지 IP 멀티캐스트가 적절한 메커니즘으로 여겨져 왔으나 IP 멀티캐스트는 라우터의 구현, 혼잡 제어와 신뢰성 있는 전송에서 여러 가지 문제점이 발생하였으며 이에 대한 대안으로 호스트(Host) 멀티캐스트

트 방법이 제안 되었다.[7] 본 논문에서는 호스트 기반 멀티캐스트를 이용한 트리구성을 통하여 기존의 IP 멀티캐스트의 단점을 극복하고 최상의 트리 구성을 통하여 보다 많은 사용자에게 서비스를 제공하고자 한다. 또한 VOD 서비스를 수행하기 위해 수신 호스트가 재생 데이터를 일정시간 동안 다른 호스트를 위해 공유 버퍼에 보관 하게 하여 새로운 호스트의 서비스 요청 시 호스트 노드들의 서비스 상태를 검토하여 최상의 서비스가 가능한 VOD 시스템을 구성하는 기법을 제시한다. 이러한 방법은 기존의 유니캐스트 서비스를 사용하는 시스템에 비해 특별한 기능적 제약이나 서버의 부하없이 멀티캐스트 서비스를 수행하여 VOD 서비스의 성능을 향상시킬 수 있다.

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

2. 관련연구

VOD 서비스는 On-Demand가 의미하듯이 사용자 지

향적인 서비스이다. 실시간 서비스와는 달리 사용자의 요구에 따른 서비스가 진행되어야 하므로 VOD 서비스에 멀티캐스트를 적용하기 위해서는 사용자 서비스 스트림을 병합시켜야 하는 문제가 발생한다. 멀티캐스트에서 새로운 클라이언트가 접속 하였을 때, VOD 서비스에서는 미디어의 시작부분부터 접속한 시간까지의 모든 스트림이 필요하다. 따라서 미디어의 시작 부분을 전송해줄 수 있는 특별한 기법이 필요하며, 이를 패칭(Patching)기법 이라고 한다[5]. 일반적인 패칭기법은 전체 비디오를 멀티캐스트 하는게 아니라 시간차에 의해서 채널의 빠진 부분을 채우는 기법, 패칭이전에 멀티캐스트 스트림의 공유를 위해 사용자의 대기를 전제로 성능을 높이는 배치기법, 피기백킹을 통하여 동일한 객체에 대한 입출력 스트림이 하나로 병합될때까지 진행중인 스트림의 영상 디스플레이율 조정하는 기법등이 있다. 호스트 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 [5]. 호스트기반 멀티캐스트에서 VOD 서비스를 하는 방안에 대한 연구를 살펴보면 다음과 같다.

P2Cast[1]는 P2P네트워크와 IP 멀티캐스트의 패칭채널, 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 시작부분을 캐쉬에 저장하고 있는 다른 클라이언트들로부터 수신 받게 된다. 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. 두 번째로 P2Vod[2]는 IP 주소의 부족, 보안 및 보급상의 문제로 실현 불가능한 IP 멀티캐스트에 대한 대안으로 제안된 호스트 멀티캐스트를 이용하여 VOD서비스를 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 저장하기위한 특정 크기의 버퍼를 구성하게 된다. 이러한 그룹화는 장애가 발생 했을 때 복구에는 효과적일 수 있지만 메커니즘의 구현이 복잡해질수 있는 단점이 있다.

3. 제안하는 호스트 멀티캐스트 트리 구성

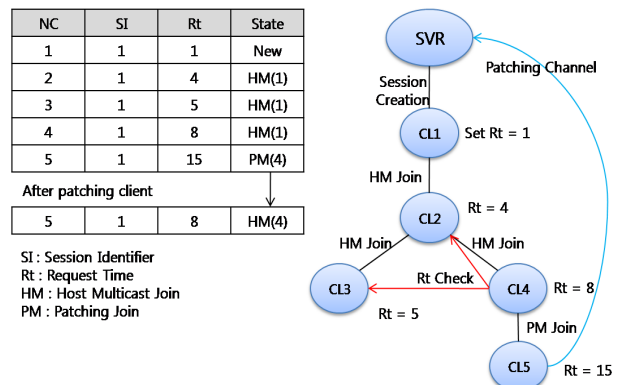
3-1. 개요

본 논문에서 제안하는 호스트 기반 멀티캐스트 하에서 VOD 시스템은 일반적인 On-Demand 서버가 제공하는

미디어의 서비스 목록과 특성, 서비스 요청시간 등을 세션 정보에 이용하고 세션정보에 있어서 유니캐스트 방식과 다른점은 요청 콘텐츠에 대한 논리적인 전송 경로가 포함 되어져 있다. 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 버퍼를가지고 있으며, 이 공유 버퍼는 부모 노드나 최상위 VOD 서버로부터 수신된 스트림을 저장하고 그 스트림을 자식 노드에게 전달하는데 이용된다. 호스트 멀티캐스트 기법은 효율적으로 VOD서비스를 수행하기 위한 전송 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

3-2. 패칭을 활용한 On-Demand 시스템 호스트 멀티캐스트 트리 구성 알고리즘

본 논문에서 제안한 시스템은 호스트 기반 멀티캐스트를 사용하여 사용자 공유 버퍼를 통하여 사용자 간에 미디어를 전송하도록 하고 패칭기법을 통하여 멀티캐스트 스트림에 가입되는 멤버수를 극대화 한다. 따라서 기본적으로 발생하는 사용자들간의 데이터를 공유시키고 이에 대한 제어정보를 서버에 저장함으로써 유연한 시스템 구조를 가지도록 구성하였다. 사용자 트리 구성은 사용자 서비스가 최우선적으로 고려되었고 그 다음으로 물리적인 네트워크에서의 최적의 망 사용 효율을 가질 수 있도록 호스트 기반 멀티캐스트 트리를 구성하였다.



(그림 1) 세션테이블 및 호스트 멀티캐스트 트리 구성

그림 1은 트리의 형성과정을 전체적으로 도식화 한 그림이다. 클라이언트의 공유버퍼의 크기가 5일 때, 15 요청 시점에 새로운 클라이언트 CL5가 가입을 요청하면 호스

트 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.

본 논문에서는 호스트 멀티캐스트를 트리를 구성하고 새로운 노드 가입의 경우 3가지의 형태로 구분하였다. 먼저 새로운 노드는 VOD서버에 세션 가입 요청을 하고 요청을 받은 VOD서버는 세션테이블을 검사하여 새로운 노드의 Join Type과 부모 노드가 될수 있는 클라이언트의 리스트를 전송한다. 새로운 클라이언트는 리스트를 이용하여 부모노드를 선택하고 세션을 설정한다.



(그림 2) Join Procedure

1) HM join (Host Multicast Join)

HM join형식으로 가입된 클라이언트는 오직 다른 클라이언트에 의해서만 스트림을 수신 받는다. HM 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. 먼저 충분한 대역폭을 가지는 부모노드가 있는지 검사하고 이러한 대역폭을 가지지 못하는 부모노드는 후보에서 제외한다. 두 번째로 대역폭이 충분한 노드들 가운데 새롭게 가입한 클라이언트와의 RTT가 최소값인 클라이언트를 부모노드로 선택한다. 만일 위의 조건을 만족하는 호스트가 없다면 PM join 방식을 수행한다.

2) PM join (Patching Join)

PM join형식으로 가입된 클라이언트는 다른 클라이언트와 VOD서버에게서 동시에 스트림을 수신 받는다. 이러한 PM 상태는 VOD서버로부터 패칭 스트림의 수신이 완료 되면 VOD서버와의 연결은 종료되고 가입된 클라이언트는 HM join 상태로 변경 된다. PM join 역시 HM join과 마찬가지로 노드의 대역폭을 고려하고 다수의 부모노드 후보가 존재할 때 RTT가 최소인 클라이언트를 부모노드로 결정한다.

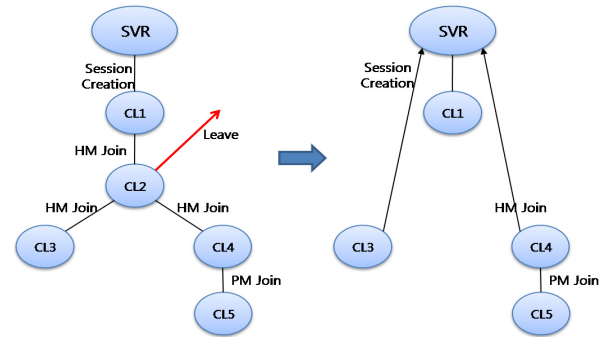
3) VM join (VOD Server Join)

새로운 클라이언트가 위의 두 join 조건을 만족하는 부모노드를 찾지 못하면 새로운 클라이언트는 VOD 서버와의 직접 연결을 생성하고 VOD서버로부터 비디오 스트림을 수신 받는다.

3-3. 노드 이탈시 트리 재구성 방안

호스트 기반 멀티캐스트의 경우 중간 노드의 이탈시 큰 문제가 발생하게 된다. 따라서 이러한 문제점을 해결하기 위하여 이러한 장애가 발생했을 때 세션을 빠르게 복구하기 위한 메커니즘이 필요하다.

그림 3은 호스트 멀티캐스트 트리가 구성된 상태에서 CL2 노드가 이탈시에의 복구 과정을 보여주고 있다. CL2 노드의 이탈시에 CL2의 자식노드인 CL3과 CL4는 새로운 세션을 생성하기 위하여 VOD서버에 세션 설정을위한 세션 테이블 정보를 요청하고 이 테이블을 통하여 위에서 언급한 join 과정을 되풀이 하여 새로운 세션을 설정한다. 이는 간단하면서도 장애 복구에 소요되는 시간이 적고 오버헤드를 최소화 시킬 수 있는 효율적인 방법이라고 할 수 있다.



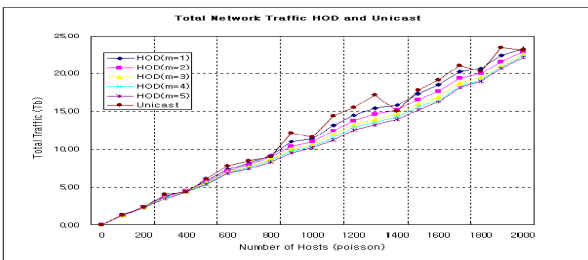
(그림 3) 노드 이탈시 트리 재구성 방안

4. 시뮬레이션 및 성능평가

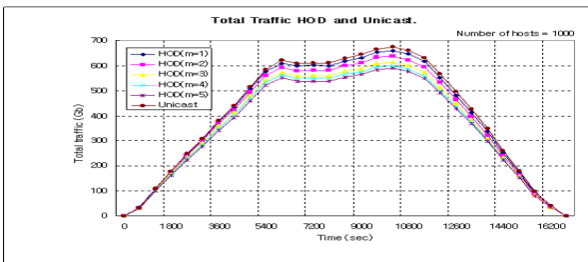
제안된 모델의 성능을 평가하기 위해 GT-ITM을 이용하여 생성한 서로 다른 다섯 개의 Transit-Stub 네트워크 토폴로지를 사용하였다. Transit-Stub 모델은 상위 레벨인 Transit영역과 하위 레벨인 Stub영역이 연결된 2 계층으로 이루어져 있다. Transit영역끼리의 연결 대역폭을 50Mbps부터 100Mbps까지, Transit영역과 Stub영역의 연결 대역폭을 10Mbps부터 50Mbps까지, Stub영역끼리의 연결 대역폭을 5Mbps부터 10Mbps까지 변화 시켜보았고 LAN상의 호스트 수에는 제한을 두지 않고 성능을 평가하였다. 또한 시뮬레이션을 위해 각각 다른 재생률을 갖는 1시간 30분(5400초) 길이의 10가지의 비디오 타입을사용하였다. 각 호스트는 비디오 ID, 서비스 요청시간을 포함하고 있으며, 이러한 호스트의 수를 100명부터 2000명까지의 증가 시켜가며 성능을 평가하였다.

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 비교하였다.

그림 4(a)는 클라이언트의 수에 따라 전체 네트워크 트래픽이 변화되는 모습을 보여준다. 전체적으로, 본 논문에서 제안한 방법이 유니캐스트를 사용한 방법에 비해 큰 성능 향상이 없는 이유는 서비스를 요청한 클라이언트들의 불규칙적인 위치 및 시간과 관련이 있다. 본 논문에서 사용한 방법에서 공유 버퍼 크기가 증가할수록 트래픽은 감소하고 있다는 사실을 알수 있는데 이는, 공유 버퍼가 클수록 클라이언트간의 연결이 PM join 보다는 HM join 형태를 취하기 때문에 나타날 수 있는 현상이다. 그러나 공유버퍼의 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.

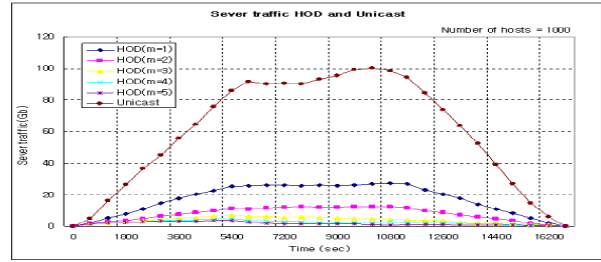


(그림 4(a)) Host Number vs Total Network Traffic



(그림 4(b)) Time vs Total Network Traffic

그림 5는 시간의 흐름에 따른 VOD서버상의 네트워크 트래픽 변화량이다. 유니캐스트를 사용한 방법에서는 모든 사용자들이 서버에 직접 연결되어 서비스를 받기 때문에 서버측의 트래픽은 그림 4(b)의 전체 네트워크 트래픽과 유사한 모습을 보이고 있다. 본 논문의 제안기법을 사용한 경우, 공유버퍼의 크기에 따라 다소 차이는 있지만 대체적으로 유니캐스트를 사용한 방법에 비해 서버측의 트래픽이 상당히 감소되었으며 그림 5의 결과로 볼 때, 유니캐스트를 이용한 경우보다 월등히 많은 사용자를 수용할 수 있음을 알 수 있었다. 또한 공유버퍼의 크기를 3분 이상으로 설정하는 것은 성능향상에 크게 영향을 미치지 않았다.



(그림 4(b)) VOD서버 Bandwidth

4. 결론

본 논문에서는 호스트 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. 시뮬레이션을 통해서 기존의 유니캐스트 방법과 제안된 방법의 전체 네트워크와 서버측의 트래픽을 비교한 결과 본 논문에서 제안한 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.

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
- [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.
- [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] Y. Okada, M. Oguro, J. Katto, S. Okubo, "A New Approach for the Construction of ALM Trees using Layered Video Coding", in Proceedings of ACM Multimedia 2005.
- [7] De-Nian Yang, Wanjiun Liao "On Bandwidth-Efficient Overlay Multicast" Parallel and Distributed Systems, IEEE Transactions on Volume 18, Issue 11, Nov. 2007.