

P2Patching : 주문형 P2P 서비스를 위한 효율적인 패칭 기법

정희원 김 중 경*, 이 재 혁*, 박 승 규*

P2Patching : Effective Patching Scheme for On-Demand P2P Services

Jong-gyung Kim*, Jae-hyuk Lee**, Seung-kyu Park*** *Regular Members*

요 약

본 연구는 P2P 환경에서 패칭을 이용하여 주문형 스트림을 효율적으로 서비스하기 위한 응용 계층 멀티캐스트 기법인 P2Patching을 제안한다. P2Patching은 전통적인 패칭 기법을 응용 계층 멀티캐스트 기법에 확장 적용하여 서버의 부하를 감소시키고 초기 지연을 최소화하는 멀티캐스트 트리의 구축 기법이다. 그리고 트리의 장애로부터 빠른 복구 기법과 잦은 트리의 단절로 인하여 연속적인 재생을 보장하기 위한 동적 버퍼링 기법도 제안한다. 또한, 본 논문과 유사한 P2CAST^[12]와의 몇 가지 중요한 실험에서 향상된 결과를 보여준다. 실험 결과에서는 평균 조인 횟수, 서비스 요청 거부율, 트리 복원 실패율 그리고, 버퍼 스타베이션(Buffer Starvation)이 발생한 노드 수 등의 측정에서 약 16% 정도의 성능 향상을 보였으며, 특히, 버퍼 스타베이션과 평균 조인횟수에 대한 실험에 대해서는 우수한 성능 향상을 보였다.

Key Words : Video on Demand, Patching, Peer-to-Peer Networking, Multicasting

ABSTRACT

In this paper, we propose a multicast P2Patching technique in the application layer. The P2Patching technique serves VOD stream effectively with Patching in P2P environment. The P2Patching provides multicast tree construction technique that reduces the server load and minimizes the start delay with extended multicast technique. And we provide a fast recovery technique by tree failure and dynamic buffering scheme that guarantees the continuous streaming by frequent tree disconnections. Comparing the method with that of P2Cast^[12], we obtained the better performance by our scheme in terms of average join count, service rejection probability, tree recovery failure and buffer starvation. The average about 16% of the improvement is shown by the simulation. Most of all, the performance of buffer starvation and average join count shows a significant improvement than that of P2Cast.

I. 서 론

인터넷의 인기가 날로 상승함에 따라 초기의 WWW는 텍스트 기반의 하이퍼링크(Hyperlink)에 기반을

두었으나 오늘날에는 오디오, 비디오 등을 혼합한 멀티미디어 데이터로 표현하는 형태로 발전하고 있다. 이러한 콘텐츠 형태의 변화는 사용자의 다양한 요구를 수용하는 동기라고 볼 수 있다. 특히, 고속

* 아주대학교 일반대학원 (jgkimpower@nate.com)

논문번호 : KICS2005-11-484, 접수일자 : 2005년 11월 29일

네트워크 환경에서 VOD와 같은 비디오 콘텐츠의 인기는 날로 늘어나고 있는 실정이다.

그러나 이러한 VOD 서비스는 VOD 서버가 다수의 클라이언트들이 대역폭의 절약하는 방식으로 하나의 스트리밍을 공유하는 개념의 멀티캐스트 기법을 사용한다. 멀티캐스트 기법의 한 유형인 Patching^[1], Batching^[2] 그리고 Piggybacking^[3] 등의 IP 기반의 VOD 멀티캐스트 기법들은 특정 사용자 행위나 사용자 인지의 유형을 만들어 VOD 스트림들을 전송하는 것으로서, 이 기법의 활용에는 몇 가지의 문제점이 발생한다^[5, 6]. 이 문제점들을 해결하기 위한 대안으로서 많은 논문들이 응용 계층 멀티캐스트(Application layer multicast)를 활용하는 기법들을 제안하였다^[7~10].

이들의 특징은 첫째, 라이브 스트림(live stream)과 같이 동시에 사용자들이 집중될 때 다수의 사용자들이 하나의 스트림을 공유하여 멀티캐스트 트리를 구성하는 방식이다. 둘째, 이 연구들은 인터넷에서 스트림 배송(delivery)을 다루기 위한 기법들이다. 따라서 VOD 서비스에서는 대역폭(bandwidth), 지연(latency) 그리고 지터(Jitter) 현상이 발생할 가능성이 높다. VOD와 같이 사용자의 요구가 간헐적으로 발생하거나 다양한 스트리밍 서비스를 요구하는 경우에 적용하기에 무리가 따른다.

따라서 이러한 문제들을 해결하기 위한 방법으로 배분(distribution) 기법, 비디오 스트림의 버퍼링(buffering), 네트워크 환경을 고려한 스트리밍 메커니즘 등의 다양한 기법들의 협력이 요구된다.

본 논문에서 P2P 환경에서 효과적인 VOD 스트리밍 서비스를 위한 방법으로 P2P 개념과 패칭 기법(patching technique)을 혼용하여 자원을 효율적으로 활용하기 위한 P2Patching 기법을 제안한다. P2Patching은 전통적인 패칭 기법을 응용 계층에서 구현하는 것으로서 일정시간 동안에 발생하는 사용자들이 하나의 세션(session)을 형성하고 각각의 세션에 속해 있는 클라이언트들은 응용 계층 멀티캐스트 트리로 구성된다. 비디오 서비스를 받기 위해 클라이언트들은 일정한 시간에 해당하는 비디오 스트림의 초기부분을 버퍼링한다. 어떤 클라이언트가 멀티캐스트 트리로부터 장애를 당할 경우 버퍼로부터 저장된 비디오 스트림을 정상적으로 재생하고 두 스트림 즉, 정규 스트림과 패칭 스트림을 다른 피어들이나 혹은 서버에 연결함으로써 해당 세션에 재연결(rejoin)을 한다. 여기서 패칭 스트림은 멀티캐스트 트리로부터 단절되는 동안에 수신하지 못한

스트림을 받고, 정규 스트림은 멀티캐스트 스트림을 받기 위해 사용된다. P2P 환경에서 VOD 서비스를 구현하기 위하여 본 P2Patching에서는 다음 기법들을 다룬다.

1.1 효율적인 응용 레벨 트리의 구축 기법

트리의 높이는 낮고 넓게 유지해야 하고 조인 프로세스는 빠르게 실행되어야 한다^[8]. 이는 대역폭 소비를 줄이고 장애 발생으로부터 서비스 중단 가능성을 줄인다. 따라서 대역폭이 가장 많이 존재할 리프 노드(leaf node)를 우선으로 조인하고, 빠른 장애 복원이나 재연결을 보장한다.

1.2 비디오 스트림의 연속성 보장

멀티캐스트 트리 상에서 연속적으로 발생하는 부모 피어들의 장애는 자식 클라이언트들의 지터와 같은 QOS 저하현상을 초래하고, 더 나아가 VOD 특성상 시청자들의 시청을 포기하는 확률이 높아진다^[2]. 이를 개선하기 위하여 비디오 스트림의 초기 부분뿐만 아니라 부모 노드의 장애로 인하여 분실된 비디오 스트림을 수신하기위해 패칭 기법을 확장 사용하고, 부드러운 재생을 위하여 버퍼에 일정량의 재생할 비디오 스트림을 유지하도록 하는 동적 버퍼링 기법을 제안한다.

이 논문의 구성은 제1장에서는 전체적인 소개를 하였으며, 제2장에서 관련연구, 제3장에서 P2Patching 기법의 개요와 멀티캐스트 트리 구축 기법, buffering 기법 그리고 failure recovery에 대하여 기술하고, 제4장에서는 P2Patching과 P2Cast^[12]와의 비교한 실험을 보여주고 제5장에서 결론을 내린다.

II. 관련 연구

2.1 IP 기반 멀티캐스트 기법

이 기법들은 동일한 비디오에 대하여 인접한 시간에 발생한 요청들에 대한 서비스를 위해서 일부의 서버 채널을 사용하는 기법으로 서버의 자원과 대역폭을 공유하는 개념이다. 서버가 하나의 서버 채널에 짧은 시간대에 발생하는 요청을 "Batch" 단위로 묶어 스트림을 전송하는 개념의 Batching^[2]과 정적 멀티캐스트 개념을 확장하는 기법으로서 조금 늦게 도착하는 클라이언트들을 "Batch"에 같이 묶어 멀티캐스트 서비스를 지원하는 Adaptive Piggybacking^[3], Patching^[1] 등이 있다. 그리고 사용자가 서버에 비디오 스트림에 관한 요청을 하지 않고 서버는 같은 비디오를 일정시간마다 주기적으로 방송

하는 Pyramid Broadcasting^[4]이 있다.

이는 디스플레이 속도와 대기 시간에 대해 사용자가 인지할 수 있는 한계성을 이용하는 방법이라 할 수 있다. 이 기법들은 현실에 적용하기에는 몇 가지의 문제들이 발생하여 다수의 논문들이 응용 계층에서 멀티캐스트 서비스를 할 수 있는 기법들을 제안하였다^[5, 6].

2.2 응용 계층 멀티캐스트(Application layer Multicast)

응용계층 멀티캐스트 기법은 종단 시스템에서 P2P 기법의 멀티캐스트 트리를 구성하는 방식이다. Coopnet^[10]은 실시간 비디오 스트리밍을 위한 시스템으로서 Multiple Description Coding을 적용하여 피어의 장애가 발생하더라도 데이터의 손실을 최소화한다. 이는 VOD를 위한 기능은 지원하지만 재생점이 다른 사용자들을 관리하는 기법은 제공되지 않는다.

NICE^[13]와 ZIGZAG^[7]는 계층 구조의 클러스터 기반 기법으로서 모든 멤버들을 확장성 있는 계층형 구조로 형성하고, 동적 특성들을 소화할 뿐만 아니라 멀티캐스트 트리의 제어 트래픽을 감소시킬 수 있으나 네트워크의 유동적인 환경에서 개념적인 대역폭을 다루는 이러한 기법들은 현 인터넷에서 적용하기엔 무리가 따른다. 특히 VOD 스트리밍 서비스에는 이런 구조로는 해결될 수 없다.

P2Cast^[12]는 본 논문과 유사한 논문으로서 패칭을 사용하여 VoD 서비스 설계에 초점을 둔 응용계층 멀티캐스트 기법이다. 이 논문은 장애가 발생할 때마다 서버가 개입하여 해결하는 방식을 사용함으로써 서버의 병목현상 때문에 문제가 발생할 소지가 크다.

VOD 특성상 아주 빈번하게 조인(join)과 이탈(leaving) 같은 이벤트들이 발생함은 물론이고 예기치 않은 네트워크 상황으로 인하여 화질의 저하와 멀티캐스트 트리의 단절이 발생할 가능성이 높아 이로 인한 VOD 스트리밍 서비스의 영향과 문제점 그리고 대안을 연구할 필요성이 제기된다.

III. P2Patching

3.1 P2Patching overview

P2Patching는 P2P 환경에서 응용 계층 멀티캐스트와 패칭 기법을 혼용 적용한 것으로서 비디오 스트림을 원하는 클라이언트는 멀티캐스트 트리에 조

인하여 VOD 서버나 다른 피어들로부터 요청한 비디오 데이터를 수신하고 버퍼링하며, 버퍼링한 데이터를 다른 클라이언트에게 제공하여 시스템에 기여함으로써 서버의 이용률(utilization)을 감소시키고 시스템의 확장성을 극대화하려는 기법이다.

본 논문에서 P2Patching의 멀티캐스트 트리는 한 세션 단위로 구성한다. 이 세션의 개체들은 비디오 스트림들이 저장되어 있는 VOD 서버와 비디오 디스크, 세션을 처음 생성한 피어 즉, 세션 루트(root)와 비디오 스트림 수신 및 재생하는 클라이언트, 그리고 수신된 비디오 스트림을 다른 클라이언트들에게 제공하는 피어들로 구성되며 이를 도해하면 그림 1과 같다.

일반적으로 응용 계층 멀티캐스트에서 장애로 인하여 트리가 단절되는 경우에 자연스러운 비디오 스트림의 재생을 어렵게 만든다. 따라서 이를 개선하기 위하여 본 논문에서는 확장된 패칭 기법을 제안한다. 확장된 패칭 기법은 다음의 세 가지 경우에 사용된다. 첫째, 멀티캐스트 트리의 단절로부터 연속적인 재생을 위하여 초기 버퍼링이 필요하다.

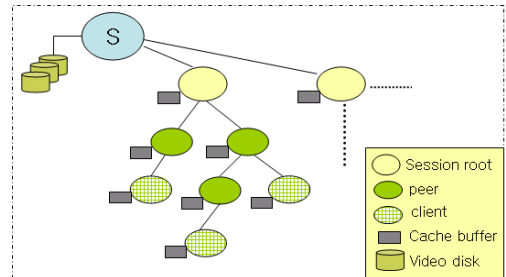


그림 1. P2Patching multicast system 개요

이때, 세션 루트인 피어는 초기 대기시간을 최소화하기 위하여 조인 초기에 패칭 스트림을 사용한다. 둘째, 부모 피어들의 장애로 인한 멀티캐스트 트리의 단절에서 재연결된 피어가 장애동안에 손실된 비디오 스트림들을 수신하기 위하여 피칭 스트림을 사용한다. 셋째, 가장 최근에 출발한 세션이 시작된 직후 도착한 클라이언트들은 세션의 출발시간부터 세션에 조인된 시간까지의 미수신 부분에 대한 스트림을 수신하기 위하여 패칭 스트림을 사용한다.

일반적으로 패칭 스트림을 받는 동안에는 실제로 정규 스트림 전송물의 두 배로 데이터를 받는다. 이 패칭 스트림의 수신이 완료되면 이 패칭 스트림은 해지가 되고 그 클라이언트는 오직 정규 스트림만 수신하게 된다. 본 논문과 달리 P2Cast^[12]에서는 위

에서 언급된 세 번째 경우만 적용하였다.

3.2 P2patching Basic Operation

일반적으로 패칭 기법에서는 재생의 질을 보장하기 위하여 일정시간동안에 버퍼링 후 재생을 시작한다. 특히 P2cast^[12]에서는 비디오 길이의 $\Delta_m + \tau$ (Δ_m : patching size, τ : playback delay)이상의 스트림을 미리 버퍼링 후에 재생하는 “shifted forwarding technique”을 제안하여 연속적인 재생을 보장하였다. 그러나 이것은 긴 시간동안의 버퍼링에서 오는 사용자 이반율(defection rate)을 가중시킬 뿐만 아니라, 계속해서 부모 노드의 장애로부터 멀티캐스트 트리의 단절이 되어 재생의 질을 보장하지 못하는 문제점을 발생하여 개선할 필요가 있다.

3.2.1 New Client Join

P2Patching에 멀티캐스트 트리에 조인하려고 하는 클라이언트는 두 가지 경우가 발생한다. 하나는 세션을 개설하는 세션 루트이고 또 하나는 이미 개설된 세션으로 조인되는 경우이다. 전자의 경우는 서버로부터 직접 정규 스트림과 패칭 스트림을 받고 후자의 경우는 비디오 서비스를 받기 위해서 임계시간 이전에 도착한 피어들로 구성된 세션 중에서 현재의 서비스 요청 시간과 가장 가까운 세션에 조인하여 정규 스트림과 패칭 스트림을 제공해 줄 피어들을 선택해야 한다. 만일 정규 스트림을 제공해 줄 피어를 찾지 못하면 서버를 통하여 정규 스트림과 패칭 스트림을 전송 받을 수 있다. 정규 스트림은 로컬 윈도우 버퍼(Local Window Buffer)에 저장되고 패칭 스트림은 바로 재생을 시작한다. 한편 서버가 그 클라이언트에 대하여 서비스를 하지 못하는 경우에는 그 클라이언트의 서비스는 reject된다. 그림 2에서 멀티캐스트 트리의 예를 보면, 임계시간이 “5”인 경우 두 개의 세션이 형성되어 있다. 세션 1에서는 피어 X가 시간 5.0에 도착하여 피어 Y와 같이 하나의 세션을 형성하여 서버로부터 스트림 서비스를 받고 있고, 그 이후에 형성된 세션 2에서는 피어 A가 시간 11.0에 도착을 하였기 때문에 패칭 스트림 길이(11.0-5.0=6.0)가 임계시간(5)보다 길어 피어 B,C,D와 같이 다른 세션을 구성한다. 클라이언트 B는 피어 C에서 멀티캐스트 스트림을 받는 동시에 피어 A로부터 패칭 스트림을 전송받고 있는 것을 보여주고 있다. 원 부근에 표시된 숫자는 피어들이 도착한 시간을 의미하고 실선과 점선은 멀티캐스트 스트림과 패칭 스트림을 의미한다.

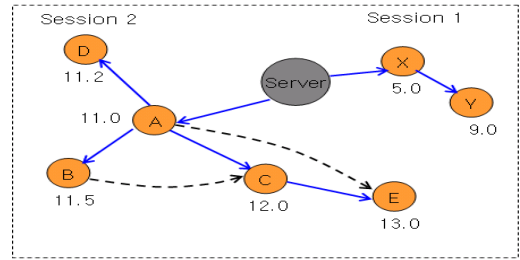


그림 2. P2Patching에서의 멀티캐스트트리의 예

3.2.2 Tree disconnection and rejoin

부모 노드의 장애로 인하여 멀티캐스트 트리의 단절이 되는 경우는 새로운 부모 피어를 선택해야 한다. 멀티캐스트 트리에서 정규 스트림을 제공해 줄 새로운 부모 피어와 멀티캐스트 트리의 단절시간에서 재연결 시간까지의 손실된 부분 즉, 패칭 스트림을 제공할 수 있는 피어를 선택하여야 한다. 이에 대한 상세한 내용은 3.3절에서 기술한다.

3.3 Multicast Tree Construction

3.3.1 Regular Stream

비디오 스트림을 수신하기 위하여 P2Patching의 클라이언트들은 멀티캐스트 세션에 조인해야만 한다. 최근에 생성된 멀티캐스트 세션이 존재하지 않는 경우 VOD 서버는 새로운 세션을 생성하여 그 클라이언트에게 정규 스트림을 전송하지만 생성된 세션이 존재하는 경우에는 멀티캐스트 트리에서 비디오 스트림을 전송해 줄 부모 피어를 선택한다. 이러한 시스템 목적에 부합하는 배분 트리를 구축하기 위한 비용 및 지연 등과 같이 여러 측정 요소를 동시에 최소화하는 것은 NP-complete 유형의 Steiner Tree problem이 알려져 있다. 따라서 여러 피어들에게 동시에 스트림을 배분하기 위해서는 효율적인 멀티캐스트 트리의 구축할 필요성이 제기된다.

본 P2Patching 멀티캐스트 트리는 VOD 특성을 고려하여 대역폭 사용을 가장 우선적으로, 조인(join) 지연을 차선으로 선택하여 구축하는 기법을 사용한다. P2P 환경에서 피어들의 대부분은 멀티캐스트 트리의 리프(leaf) 노드로서 자신의 자원을 시스템에 기여하지 않는다^[8]. 따라서 기본적인 조인 방법에 있어서 리프 노드가 사용 가능한 대역폭이 존재할 가능성이 높다는데 착안을 한다. 본 논문에서 이러한 사실에 기초하여 멀티캐스트 트리의 조인 기법은 서버로부터 가장 홉(hop)수가 적은 하위 레벨(bottom)의 리프(leaf) 노드에 우선 접근한다. 리프 노드의 부모 대역폭이 새로운 자식 클라이언

```

Multicast-stream-join
/* L : sort leaf-node set in ascending order of level-number
/* N : new client
for each p ∈ L
    probe bandwidth of the p's parent
    if p's parent is enough bandwidth for N then
        N became p's sibling
        return(true)
    else
        probe bandwidth of the p
        if p has is enough bandwidth for N then
            N became p's child
            return (true)
        end if
    end if
end for
return(false)
    
```

그림 3. 노드 조인 알고리즘

트를 받아들이기에는 충분한 경우는 자식 노드로 조인하고, 그렇지 않을 경우는 리프 노드의 자식 노드로 조인하도록 하는 기법을 사용하며 이에 대한 알고리즘은 그림 3과 같다.

3.3.2 Patching Server Selection

본 논문의 패칭 서버 선택 알고리즘은 다음과 같이 실행된다. 이는 대역폭 소비를 최소화하고 프로세스 오버헤드를 줄이기 위하여 피어가 접속해 있는 멀티캐스트 트리에서 가장 가까운 위치에 있는 피어를 선택하고, 이를 위해서 현 위치에서 한 홉씩 bottom-up 방식으로 프로세스를 실행시킨다.

그림 4는 P2Patching에서 패칭 서버를 선택하는 예를 보여주고 있다. 그림 4(a)와 같이 빗금 친 노드 F가 멀티캐스트 트리에 조인하였다고 가정하자. 노드 F는 패칭 서버를 선택하기 위하여 자신의 부모 피어 C가 패칭 스트림을 제공할 수 있으면 부모 피어로부터 패칭 스트림을 제공받고 그렇지 않으면 그림 4(b)와 같이 부모 피어로부터 한 홉 떨어진 형제 피어들 즉, E, D중에서 대역폭이 충분한 피어를 패칭 서버로 선택한다. 만약 이들 중에서도 대역폭이 충분한 피어가 존재하지 않으면 그림 4(c)에서 보는 바와 같이 피어 C의 부모 피어 A를 선택한다. 그리고 부모 피어 C부터 두 홉 떨어진 노드(여기서는 피어 B)들을 차례로 여분의 대역폭을 가진 피어를 패칭 서버로 선택한다. 위 그림에서 직선은 정규 스트림의 경로이고 점선은 패칭 스트림의 경로이다. 그리고 이에 대한 알고리즘은 그림 5와 같이 도해하였다.

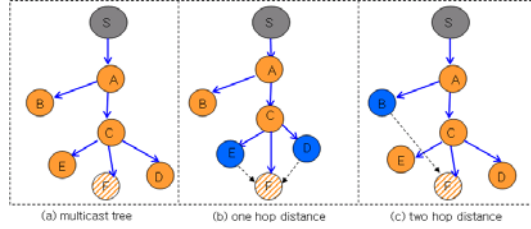


그림 4. P2Patching에서 패칭 서버 선택의 예

```

Patching-stream-join
/* num : level number
/* Y : current node
/* X : Y's parent
/* C : Y's sibling set
num ← level number of N
num ← num - 1
Y ← N
while (level ≠ 0)
    X ← Y's parent
    if X is enough bandwidth for N's patching server then
        X became N's patching server
        return(true)
    end if
    for each c ∈ C
        probe bandwidth of c
        if c is enough bandwidth for N's patching server then
            c became N's patching server
            return(true)
        end if
    end for
    num ← num - 1
    Y ← Y's parent
end while
return(false)
    
```

그림 5. 패칭 서버 선택 알고리즘

3.4 패칭 스트림과 동적 버퍼링 기법

멀티캐스트 트리에 조인되어 있는 피어들은 빈번한 부모 피어의 장애가 발생한다면 연속적인 재생이 불가능해진다. 따라서 사용자의 화면은 깜박임(flickers), 멈춤(freezes), 그리고 장시간의 서비스 중단(long pauses)과 같은 현상이 발생한다. 이 경우에 멀티캐스트 트리로부터 고립된 피어들은 같은 세션에서 다른 부모 피어를 찾거나 가장 인접한 시간의 다른 세션으로 재연결하여 스트리밍 서비스를 받을 수 있다. 그러나 비디오 스트림 수신 중단된 시간과 새 부모 피어로 재연결된 시간만큼의 시간 차이 즉, 재연결 지연이 발생한 만큼 스트림의 손실을 가져온다. 각각의 트리의 단절로부터 복원 될 때까지의 시간을 동일하다고 가정할 때, 그림 6의 예는 연속적인 비디오 스트림을 제공하기 위해서 동적 버퍼링 기법에 대해서 설명한다. 비디오 스트림을 받기 위하여 새로 조인하는 클라이언트는 멀티

캐스트 스트림과 패칭 스트림을 동시에 받기 때문에 Δ_m 시간 내에 버퍼가 찬다. 그리고 첫 번째로 시간 t_1 에서 부모 피어의 장애로부터 멀티캐스트 트리가 단절되었고, 시간 t_2 에서 새로운 부모 피어로 재연결되었을 때 재연결 지연은 $d = t_2 - t_1 < \Delta_m$ 이다. 패칭 스트림을 통하여 rejoin 지연시 손실된 부분 d 를 복구할 수 있는 동안, 두 번째로 시간 t_3 에 멀티캐스트 트리의 단절이 발생하였다고 가정하자. 또한, 시간 t_4 다시 재연결되었다고 한다면 첫 번째 복구된 비디오 스트림의 양은 $r_1 = t_3 - t_2$ 이 되며, 두 번째의 손실된 스트림의 양을 포함하면 $(t_4 - t_3) + (d - r_1)$ 이 된다. 이와 같이 어떤 클라이언트가 i 번째 멀티캐스트 트리로부터 스트림 서비스가 단절이 된다면 총 손실된 비디오 스트림의 양은 식 (1)과 같다.

$$D = d + \sum_{i=1}^n (d - r_i) \quad (1)$$

비디오의 QOS을 보장하기 위해서는 식 (1)에서 $D < \Delta_m$ 가 성립되어야 한다. 만약에 $D \geq \Delta_m$ 가 성립되면 사용자가 느끼는 화질에 문제가 발생할 가능성이 높아진다. 이것을 개선하려면 버퍼량 Δ_m 을 증가시키면 된다. P2P 환경이 유동적인 상황이라면 Δ_m 의 길이를 늘이고 이와 반대로 안정적인 상황이라면 Δ_m 의 길이를 짧게 할 필요가 있다. P2Cast에서는 긴 시간 즉, $\Delta_m + \tau$ (Δ_m : patching size, τ : playback delay)의 버퍼링을 요구하여 멀티캐스트 트리의 단절로부터 오는 QOS 문제를 해결하는 기법을 제공한다. 이 방법은 P2P 환경을 고려할 때 빈번한 부모 피어들의 장애로 인하여 재생할 버퍼의 내용이 고갈되어 서비스의 질을 보장하기 힘든 문제가 발생할 수 있다. 따라서 본 논문에서 패칭 스트림을 이용한 동적 버퍼링 기법으로 이를 대처할 수 있다.

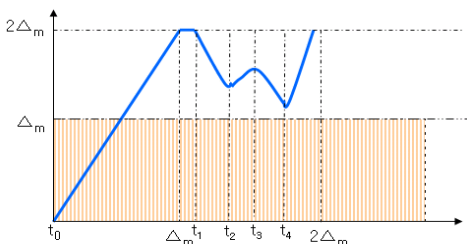


그림 6. 손실 비디오 스트림 수신 의 예

3.5 결함 복원(Failure Recovery)

P2P환경에서는 잦은 부모 피어 장애로 인하여 부모 피어를 잃은 클라이언트들을 VOD 서버가 개입하여 서비스를 한다. 특히 많은 클라이언트들에 장애가 발생한다면 서버의 부하는 급증할 것이다. P2Cast^[12]의 조인 알고리즘은 부모 노드의 장애가 발생할 때마다 서버가 개입함으로써 서버의 과부하가 발생하여 서비스의 Rejection될 가능성이 높아지고 또한, 긴 Block time이 발생한다. 본 논문에서는 이러한 문제들을 개선하기 위하여 트리의 장애가 발생할 때 서버를 통하지 않고 조부모 피어들이나 혹은 형제 피어들 중에서 가장 가까운 피어를 선택하는 알고리즘을 사용하며, 트리의 루프(loop) 발생 가능성을 배제하기 위해 착안한 방법이다.

3.5.1 recovery process

형제 피어들이나 조부모 피어들 중에서 가장 가까운 피어를 bottom-up 방식으로 새 부모 피어로 선택하는 것은 트리 비용을 감소시킬 뿐만 아니라 트리의 루프를 방지할 수 있는 방법이라 할 수 있다. 그림 8 (a)의 그림은 비디오 서버로부터 2개의 세션 즉, 시간 5.1에 시작되는 세션 1에는 피어 X, Y와 시간 11.0에 시작한 세션 2에는 피어 A, B, C, D가 정규 스트림을 받고 있고 현재시간 13.5에는 피어 E가 조인하였기 때문에 정규 스트림과 패칭 스트림을 동시에 받고 있다. 이 때 피어 C가 트리로부터 이탈을 했다고 가정할 때 피어 D, B는 멀티캐스트 트리의 단절을 복원하기 위하여 새로운 부모 피어를 찾아야만 한다.

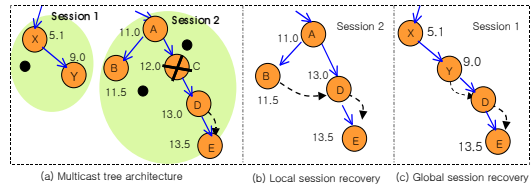


그림 7. 멀티캐스트트리에서 session recovery

그림 7의 (a)과 같이 부모 피어 C의 장애로 인하여 피어 D는 서브 트리의 루트 피어로서 같은 세션 안에서 형제 피어들이나 조부모 피어들 중에서 비디오 스트림을 지원 가능한 대역폭을 가진 피어들을 선택하는 장애 복원 프로세스를 실행한다. 따라서 서브 트리의 루트 피어 D가 복원 되면 간단하게 이 서브 트리 전체가 복원될 수 있다. 그림 7의 (b)의 도해는 D는 세션 2의 루트 피어 A로부터 정규

스트림을 그리고 피어 B로부터 패칭 스트림을 받고 있다. 한 피어가 다른 피어들에게 비디오 스트림을 제공할 수 있는 자원은 매우 제한적이다. 따라서 지역 복원 프로세스가 실패한 경우는 서버를 경유하여 고아가 된 피어가 속해있는 세션과 가장 가까운 시간에 형성된 세션의 피어들 중에서 하나의 부모 피어를 찾아 조인하는 것이다. 이는 다른 트리로 점프(jump) 기능을 통해 복원의 성공률을 높임으로써 트리의 신뢰성을 향상시키기 위한 방법 즉, 전역 복원 프로세스를 실행한다.

IV. 실험 및 고찰

본 P2Patching의 성능을 평가하기 위하여 다음의 측정 기준에 의거 시뮬레이션 기법을 사용한다.

- (1) 클라이언트가 스트림 서비스를 받기 위하여 서버에 조인을 하였을 때 Rejection 발생 확률과
- (2) Parent failure의 장애로 인하여 복원을 위한 Peer의 평균 접속 횟수 즉, Recovery 동안에 접속한 클라이언트 노드 수를 의미하는 것으로 얼마나 빨리 결함에 대해 복원하는 가를 측정하고, (3) 조인 성공을 위한 피어들의 평균 접속 횟수 그리고
- (4) 노드들의 평균 네트워크 사용량에 대한 정보와
- (5) 버퍼 스태이션이 발생하는 노드 수를 측정하는 실험으로 나눈다.

P2cast와 비교 실험을 위하여 동일한 환경으로 구성하고 시뮬레이션은 SSFNet^[11] 시뮬레이션 프레임워크를 사용하였으며, SSFNet은 프로세스 기반 이산 사건 중심 시뮬레이션 커널(Process-based Discrete Event-oriented Kernel)이다. 시뮬레이션 커널의 소스는 공개되지는 않았으나 그 중에서 네트워크의 시뮬레이션을 지원하는 라우터 링크 네트워크 인터페이스 카드 등 대부분의 인터넷 서브시스템들을 시뮬레이션 하는데 필요한 다양한 객체들이 Java로 시뮬레이션 특성을 변경하여 구현할 수 장점을 가지고 있다.

본 논문에서는 멀티 캐스트 시뮬레이션을 위해 적합한 시뮬레이션 환경을 구축하였다.

전체 네트워크는 총 100 노드들로 4 노드들로 구성이 된 하나의 Transit 네트워크와 12 Stub 도메인 구성하였다. 이 네트워크에서 각 노드는 하나의 클라이언트를 구성할 수 있으며, 비디오 스트림을 제공할 수 있는 충분한 대역폭을 가지고 있는 LAN(local area network)이다. 두 노드 사이의 라

우팅은 Shortest Path Algorithm을 사용하였으며 네트워크 대역폭 용량은 동시에 22개를 처리할 수 있는 backbone 링크와 7개의 비디오 스트림을 제공할 수 edge 링크로 설정하였다. 서버는 Transit 네트워크에 설치하고 서버에 저장된 비디오는 100분 길이의 하나의 비디오를 사용하는 것을 가정한다. 그리고 클라이언트의 도착율은 포아송 분포(Poisson distribution)에 따르며 새로 도착한 클라이언트는 네트워크 노드들 중에서 임의 위치에 조인시킨다. 멀티 캐스트 트리를 형성된 이후에 2분마다 하나의 클라이언트가 Connection Failure 발생하도록 하였다. 이것은 Failure 장애로 인한 재연결의 성능을 살펴 보기 위해 사용된다.

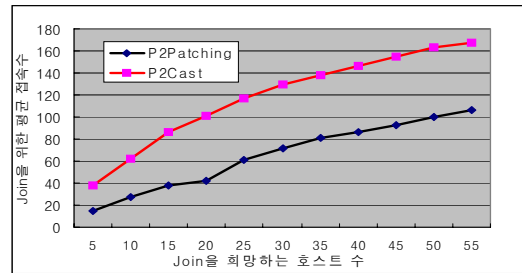


그림 8. Join 평균 접속 횟수

조인을 하려는 노드들의 증가에 따라 조인을 위한 노드들의 평균 접속 횟수를 측정한 결과를 그림 8에 나타내었다. P2Patching의 빠른 트리 탐색 알고리즘으로 인하여 P2Cast보다 적은 접속 횟수로 부모노드를 선택하는 결과를 보여준다. 이러한 적은 접속 횟수는 노드들에게 서비스를 보다 빠르게 제공할 뿐만 아니라 전체 네트워크 트래픽에 많은 영향을 미치게 된다.

그림 9는 서버로 서비스를 요구하는 클라이언트들의 요청 횟수에 따라 서비스 요청 거부 비율을 나타내는 그래프이다.

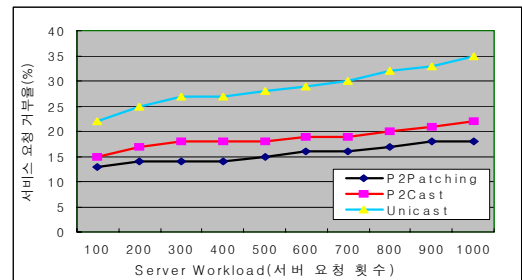


그림 9. 서비스 요청 거부율

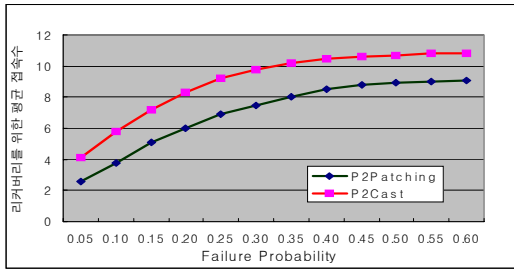


그림 10. Failure Probability

이 그림에서 보면, 유니캐스트 기법은 클라이언트들이 서버로 요청하는 횟수가 많아지면 많아질수록 서버의 서비스 요청에 대한 거부율이 증가하는 특징을 보여주고 있다. 이와 반대로 P2Patching이 P2Cast 보다 다소 우수한 성능을 보이고 있다. 이는 서비스를 요구하는 클라이언트들의 조인요청이 발생하면 P2Cast는 서버에서 조인과 복원 등을 모두 수행하는 반면 P2Patching은 서버와 세션 루트가 이 일을 수행한다. 또한, 조인 수행시 P2Patching은 패칭 스트림을 사용함으로써 짧은 시간에 조인 프로세스를 완료하는 특성으로 인하여 서비스 요청 거부율이 낮게 나타나 다소 성능의 우수함을 보이고 있다.

그림 10은 Failure Probability에 따라 복원을 위한 노드들의 평균 접속 횟수를 측정한 그림이다. 멀티캐스트 스트림을 전송하는 도중 부모 노드들의 결함이 발생하게 되면 복원을 위하여 새로운 부모 노드에 대한 재연결 평균 접속 횟수를 표현한 것이다. P2Patching은 인접해 있는 노드를 찾아 한 홉씩 bottom-up 방식을 채택하는 복구 트리 구축 알고리즘인 반면에 P2Cast는 “Fattest pipe”를 가진 노드를 찾아 Top-down 프로세스를 수행하는 방법이다.

그림 11은 전체 호스트 수에 비례하여 부모 노드들의 장애가 발생한 호스트 수가 증가할수록 각 노드의 버퍼 스타베이션이 발생한 경우를 측정한 그

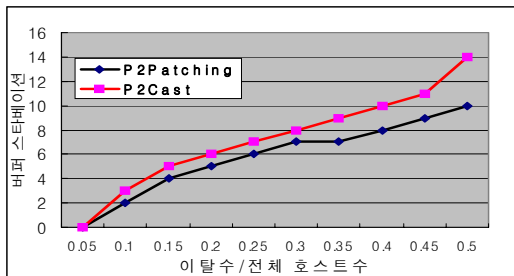


그림 11. 버퍼 스타베이션이 발생한 노드 수

림이다. 버퍼 스타베이션이 발생은 지터와 같은 QoS를 저하시키는 원인이 된다. 패칭 초기에는 거의 동일한 성능을 보이지만, 패칭 주기가 경과한 후에는 점차 격차를 보여주고 있다. 잦은 부모 노드의 장애가 발생할수록 스타베이션 현상은 P2Cast보다 P2Patching이 완만한 상승 곡선을 보임으로써 짧은 트리 탐색 알고리즘과 패칭 스트림을 이용한 동적 버퍼링 기법의 효율성을 보여주고 있다. 특히 P2Cast는 부모의 장애율이 45% 이상 발생한 경우에는 스타베이션 현상이 급상승한 결과를 보인다.

V. 결론

본 연구는 P2P 환경에서 패칭을 이용하여 주문형 스트림을 효율적으로 서비스하기 위한 응용 계층 멀티캐스트 기법인 P2Patching을 다음과 같이 제안하였다. 첫째, 패칭 스트림을 확장 사용하는 새로운 기법을 제안하여 QoS를 보장하고 초기 지연을 줄이는 시도를 하였다. 둘째, 멀티캐스트 트리로 조인할 때의 조인 지연을 감소시켰을 뿐만 아니라, 트리의 결함으로 인한 트리의 복원을 위하여 적은 지연 시간에 재연결하는 멀티캐스트 트리 구축 기법과 마지막으로 동적 버퍼링 기법을 통하여 잦은 결함에도 불구하고 연속적인 재생을 보장하는 기법도 제시하였다. 또한, 본 논문과 유사성을 가진 P2Cast와의 여러 가지 실험을 통하여 약 16% 정도의 평균적인 성능 향상에 대한 결과를 입증해 보였다.

그러나 향후 멀티캐스트 트리의 신뢰성을 보장하기 위한 다양한 기법들이 연구되어야 할 것이다. 특히, P2P 환경의 스트림 서비스에서 다기능을 제공하는 연구는 그리 많지 않다. 따라서 P2P 환경에서 사용자 서비스 편리성을 제고하기 위한 다기능을 위한 주문형 스트림 서비스에 대한 연구의 필요성이 제기된다.

참고 문헌

- [1] K. A. Hua, Y. Chai, S. Sheu, “Patching : A Multicast Technique for True Video-on-Demand Service”, *Proc. of ACM Multimedia* 1998, pp. 191-200, Bristol, England, September 1998.
- [2] 최성욱, 김종경, 박승규, 최경희, 김동윤, 최덕규, “효율적인 사용자 서비스를 위한 적응적 배칭 스케줄링 정책”, *대한전자공학회 논문지*

제37권 제2호, pp. 44-53, 2000.

[3] L. Golubchik, J. C. S. Lui, R. R. Muntz, "Adaptive Piggybacking: A Novel Technique for Data Sharing in Video-on-Demand Storage Servers", *Multimedia Systems* 4, pp. 140-155, 1996.

[4] C. C. Aggarwal, J. L. Wolf, P. S. Yu, "A Permutation-based Pyramid Broadcasting Scheme for Video-on-Demand Systems", *Proc. of ICMCS*, pp. 118-126, June 1996.

[5] Y. Chu, S. Rao and H. Zhang, "A case for end system multicast", in *Proceedings of ACM SIGMETRICS*, Santa Clara, CA, June 2000, pp. 1-12.

[6] M. Bawa, H. Deshpande, and H. Garcia-Molina, "Transience of peers and streaming media", *ACM SIGCOMM Computer Communication Review*, vol. 33 Issue 1, January 2003.

[7] D. A. Tran, K. A. Hua and T. T. Do, "A Peer-to-Peer Architecture for Media Streaming", in *IEEE journal on Selected Areas in Communications*, vol. 22, no. 1, Jan 2004.

[8] M. Castro, P. Druschel, A. -M. Kermarrec, A. Nandi, A. Rowstron and A. Singh, "SplitStream: High-bandwidth Multicast in Cooperative environments", in *ACM SOSR*, Bolton Landing, NY, USA, October 19-22, 2003.

[9] V. N. Padmanabhan, H. J. Wang, P. A. Chou and K. Sripanidkulchai, "Distributing streaming media content using cooperative networking", in *ACM/IEEE NOSSDAV*, Miami, FL, USA, May 12-14 2002.

[10] V. N. Padmanabhan, H. J. Wang, P. A. Chou, "Resilient peer-to-peer streaming", in *IEEE ICNP*, Atlanta, GA, USA, November 19-22 2003.

[11] "SSF Simulator implementation", <http://www.ssfnet.org/ssfImplementations.html>.

[12] Yang Guo, Kyungwon Suh, James F. Kurose, Donald F. Towsley, "P2Cast: peer-to-peer patching scheme for VOD service", in *Proceeding of the twelfth International Conference on WWW*, 2003.

[13] S. Banerjee, B. Bhattacharjee and C. Kommareddy, "Scalable application layer multicast" in *ACM SIGCOMM*, Pittsburgh, PA, USA, 2002.

김 종 경 (Jong-gyung Kim)

정회원



1990년 2월 호원대학교 컴퓨터 공학 졸업

1993년 2월 경희대학교 산업보대학원 컴퓨터공학과 석사 졸업

1996년 8월~현재 아주대학교 일반대학원 컴퓨터공학 박사수료

<관심분야> 멀티미디어 시스템 응용 및 S/W 시스템 구조, P2P 네트워크, 멀티캐스팅

이 재 혁 (Jae-hyuk Lee)

정회원



2001년 2월 아주대학교 정보 및 컴퓨터공학부 졸업

2003년 2월 아주대학교 정보통신전문대학원 정보통신공학부 석사 졸업

2005년 2월~현재 아주대학교 정보통신전문대학원 정보통신공

학부 박사수료

<관심분야> 네트워크 관련, 네트워크 보안, 네트워크 보안 시뮬레이터, 무선 게이트웨이, 홈 게이트웨이

박 승 규 (Seung-kyu Park)

정회원



1976년 3월~1977년 10월 한국과학기술 연구소(연구원)

1977년 11월~1978년 9월 한국전자기술 연구소(연구원)

1978년 10월~1982년 3월 프랑스 그레노블IMAG(연구원/학생)

1982년 4월~1984년 2월 한국전

자기기술연구소(실장/선임연구원)

1984년 3월~1985년 9월 미국 IBM 왓슨연구소(연구원)

1985년 10~1992년 2월 한국전자통신연구소(연구위원/책임연구원)

1992년 3월~현재 아주대학교 정보통신대 교수, 정보통신전문대학원장

<관심분야> 다중 및 분산처리 컴퓨터구조, 멀티미디어 스트림 처리 및 시스템 구조, 임베디드 시스템 및 테스트 분야, 게이트웨이 및 프락시, S/W 구조