협력 P2P 환경에서 VCR 기능을 위한 멀티캐스트 전송 기법

정회원 김 종 경*, 김 진 혁**, 박 승 규***

A Multicast Delivery Technique for VCR-like Interactions in Collaborative P2P Environment

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

요 약

멀티캐스트 기법은 큰 규모의 VOD 서비스에서 비용을 줄이기 위한 방법 중에 하나다. 그러나 멀티캐스트 전송 시스템에서 사용자의 편의를 위한 VCR 기능(VCR-like Interaction)을 구현하는 것은 복잡한 문제들을 내포하고 있어 이에 대한 해법이 필요하다. 따라서 본 연구는 P2P 개념과 패칭(Patching)을 혼합 적용한 P2Patching 기법을 제안한다. P2Patching은 VCR 기능 수행 주기 동안에 빈번한 멀티 캐스트 트리의 이탈(Departure)과 참여(Join) 행위에 대한 문제를 개선하기 위하여 다수의 피어(Peer)들이 협력적으로 VCR 기능 스트림을 VCR 기능을 수행하는 클라이언트에게 전송하고, VCR 기능 수행 후에는 이 클라이언트가 원하는 위치의 멀티 캐스트 세션으로 점프를 허용하는 CISS(Collaborative Interaction Streaming Scheme) 알고리즘으로 네트워크의 트래픽을 분산시키고 컨트롤 오버헤드와 지연(Latency)을 줄일 수 있다. 그리고 제안 기법의 성능 평가를 위하여 P2Cast[10] 그리고 DSL[11]에 대하여 스트림 전송 토플러지 성능과 컨트롤 오버헤드 그리고 스트리밍 질(Streaming Quality)을 실험하여 제안한 방법과 비교하였다. 이 실험에서 VCR 기능 수행 시에 DSL보다 P2Patching이 컨트롤 오버헤드는 약 30%정도 감소시켰으며 스트리밍의 질은 25~30%정도를 향상시켰다.

Key Words : Video on Demand, Patching, P2P Networking, VCR, Multicasting

ABSTRACT

Delivering multicast stream is one of the cost-saving approach in the large scale VOD environment. Because implementing VCR-like interactions for user's convenience in the multicast streaming system involves complex problems, we need the proper solutions for them. In this paper, we propose a hybrid scheme which uses the general P2P and the patching scheme with the Collaborative Interaction Streaming Scheme(CISS). CISS provides jumping functionability to the appropriate multicast session after VCR-like interaction in the environment in which multiple peers transmit VCR-like interaction streams to the VCR-like functionability request node to reduce the loads generated by frequent join or departure of peers at the multicast tree during providing VCR-like functionability. Therefore, with the proposed scheme we can distribute network traffic and reduce control overhead and latency. And to evaluate the performance of proposed scheme we compare it in the aspect of the performance of streaming delivery topology, control overhead and streaming quality with P2Cast[10] and DSL[11]. The simulation result shows that proposed P2Patching reduces about 30% of process overhead and enhances about 25~30% of streaming quality compared with DSL.

* 아주대학교 컴퓨터공학과 멀티미디어 및 컴퓨터구조 연구실 (Jgkimpower@hanmail.net)
** 아주대학교 정보통신공학과 멀티미디어 및 컴퓨터구조 연구실 (gene@ajou.ac.kr)
*** 아주대학교 정보통신공학과 멀티미디어 및 컴퓨터구조 연구실 (sparky@ajou.ac.kr)
논문번호 : KICS2006-05-194, 접수일자 : 2006년 5월 2일, 최종논문접수일자 : 2006년 7월 10일

Ⅰ. 서 론

멀티캐스트 기법의 활용은 많은 사용자들을 수용하여 비용/효율성의 측면에서 탁월한 확장성을 기대할 수 있다. 확장성은 스트리밍 서비스의 시장성을 증진시키기 위한 중요한 요소라고 한다면 기존의 클라이언트/서버 방식의 스트리밍 서비스[1]에서는 네트워크와 서버의 부하로 인하여 한정된 사용자를 수용할 수 있는 서비스의 한계를 지니고 있었다.

이러한 문제들을 극복하기 위한 새로운 패러다임으로서 P2P 환경에서 응용 계층 멀티캐스트(Application Layer Multicast)[3~8]를 활용하는 기법들이 연구되었다. 이들 대다수의 논문들은 멀티캐스트 트리를 구성하는데 있어서 피어들이 중간 노드(Intermediate node)나 리프 노드(Leaf node)로서 스트리밍 서비스를 수행하기 위한 라이브 스트림에 적합한 가상 멀티캐스트(Virtual Multicast) 기법을 제안하였다. 그러나 이 기법들은 사용자의 요구가 간헐적으로 발생하거나 다양한 스트리밍 서비스를 요구하는 VOD 서비스에 적용하기엔 무리가 따른다. 그러므로 네트워크의 혼잡 그리고 피어들의 이탈(Departure) 등과 같은 유동적인 P2P 환경에서 비동기 사용자들을 위한 VOD 서비스 기법들이 필요하였다. 최근에는 VOD 서비스 효율성을 향상시키기 위하여 멀티캐스트 전송 기법에 초점에 맞춘 연구[11-13]들이 발표되었다.

그러나 Fast-forward/reverse 그리고 Pause, Resume와 같은 VCR 기능(VCR-like Interaction)을 제공하는 것은 많은 서버 및 네트워크 자원의 소비와 오버헤드가 발생하여 어려움이 따른다. VCR 기능을 지원하기 위해서는 단순한 알고리즘을 구현하여 네트워크와 서버의 자원을 경제적으로 소비하기 위한 효율적인 VCR 기능 구현기법이 필요하다. 그러나 오버레이 네트워크에서 VCR 기능구현에 관한 연구는 극히 일부에 지나지 않는다.

따라서 본 연구에서는 VCR 기능을 지원하기 위하여 P2P 개념과 패칭(Patching) 기법을 혼합 적용한 것으로 네트워크상에서 다수 피어들의 버퍼를 하나의 거대한 분산 메모리 형태로 구성된 버퍼 구조에서 VCR 기능을 수행하는 클라이언트에게 VCR 기능 수행을 위한 스트림을 협력 제공하도록 하고, VCR 기능수행을 완료한 클라이언트에게 원하는 위치의 트리로 점프를 허용하는 CISS (Collaborative Interaction Streaming Scheme)을 제안한다. 이 기법은 VCR 기능 수행 주기 동안에 잦은 멀티 캐스트 트리의 이탈과 참여 수행에 대한 문제점을 제거하기 위하여 다수의 피어들로부터 협력적으로 VCR 기능 스트림을 전송하게 하는 방법으로 네트워크의 트래픽을 분산시키고 컨트롤 오버헤드와 지연을 줄일 수 있는 기법이다.

이 논문의 구성은 제1장에서는 연구배경의 전반적인 개요를, 제2장에서는 관련 연구를 언급한다. 제3장에서 본 논문에서 제안하는 시스템에 대한 개요와 트리 구축 알고리즘, 그리고 CISS 기법을 소개한다. 그리고 제4장에서는 VCR 기능 구현과 제5장에서는 결함 복원을 논하고 제6장 실험 및 고찰에서는 본 논문에서 제안한 기법과 P2Cast[10] 및 DSL[11]에 대하여 비교한 실험결과를 보여준다. 마지막으로 제7장에서 결론을 기술한다.

Ⅱ. 관련 연구

2.1 전통적 패칭 기법

전통적 패칭[1]은 특정 사용자의 행위나 사용자 인지의 유형을 만들어 서비스 대기 시간에 대한 한계성을 이용하는 True VOD의 스트림 전송 기법이다. 이 기법은 일정한 시간 간격마다 한 편의 비디오 전체를 멀티캐스트로 전송하여 사용자 그룹에게 서비스하는데, 이때 두개의 연속적인 멀티캐스트 스트림 사이의 전송 시간 차이를 패칭 윈도우(Patching Window)라고 한다. 패칭 윈도우 크기 이내에 발생하는 서비스 요청은 바로 이전에 전송을 시작한 멀티캐스트 그룹에 참여하게 하고 멀티캐스트 전송 시작부분부터 참여 이전의 수신하지 못한 비디오 부분은 서버가 서비스 요청을 한 클라이언트에게 패칭 스트림을 보냄으로써 지연이 없는 VoD 서비스를 할 수 있다. 패칭 스트림을 받는 클라이언트들은 초기 부분을 우선적으로 재생할 수 있도록 하기 위하여 멀티캐스트 스트림들을 로컬 슬라이딩 버퍼(Local Sliding Buffer)에 저장한다. 그리고 패칭 스트림을 통하여 수신한 스트림들을 먼저 재생하고 그 이후에는 로컬 슬라이딩 버퍼의 스트림들을 재생을 시작한다. 이러한 기법들은 네트워크 계층에서 기본적으로 브로드캐스트나 멀티캐스트 채널을 이용하는 방식이다.

2.2 VCR 기능

클라이언트/서버 환경에서 구현한 연구들은 다음과 같다. VCR 기능 수행하기 위한 스트림을 전송하기 위하여 약간의 채널을 사용하는 기법[12][13]과

버퍼에 저장되어 있는 스트림을 X 배속으로 재생하는 방식으로서 버퍼 용량의 한계 때문에 다른 멀티캐스트 스트림으로 병합하는 기법[14][15], 그리고 프락시 서버를 두어 VCR 기능을 구현하도록 하는 기법[16] 등이 있다.

한편, 응용 계층 환경에서 VCR 기능의 구현을 위한 연구[11]들은 극히 일부에 지나지 않는다. DSL[11]은 VCR 기능의 VoD 서비스를 위하여 Layer 집합으로 구성된 Dynamic Skip List를 구축한다. 오버레이 구조에서 재생점 키(Playout Offset Key)를 가진 피어들을 병렬 링크로 구성하고 네트워크 코딩(Network Coding)을 사용하여 네트워크의 대역폭을 극대화하려는 기법이다. 그러나 이 기법은 VoD 특성상 간헐적으로 발생하는 사용자 요청이 균형 Skip List 구성하는데 어려움이 따르고, Skip List 구조 구성과 유지를 위한 고비용을 지불해야 하는 문제점이 있다.

2.3 응용 계층 멀티캐스트

CoopNet[8]은 실시간 비디오 스트리밍을 위한 시스템으로 미디어 콘텐츠를 여러 부분으로 나누어 분배(Distribution)시키는 MDC(Multiple Description Coding)를 적용하여 피어의 장애가 발생하더라도 데이터의 손실을 최소화한다. 이는 VoD를 위한 기능은 지원하지만 재생점들을 관리하는 기법은 제공되지 않는다.

NICE[18]와 ZIGZAG[4][5]은 계층 구조의 클러스터 기반 기법으로 확장성 있는 계층 구조의 P2P 동적 네트워크 특성을 고려하여 멀티캐스트 트리의 제어 트래픽을 감소시키는 방법을 연구하였다. 그러나 네트워크의 유동적인 환경에서 개념적인 대역폭을 다루는 기법으로 VoD 스트리밍 서비스에는 문제가 따른다.

P2Cast[10]은 P2P 환경에 패칭 기법을 사용하여 VoD 서비스 설계에 초점을 둔 응용 계층 멀티캐스트 기법이나 VCR 기능은 지원하지 않는다.

Ⅲ. 제안 시스템

3.1 시스템 개요

본 논문은 P2P 환경에서 응용 계층 멀티캐스트와 패칭 기법을 혼용한 기법인 VCR 기능을 지원하는 P2Patching을 제안한다. 이 기법에서 비디오 스트림을 원하는 클라이언트는 멀티캐스트 트리에 참여하여 VoD 서버나 다른 피어들로부터 요청한 비디오 스트림을 수신하고 버퍼링하며, 자식노드나 VCR 기능을 수행하는 다른 클라이언트들에게 버퍼링한 스트림 데이터를 제공한다. 이 기법은 새로운 클라이언트 요청이 발생할 때마다 멀티캐스트 스트림을 새롭게 생성하는 것 대신에 피어들의 버퍼에 저장되어 있는 많은 량의 비디오 스트림들을 재사용함으로써 시스템의 처리량을 증진시키고 서버 자원의 사용을 감소시키기 위한 것이다.

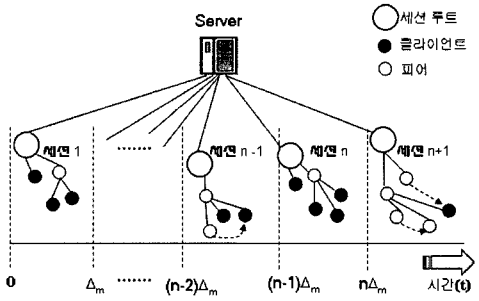


그림 1. System 개요

P2Patching 시스템의 개체는 그림 1과 같이 VoD 서버, 루트 피어, 피어 그리고 클라이언트들로 구성된다. VoD 서버는 비디오 스트림들이 저장되어 있는 개체로서 사용자 그룹인 멀티캐스트 세션을 형성하여 멀티캐스트 스트림을 전송하고 각 세션에 대한 정보를 관리한다. 피어는 세션을 처음 개설한 루트 피어와 중간 노드로서의 피어로 구분하며, 루트 피어는 세션 트리에 대한 개체 정보를, 중간 피어는 자식노드와 부모 노드에 대한 정보를 갖는다. 피어들은 수신된 비디오 스트림을 재생하고 다른 피어나 클라이언트에게 전송한다. 그리고 클라이언트는 스트림을 제공 받는 개체이다.

피어와 클라이언트는 표준 재생을 위한 P-stream과 VCR 기능 수행을 위한 스트림인 I-stream들을 버퍼링 할 수 있는 공간이 있어야 한다. 그림 1의 실선은 멀티캐스트 스트림을, 그리고 점선은 패칭 스트림을 전송받고 있는 모습을 보여주고 있다.

멀티캐스트 트리를 한 세션 단위로 구성하고 두 연속적인 멀티캐스트 스트림 사이의 패칭 윈도우(Patching Window) 크기를 Δ_m이라고 하고 시간 0에서 멀티캐스트 스트림이 시작한다고 가정할 때, 패칭 스트림의 전송 길이는 식 (1)과 같이 계산할 수 있다.

$$\forall t \in [(n-1)\Delta_m, n\Delta_m): \quad len(t) = t \bmod \Delta_m \quad \cdots \quad (1)$$

그리고 본 논문에서 제안하는 확장 패칭 기법은 사용자 대기시간과 멀티캐스트 트리의 단절이 발생했을 때 효율적으로 버퍼링을 할 수 있도록 한 기법으로서 전통적 패칭 기법을 확장한 것이다. 이것은 다음과 같은 기능을 수행한다.

첫째, 멀티캐스트 트리를 구성하고 있는 개체인 루트, 피어, 그리고 클라이언트가 멀티캐스트 트리로 참여할 때 초기 버퍼링에 필요한 초기 대기시간을 줄이기 위해 사용된다. 둘째, 멀티캐스트 스트림을 전송하는 과정에서 멀티캐스트 트리의 단절이 발생했다면 단절부터 복원될 때까지의 스트림들에 대한 손실을 가져오는데, 이를 효율적으로 복원하기 위한 방법으로 사용한다. 마지막으로 VCR 기능을 수행한 후, 다른 세션으로 점프를 시도할 때에 피어의 재생점과 멀티캐스트 전송점 갭(Gap)에 해당하는 부분의 스트림을 패치하기 위하여 확장 패칭 기법을 사용한다[2].

3.2 Multicast Tree Construction

멀티캐스트 트리 구축 기법은 멀티캐스트 트리 참여와 패칭 스트림을 제공하는 패칭 서버를 선택하는 알고리즘으로 구분할 수 있다. 본 논문에서는 VoD 특성을 고려하여 참여 지연(Join Delay)을 가장 우선적으로, 사용 가능한 큰 대역폭을 가진 피어를 차선으로 선택하는 기법을 사용한다. P2P 환경의 대다수 피어들은 리프(Leaf) 노드로서 자신의 자원을 시스템에 기여하지 않는다. 따라서 기본적으로 리프 노드가 사용 가능한 대역폭이 존재할 가능성이 가장 높다는데서 착안하여 멀티캐스트 트리에 참여하는 전략은 서버로부터 가장 홉(Hop) 수가 적은 하위 레벨의 리프 노드에 우선 접근한다. 리프 노드의 부모 대역폭이 새로운 자식 클라이언트를 받아들이기에 충분할 경우는 자식 노드로 참여하고, 그렇지 않을 경우는 리프 노드의 자식 노드로 참여하도록 하는 노드 참여 알고리즘을 사용한다.

그리고 패칭 서버 선택 알고리즘은 짧은 거리를 유지하기 위하여 피어가 접속해 있는 멀티캐스트 트리에서 가장 가까운 위치에 있는 피어를 선택하는 기법을 사용한다. 이를 위해서 현 위치에서 한 홉씩 Bottom-Up 방식으로 프로세스를 진행시킨다. 이러한 방법은 대역폭 소비를 최소화할 수 있을 뿐만 아니라 프로세스 오버헤드를 줄일 수 있다[2].

3.3 CISS 알고리즘

본 논문에서 제안하는 CISS의 수행하는 기본 개념은 다음과 같다. 이 기법은 VCR 기능의 수행을 원하는 클라이언트가 이를 수행하기 위해 필요한 세션들과 피어들을 선택하는 과정을 수행한다. 선택된 피어들은 VCR 기능 수행을 위한 스트림 즉, VCR 스트림을 생성하여 VCR 기능 수행 클라이언트에게 전송한다. 이 논문에서는 이 피어를 VCR 기능 서버라고 한다. 그리고 VCR 기능을 수행하는 클라이언트는 VCR 기능을 수행 종료한 후에는 종료점에 해당하는 가장 가까운 세션으로 점프를 시도하는 기법이다.

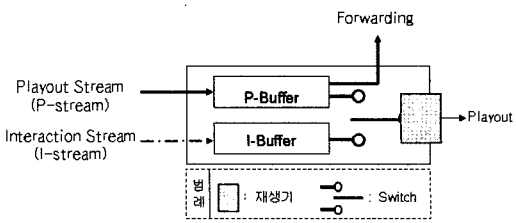


그림 2. 버퍼 구조도

3.3.1 Buffer Architecture

이 스트리밍 기법을 수행하기 위하여 사용되는 버퍼 구조는 그림 2와 같이 사용자가 표준시청을 위한 스트림인 P-stream을 저장하기 위한 공간인 P-Buffer와 VCR 스트림인 I-stream을 저장하기 위한 I-Buffer 공간으로 구분된다. VCR 기능을 수행할 때는 P-Buffer와 I-Buffer를 동시에 사용하고 표준 재생을 수행할 경우는 P-Buffer만을 사용한다. 전자의 재생점은 I-Buffer에 존재하는 반면에 표준 재생을 수행할 경우는 P-Buffer에 재생점이 존재한다.

3.3.2 P-Buffer 동작

한 클라이언트가 멀티캐스트 트리에 참여할 때는 식 (1)에서 정의한 $len(t)$ 시간 동안에 멀티캐스트 스트림과 패칭 스트림을 동시에 받는다. $len(t)$ 이후에는 패칭 스트림은 해지되고 가장 최근에 수신한 멀티캐스트 스트림들을 P-Buffer에 버퍼링한다. 또한, 각 피어들은 자신이 수신된 스트림들을 자식 노드들에게 전송해야하는 가정 때문에 P-buffer의 필요한 최대 용량 즉, buf_{size}는 $len(t)$가 $\triangle_m/2$ 보다 작거나 같을 경우에 $\triangle_m$의 크기가 필요하고, 그렇지 않을 경우는 $2 * len(t)$의 크기가 필요하다.

$$\rho - buf_{size} < P < \rho \quad \cdots \quad (2)$$

그리고 한 피어가 시간 t에서 멀티캐스트 스트림

을 전송받을 때의 위치가 ρ라고 할 때 필요한 버퍼 구간은 $[(\rho - buf_{size}), \rho]$와 같다.

또한, 같은 세션을 구성하고 있는 피어들의 재생점 집합인 재생점 구간 P는 식 (2)와 같이 표현할 수 있다. 한 피어가 버퍼 구간인 $[(\rho - buf_{size}), \rho]$의 비디오 스트림 내용들은 다른 자식 노드들에게 포워딩, 그리고 VCR 스트림을 생성하는데 사용한다.

3.3.3 I-Buffer 동작

Fast-Forward, Fast-Reverse와 같은 X 배속의 VCR 기능 구현은 표준 MPEG 스트림에서 X개의 GOPs (Group of Pictures) 중에서 하나의 GOP을 선택하여 재생함으로써 구현할 수 있다[19].

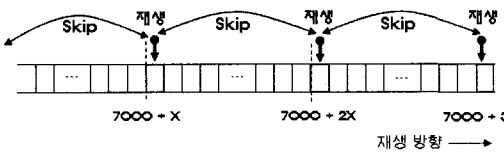


그림 3. I-Stream 선택의 예

3.3절의 제안 알고리즘에 의해 선택된 VCR 기능 서버는 자신의 P-buffer 스트림들 즉, X개의 GOP (Group of Picture) 중에서 하나의 GOP을 선택하는 방식으로 VCR 기능을 수행하는 클라이언트에게 전송한다. 다시 말하면, 그림 3에서와 같이 X 배속을 구현하기 위하여 7000+X, 7000+2X, 그리고 7000+3X번째 스트림이 전송된다. 이 때, 전송받은 이 클라이언트는 이를 자신의 I-Buffer에 버퍼링한다.

3.3.4 알고리즘 동작

VCR 기능 수행과정은 VCR 기능을 수행하는 클라이언트가 해당 세션들에서 선택된 VCR 기능 서버에게 VCR 스트림을 요청함으로써 이루어지는데 그 과정은 다음과 같다.

- **단계 1:** VCR 기능 구현에 필요한 스트림들이 버퍼링된 버퍼구간에 해당하는 세션을 선택하는 단계로서 VCR 기능을 수행하는 클라이언트가 이를 수행하기 위한 내용과 시간에 밀접한 관계있다. 예를 들면, 어떤 클라이언트가 세션 i 즉 $s(i)$에서 비디오 위치 T까지 X배속으로 Fast-Forward를 수행한다고 가정할 때, VCR 기능을 수행한 경과 시간에 따라 세션의 수는 결정이 되는데 이에 해당되는 세션은 $s(i)$, $s(i-1)$, ⋯, $s(i-j)$으로 이를 식 (3)과 같이 표현할 수 있다. 그리고 총 전송해야 할 스트림 량은 $\lfloor (T-\rho)/X \rfloor * R$과 같이 계산할 수 있다. 이 때 R은 초당 전송률이다.

$$j = \lfloor T/\Delta_m \rfloor - \lfloor \rho/\Delta_m \rfloor \quad \cdots \qquad (3)$$

- **단계 2:** VCR 기능을 수행하는 클라이언트와 해당 세션에 참여하고 있는 피어들 중에서 거리(Hop 혹은 Round-Trip Time)를 d, 그리고 측정된 대역폭을 b라고 할 때, 대역폭과 거리의 균형을 이루는 값 즉 d/b의 최소값을 가진 피어를 VCR 기능 서버로 선택한다. 선택된 j개의 VCR 기능 서버들은 VCR 기능을 수행하는 클라이언트에게 순차적으로 VCR 스트림을 전송하게 된다. 이 클라이언트는 자신의 I-Buffer의 콘텐츠를 재생함으로써 VCR 기능을 수행할 수 있다. 그리고 VCR 기능 수행 주기 동안에 현재 참여되어 있는 세션에서 수신된 스트림들을 자신의 P-Buffer에 버퍼링하여 이를 자식 노드들에게 포워딩한다.
- **단계 3:** VCR 기능 수행을 종료 후에는 종료점이 존재하는 가장 가까운 세션으로 점프를 수행하고 표준재생을 위하여 멀티캐스트 스트림을 받는다.

일반적으로 VCR 기능 수행 주기 동안 빈번하게 발생하는 트리의 이탈과 참여로 인하여 많은 컨트롤 오버헤드 및 지연시간이 발생하게 된다. 이는 새로운 사용자에 대한 서비스 지연 및 거부(Rejection)로 이어지고 서비스 질의 저하 원인이 되기도 한다. DSL[11]에서 빈번한 점프를 시도하는 알고리즘을 사용하여 과도한 오버헤드를 발생시키는 문제가 발생하였으나, 본 논문의 제안 기법은 다수의 VCR 기능 서버들이 VCR 기능을 수행하는 클라이언트에게 짧은 시간동안에 VCR 스트림들을 협력 제공함으로써 빈번하게 수행하는 트리의 이탈과 참여에 대한 문제를 개선하였다.

Ⅳ. VCR 기능 구현

3.3절에서 제안하는 기법에 의거 Fast-Forward, Fast-Reverse, 그리고 Pause-Resume과 같은 VCR 기능을 수행하는 방법과 점프 동작에 대해 설명한다.

4.1 Fast-Forward/Fast-Reverse

VCR 기능 서버들은 자신의 P-buffer에서 X개의

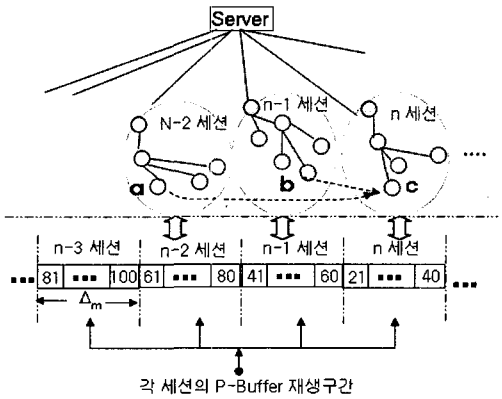


그림 4. 각 세션 VCR 스트림 전송의 예

GOP 중에서 하나의 GOP을 선택하는 방식으로 I-Stream들을 VCR 기능을 수행하는 클라이언트에게 전송한다. 이 클라이언트는 자신의 I-buffer에 있는 I-stream들을 각각 Forward 혹은 Backward 방향으로 재생함으로써 X배속의 VCR 기능을 구현할 수 있다. 이것은 한 세션에서 선택된 VCR 기능 서버가 전송량 $\Delta_m * R$대신에 짧은 시간동안에 $(\Delta_m * R)/X$을 전송함으로써 네트워크의 대역폭을 절감할 수 있다.

VCR 기능 수행 시간이 길어지면 I-stream을 제공해 줄 해당 세션의 수(VCR 기능 서버의 수)는 늘어난다. 예를 들면, 그림 4에서와 같이 n번째 세션의 클라이언트 c가 4배속 Fast Forward를 수행하려고 한다면, n-1번째 세션의 VCR 기능 서버 b가 자신의 R-Buffer에 있는 41번부터 60까지의 GOP들 중에서 41, 45, 49, 53, 그리고 57번째 GOP에 해당하는 스트림들을 전송하게 한다. 이 클라이언트가 계속해서 4배속 Fast-Forward를 수행한다면 n-2번째 세션의 VCR 기능 서버 a가 해당되는 61, 65, 69, 73, 그리고 77번째 GOP들을 전송하게 된다.

한편, 표준 MPEG 스트림을 사용하는 것은 네트워크의 트래픽을 줄여줄 수 있지만 Fast Reverse와 같은 VCR 기능을 수행할 경우는 Decoding의 문제가 발생한다. 이것은 MPEG의 특징으로서 Predictive Processing 기법을 사용하여 Forward Play에 적합하게 설계되어 있기 때문이다. 그러나 Encoding/Decoding에 관한 것은 Decoding 기법들[20, 21]을 통하여 이를 해결하여 원활한 Reverse Play를 할 수 있다.

또한, VCR 기능을 수행하는 클라이언트가 VCR 기능의 행위 종료를 원할 때는 VCR 기능 서버에게 VCR 기능 해지 메시지를 보냄으로써 I-stream 전송을 종료시킨다. 그리고 이 클라이언트가 종료점에 가장 가까운 세션으로의 점프를 수행하도록 하고 3.1절의 확장된 패칭 기법을 적용하여 표준 재생을 할 수 있다.

4.2 Pause/Resume

VCR 기능을 수행하는 클라이언트가 Pause 실행을 하였을 경우 Pause 실행 시간동안 0배속으로 I-Stream을 재생하나 재생점이 변하지 않는다. 이 경우도 이 클라이언트는 자식 노드들에게 멀티캐스트 스트림들을 전송받아 이 스트림들을 포워딩시킨다.

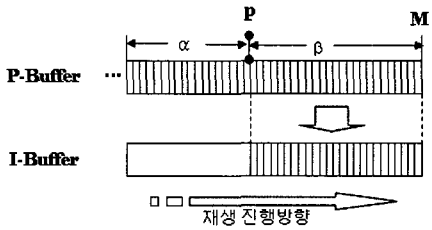


그림 5. Pause 행위에 대한 버퍼 동작

그리고 I-stream을 생성하기 위하여 그림 5.에서와 같이 P-Buffer에 전송받은 스트림들 중에서 앞으로 재생할 프레임들 즉, β 구간에 있는 프레임들을 I-Buffer에 복사하여 0 배속으로 재생한다. 이때 표현한 α 구간은 재생한 프레임 구간, M은 Pause 실행 시간에서의 멀티캐스트 스트림의 위치, 그리고 P는 재생 점을 표현하였다. 그리고 Resume이 시작되면 자신의 I-Buffer의 스트림들을 재생을 시작하는 동안에 I-Buffer의 M위치 이후부터 멀티캐스트 스트림을 전송받기 위하여 가장 가까운 세션으로 점프를 수행하는 것으로 3.1절의 확장된 패칭 기법을 적용한다.

4.3 Forward Jumping/Reverse Jumping

본 논문의 점프 시도는 다음과 같은 경우에 실행된다. 첫째, 클라이언트가 VCR 기능 행위를 종료한 다음 표준 재생을 위하여 해당 세션으로 참여를 하고자 할 때 둘째, 사용자가 비디오 시청 중에 특정 부분으로 건너뛰기를 원할 때 셋째, 부모 노드의 장애로 인하여 새로운 부모 노드로 참여할 때이다. 상기 두 가지의 경우는 4.1항과 4.2항에서 기술한 VCR 기능 수행 후 해당 세션으로 점프를 하는 행위이고, 셋째의 경우는 멀티캐스트 트리의 결함이 발생했을 때 복원을 위한 프로세스로서 제5장에서 이를 언급한다.

V. 결함 복원(Failure Recovery)

사용자의 로그아웃이나 컴퓨터, 그리고 네트워크의 결함 등으로 인하여 비디오 스트림에 대한 전송이 중단될 경우 두 가지의 문제가 발생한다. 하나는 빠른 시간에 새로운 부모 피어를 선택하여 점프하는 것이고, 또 하나의 다른 문제는 장애 시에 수신하지 못한 스트림을 복원하는 일이다. 본 논문에서 두 번째의 경우는 3.1절에서 언급한 바 있는 확장된 패칭 스트림을 통해서 간단하게 복원할 수 있다. 그러나 새로운 부모를 선택하는 알고리즘에서는 효율적이고 좀 더 빠른 기법이 필요하다. P2Cast[10]에서는 장애가 발생할 때마다 서버가 개입함으로써 서버의 과부하가 발생하여 서비스 거부률(Service Rejection Rate)이 높아지는 원인이 되고 부모 참여 알고리즘을 수행할 때 긴 Bock Time이 발생하는 문제점을 지니고 있다.

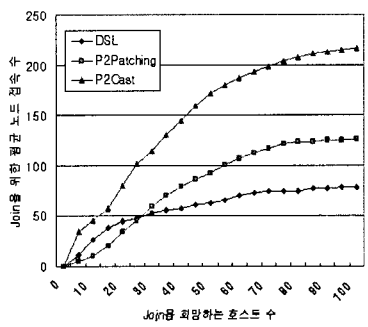
본 논문에서는 이러한 문제를 해결하기 위하여 멀티캐스트 트리의 결함 복원할 때 서버가 개입하지 않고 조부모 피어들이나 혹은 형제 피어들 중에서 가장 가까운 피어를 선택하는 알고리즘을 사용한다[2].

VI. 실험 및 고찰

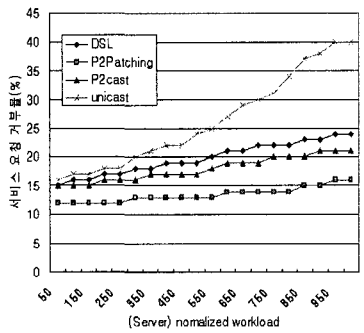
본 논문의 성능을 평가하기 위하여 시뮬레이션 기법을 사용한다. 시뮬레이션 도구는 SSFNet[17] 시뮬레이션 프레임웍을 사용하였으며, SSFNet은 프로세스 기반 이산 사건 중심 시뮬레이션 커널(Process-based Discrete Event Oriented Kernel)이다. 시뮬레이션 커널의 소스는 공개되지는 않았으나 그 중에서 네트워크의 시뮬레이션을 지원하는 라우터 링크 네트워크 인터페이스 카드 등 대부분의 인터넷 서브시스템들을 시뮬레이션 하는데 필요한 다양한 객체들은 Java로 시뮬레이션 특성을 변경하여 구현할 수 있는 장점을 가지고 있다.

6.1 시스템 실험 환경

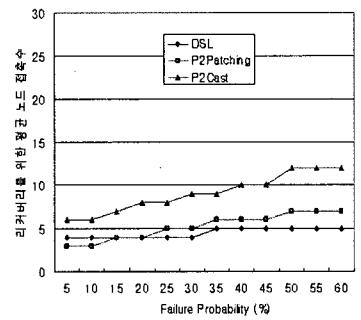
네트워크 구성은 공정한 실험을 위하여 DSL 실험 환경과 유사하게 만들었다. 전체 네트워크는 4개의 Transit 도메인으로 구성하고, 각 Transit 도메인에는 5개의 Transit 노드가 있으며, 각 Transit 노드는 8개의 Stub노드가 연결된 6개의 Stub 도메인으로 연결되어 있다. 그리고 Stub 노드사이의 대역폭은 4Mbps, Transit 노드와 Stub노드 사이의 대역폭



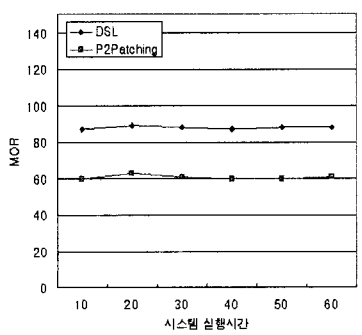
[그림. 6] 참여지연(Join Delay)



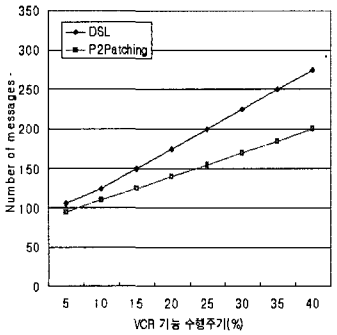
[그림. 7] 서비스 요청 거부율



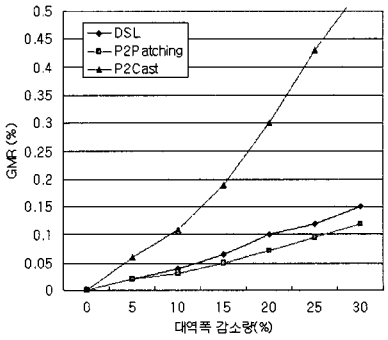
[그림. 8] 결함확률에 따른 복원지연



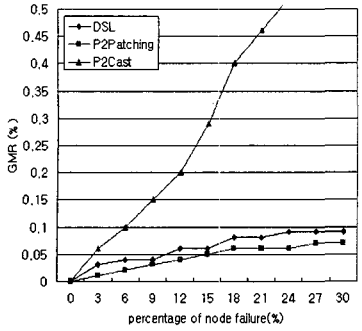
[그림. 9] 메시지 컨트롤 오버헤드



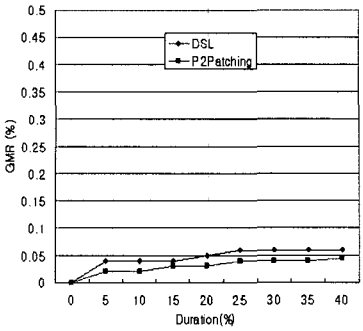
[그림. 10] VCR 기능 수행 주기동안의 메시지 수



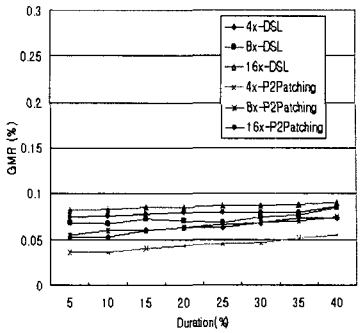
[그림. 11] 대역폭 감소에 따른 GMR



[그림. 12] 노드 결함률에 따른 GMR



[그림. 13] VCR 기능 수행주기 동안의 GMR



[그림. 14] VCR 기능 수행 속도에 따른 GMR

은 9Mbps, Transit 노드 사이의 대역폭은 12Mbps로 설정하였다. 두 노드간의 라우팅 알고리즘은 Shortest Path Algorithm을 사용하였고, 서버에 저장된 비디오는 150분 분량을 가진 하나의 비디오를 사용하는 것을 가정한다. 그리고 클라이언트들의 도착률은 포아송(Poisson) 분포에 따르고 전송 토폴로지가 형성된 이후에는 2분마다 하나의 클라이언트가 Connection Failure 발생하도록 하였다.

6.2 성능 측정 기준과 방법

실험평가는 스트림 전송을 위하여 전송 토폴로지의 성능과 컨트롤 오버헤드 그리고 스트리밍 질(Streaming Quality)을 평가를 위한 실험으로 나누어 실시하며 이에 대한 기준과 방법은 다음과 같다.

첫째, 전송 토폴로지의 성능 실험은 다음의 측정 기준과 방법을 사용한다. 1) 참여지연(Join Delay)은 서비스 요청하였을 때 전송 토플로지에 참여하기 위하여 접촉을 시도한 평균 노드의 수를 측정하였다. 2) 서비스 요청 거부율(Rejection Probability)은 스트림 서비스를 받기 위하여 서버에 접속을 시도하였으나 거부된 서비스로써 총 요청된 회수와 거부된 서비스 횟수의 비로 나타낸다. 3) 결함확률에 따른 복원 지연은 부모 노드들이 결함이 발생할 때 자식노드가 새로운 부모노드를 찾기 위한 평균 접촉 횟수를 표현한 것이다.

둘째, 컨트롤 오버헤드에 관한 실험으로, 1) MOR(Message Overhead Ratio)을 측정하는 실험으로 총 트래픽 발생량 중에서 전송 토플로지 유지와 데이터 전송을 위해 발생되는 컨트롤 트래픽 량을 비율로 표현하였다. 2) VCR 기능 수행 동안에 발생하는 오버헤드를 측정하기 위하여 메시지 수를 측정하였다.

셋째, 스트리밍 질을 평가하는 실험으로 네트워크 대역폭의 감소시켰을 때와 여러 종류의 VCR 기능을 수행했을 때의 서비스 질을 측정을 하기 위하여 재생 시각에 필요한 스트림을 이용할 수 없을 때를 Missing이라고 정의하고 전체 Missing GOPs 수와 실험주기 동안에 참여한 클라이언트 수를 비율로 표시한 GMR (GOPs Missing Rate)을 측정하였다.

6.3 전송 토폴로지 성능 실험

그림 6의 참여지연은 참여를 희망하는 노드의 증가에 따른 노드들의 평균 접속 횟수를 측정한 결과이다. P2Patching의 빠른 트리 탐색 알고리즘으로 인하여 P2Cast보다 적은 접속 횟수로 부모노드를 선택하는 결과를 보여준다. DSL은 참여 노드수가 많을 경우는 P2Patching이 우수한 결과를 보이는데 이는 Skip List의 구조적 특성상 네트워크의 크기가 늘어나도 일정한 접촉회수를 보여주는 것으로 분석된다. 참여를 위한 적은 접속 횟수는 노드들에게 서비스를 보다 빠르게 제공할 뿐만 아니라 전체 네트워크 트래픽에 많은 영향을 미치게 된다. 그림 7은 P2Patching이 P2Cast와 DSL보다 가장 우수한 성능을 보이고 있다. 이는 서비스를 요구하는 클라이언트들의 결함이 발생하면 DSL과 P2Cast는 서버가 복원을 모두 수행하는 반면 P2Patching은 세션 루트가 이 일을 수행함으로써 서버의 부하를 경감시킨 결과로 분석된다. 그림 8의 결함 복원 지연은 결함 확률에 따른 복원을 위해 접촉한 노드 수를 나타내었다. P2Patching은 인접해 있는 노드를 찾아 한 홉씩 상향식으로 동작하는 복구 트리 구축 알고리즘인 반면에 P2Cast는 "Fattest Pipe"를 가진 노드를 찾아 반복적으로 하향식 프로세스를 수행하는 방법이다. 한편 DSL과 P2Patching을 비교해보면 결함확률이 낮은 경우에는 P2Patching이 우수하나 결

함확률이 높아지면 우수한 결과를 보이고 있는데 그 이유는 그림 6에서와 같다.

6.4 컨트롤 오버헤드 실험

그림 9는 시스템 수행경과에 따라 전송 토플로지 구축 및 유지에 필요한 MOR을 측정한 그림이다. DSL이 P2Patching보다 네트워크의 트래픽을 30% 이상 더 발생시키는 것으로 나타났다. 이것은 스트림 데이터의 일정량을 전송하기 위해 DSL은 제어 트래픽이 9배, 그리고 P2Patching은 1.5배 정도를 발생시킨다. 이러한 DSL의 높은 컨트롤 오버헤드의 발생은 Skip List를 유지를 위하여 Offset Key와 같은 자료들을 주기적으로 수정하기 때문인 것으로 분석된다. 그림 10은 VCR 기능 수행 주기에 발생한 메시지 수를 측정한 그림이다. P2Cast는 VCR 기능을 제공하지 않은 기법으로 인하여 이 측정에서 제외시켰다. DSL은 VCR 기능 수행주기 동안에 빈번하게 점프를 시도함으로써 네트워크 링크에 이와 관련된 메시지 트래픽을 유발시킨다. 반면에 P2Patching은 다른 노드들로부터 협력적으로 VCR 스트림을 전송 받아 수행하는 기법을 사용하여 DSL보다 대략 30% 트래픽을 감소시킬 수가 있었다.

6.5 스트리밍 질에 관한 실험

그림 11과 그림 12는 네트워크 대역폭 감소 및 노드 결함율의 변화에 대하여 GMR을 측정한 그림이다. DSL과 P2Patching에서 GMR의 평가가 우수한 것으로 나타내고 있으나 P2Cast의 성능이 상당히 떨어지는 결과를 보이고 있다. 이것은 패칭 서버와 멀티캐스트 스트림을 제공해줄 노드를 선택하는 알고리즘이 긴 Block Time을 유발시키고 트리의 깊이가 긴 특성을 가진 결과라 판단된다. 반면에 DSL은 적응적 구조인 Skip List를 사용하였고, P2Patching은 짧은 트리를 구성하여 빠른 시간에 복원을 시킬 수 있었다. 이 결과에서도 P2Patching이 DSL보다 약 25~30% 정도의 우수한 성능을 보였다. 그림 13은 VCR 기능 수행주기 동안에 GMR을 측정한 것으로 P2Patching이 DSL보다 대략 30% 정도의 우수한 결과를 보이고 있다.

마지막으로 그림 14는 VCR 기능 수행 속도에 따른 GMR을 표현하였다. P2Patching의 8배속과 DSL의 4배속 수행결과는 거의 같은 성능을 보임으로써 P2Patching의 CISS 알고리즘 우수성을 입증하였다. VCR 기능 수행 속도가 높아갈수록 GMR이 높아지는 현상을 볼 수 있는데 이 경우도 GMR 수치가 P2Patching보다 DSL이 좀 더 높게 나타난 것을 볼 수 있었다.

Ⅶ. 결 론

VCR 기능 수행 주기 동안에 필요한 스트림들이 버퍼링이 되어있는 피어들이 협력적으로 VCR 기능 수행을 위한 스트림들을 VCR 기능을 수행하는 클라이언트에게 VCR 스트림을 제공하고, 이 클라이언트가 VCR 기능을 수행하는 종료 시점에 표준 재생을 하기 위하여 해당하는 멀티 캐스트의 트리로 점프를 허용하는 CISS(Collaborative interaction streaming Scheme)을 제안하였다. 이 기법은 긴 시간 동안에 VCR 기능 수행하는 과정에서 발생하는 빈번한 멀티 캐스트 트리의 이탈과 참여로 인하여 발생하는 오버헤드와 지연을 줄이기 위한 기법이라 할 수 있다. 그리고 제안한 기법의 평가를 위하여 P2Cast[10] 그리고 DSL[11]에 대하여 스트림 전송 토폴로지의 성능과 컨트롤 오버헤드 그리고 스트리밍 질을 측정하여 비교하였다. 이 실험에서 VCR 기능 수행 시에 DSL보다 P2Patching이 컨트롤 오버헤드는 약 30% 정도 감소시켰으며, 스트리밍의 질은 25~30% 정도를 향상시켰다. DSL은 참여지연이나 결함확률에 따른 복원지연에서 성능의 우수함을 보이고 있으나 Skip List를 유지를 위하여 고비용을 지불하고 네트워크의 트래픽을 과도하게 발생시키는 것으로 나타나 확장성 측면에서 문제점으로 지적되었다. 이를 근거로 본 논문에서 제안한 기법이 P2Cast과 DSL에 비해 확장성/비용성 측면에서 우수한 성능 결과를 보였다.

향후 연구 방향은 불특정 다수의 다양한 VOD 서비스 요구가 급증할 것으로 예측되는바 인터넷 환경과 이동 통신 환경의 혼합 구조에서 멀티캐스트 스트리밍을 위한 P2P 서비스가 이루어져할 것으로 판단된다.

참 고 문 헌

[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] 김종경, 이 재혁, 박승규, "P2Patching: 주문형 P2P 서비스를 위한 효율적인 패칭 기법",

한국통신학회 논문지, 제31권 제2호, 2006.

[3] H. Deshpande, M. Bawa and H. Garcia-Molina, "Streaming live media over peers", in Work at CS-Stanford, 2002.

[4] 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.

[5] D. A. Tran, K. A. Hua and T. T. Do, "ZIGZAG: An efficient Peer-to-Peer Scheme for Media Streaming", in IEEE INFOCOM, San Francisco, USA. 2003.

[6] M. Castro, P. Druschel, A. -M. Kermarrec, A. Nandi, A. Rowstron and A. Singh, "Split-Stream: High-bandwidth Multicast in Cooperative environments", in ACM SOSP, Bolton Landing, NY, USA, October 19~22, 2003.

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

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

[9] T. T. Do, K. A. Hua, M. A.Tantaoui, "P2VoD: Providing Fault Tolerant Video-on-Demand Streaming in Peer-to-Peer Environment". To appear in the IEEE International Conference on Communications(ICC 2004), June 20-24, Paris, France.

[10] 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

[11] D. Wang and J. Liu, A Dynamic Skip List based Overlay Network for On-Demand Media Streaming with VCR Interactions, Technical Report, Simon Fraser University, Canada, May 2005.

[12] J-M Choi, S-W Lee, K-D Chung, "A Multicast Scheme for VCR Operations in a Large VOD system", ICPADS pp. 555~561, 2001.

[13] X. Zhou, R. Luling, Li Xie, "Solving a Media Mapping Problem in a Hierarchical Server Network with Parallel Simulated Annealing", Procs. 2000 International Conference on Parallel Processing, pp. 115~124, 2000.

[14] Z. Fei, I. Kamel, S. Mukherjee, and M. Ammar, "Providing interactive functions through active client buffer management in partitioned video broadcast", Proc. of First International Workshop on Networked Group Communication, (NGC'99)Pisa, Italy, Nov. 1999.

[15] Emmanuel L., Abram-Profeta and Kang G. Shin, "Providing Unrestricted VCR Functions in Multicast Video-on-Demand Servers", In Proc. of IEEE. Int' Conf. on Multimedia Computing and Systems (ICMCS'98),. Austin, Texas, 1988.

[16] B. Qazzaz, R. Suppi, F. Cores, A. Ripoll, P. Hernandez, E. Luque, "Providing interactive Video on Demand Services in Distri- buted Architecture", euromicro p. 215, 29th Euromicro Conference (EUROMICRO'03), 2003.

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

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

[19] B. Wildemuch, G. Marchionini, M. Yang, G. Geisler, T. Wikens, A. Hughes, and R. Gruss, "How Fast is too fast ? Evaluating Fast Forward Surrogates for Digital Video", in Proc. JCDL'03, Houston, TX, May, 2003.

[20] David A. Helder, Sugih Jamin, "End-host multicast communication using switch-trees protocols", Proceedings of the 2nd IEEE/ ACM International Symposium on Cluster Computing and the Grid, pp. 419, 2002

[21] Ming-Syan Chen and Dilip D. Kandlur, "Stream Conversion to Support Interactive Video Playout" IEEE MultiMedia, vol. 3, no. 2, pp. 51~58, 1996.

김 종 경 (Jong-gyung Kim) 정회원

1990년 2월 호원대학교 컴퓨터공학과 졸업
1993년 2월 경희대학교 산업정보대학원 컴퓨터공학과 석사 졸업
1996년 8월~현재 아주대학교 일반대학원 컴퓨터공학 박사수료
1984년~1999년 경희대학교 정보처리처 팀장
1999년~2004년 (주) 원포텍 대표이사
1999년~2006년 경기대학교, 인천대학교 강사
2003년~2005년 시립 인천전문대 겸임교수
<관심분야> 멀티미디어 시스템 응용 및 S/W 시스템 구조, P2P 네트워크, 멀티캐스팅

김 진 혁 (Jin-hyuk Kim) 정회원

2003년 2월 아주대학교 정보 및 컴퓨터공학부 졸업
2005년 2월 아주대학교 정보통신전문대학원 정보통신공학부 석사
2005년 3~현재 아주대학교 정보통신전문대학원 정보통신공학과 박사과정 재학중
<관심분야> 멀티미디어 시스템 응용 및 S/W 시스템 구조, P2P 네트워크, 네트워크 보안, 네트워크 보안 시뮬레이터

박 승 규 (Seung-kyu Park) 정회원

1976년 3월~1977년 10월 한국과학기술연구소(연구원)
1977년 11월~1978년 9월 한국전자 기술연구소(연구원)
1978년 10월~1982년 3월 프랑스 그레노블IMAG(연구원/학생)
1982년 4월~1984년 2월 한국전자기술연구소(실장/선임연구원)
1984년 3월~1985년 9월 미국 IBM 왓슨연구소(연구원)
1985년 10월~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