

효율적인 VOD 서비스를 위한 RVOD와 NVOD간의 전환 프로토콜

김 명 훈[†] · 박 호 현^{††}

요 약

최근 광대역화 되어가는 네트워크 환경에 발맞춰 VOD의 수요 역시 급증하고 있다. 기존의 VOD 서비스는 RVOD 또는 NVOD 중 한 가지 방식으로만 서비스가 가능하였다. 이렇듯 한 가지 방식으로만 서비스를 하게 되면 실시간 변화하는 클라이언트의 수요에 대응할 수 없어서 서버에 과부하가 걸리거나, 서버의 대역폭을 낭비하는 결과를 초래하는 등 서버의 네트워크 자원을 효율적으로 사용하지 못하였다. 이에 본 논문에서는 효율적인 VOD 서비스를 위한 프로토콜에 관한 연구를 수행하였다. 기존의 VOD 서비스 알고리즘을 적용할 수 있는 새로운 프로토콜을 제안하고, 시뮬레이션을 통해 성능을 분석하였으며, 새로운 프로토콜을 적용시킨 서버/클라이언트 구조의 시스템을 개발하였다. 본 논문에서는 RVOD/NVOD 통합 모드 전환 프로토콜을 제안한다. 제안된 프로토콜은 RVOD와 NVOD 두 가지 방식 모두를 제어할 수 있고, 두 방식간의 모드 전환을 할 수 있다. 그러므로 본 프로토콜을 사용하여 클라이언트의 수요 변화에 맞춰 모드를 전환한다면 서버의 대역폭을 효율적으로 관리할 수 있다. 특히 기존의 RVOD 및 NVOD 연구 방식들을 적용할 수 있다. 그러므로 본 논문에서 제시한 프로토콜은 앞으로의 VOD 시장에서 활용도가 매우 높을 것으로 기대된다.

키워드 : RVOD, NVOD, 모드전환, 프로토콜

A Mode Switching Protocol between RVOD and NVOD for Efficient VOD Services

Myoung-Hoon Kim[†] · Ho-Hyun Park^{††}

ABSTRACT

Recently, as network environment has broadened, the demands on VOD have been increased. The VOD services can be categorized into two types, RVOD and NVOD. Practical VOD services adopt one of them exclusively. Since a method using only one of RVOD and NVOD is not able to deal with frequently variable demand of clients, it leads to a result of overload on a server and a waste of server bandwidth. The efficiency of the network resource usage becomes lower.

Hence this paper presents a study on the protocol for efficient VOD services. We propose a new protocol applicable for the existing VOD service algorithm, analyze its performance through simulation, and developed server/client systems applying the new protocol.

We propose a mode switching protocol combined with protocols used in RVOD and NVOD. The proposed protocol is not able only to control both RVOD and NVOD but also to change the mode between RVOD and NVOD. As a result of using the proposed protocol to meet frequently variable demand, server bandwidth can be used efficiently. Especially, it can be applied to the existing VOD service algorithms. Therefore, we expect that the proposed protocol in this paper will be widely used in emerging VOD markets.

Keyword : RVOD, NVOD, Mode Switching, Protocol

1. 서 론

지난 수년간 네트워크 환경은 HFC(Hybrid Fiber Coax), xDSL(x Digital Subscribe Line), FTTH(Fiber To The

Home)과 같이 빠른 전송속도의 환경을 구축하며 발전하고 있다. 게다가 현재 추진되고 있는 BcN이 확대 보급될 경우 데이터 통신 기술의 발전과 우수한 양방향 특성으로 인해 VOD(Video On Demand)/EOD(Education On Demand), 다 채널 방송서비스 등 다양하고 진일보된 서비스 제공이 가능해지고, 이러한 서비스를 이용하려는 수요 또한 증가하고 있다.

VOD는 통신망으로 연결된 컴퓨터나 텔레비전을 통해 원

※ 이 논문은 2007년도 중앙대학교 학술연구비(일반연구비) 지원에 의한 것이다.

† 정 회 원 : 중앙대학교 전자전기공학부 석사

†† 정 회 원 : 중앙대학교 전자전기공학부 부교수

논문접수: 2008년 3월 18일

수정일: 2008년 7월 7일

심사완료: 2008년 7월 7일

하는 프로그램을 언제든지 받아볼 수 있는 영상 서비스이다. VOD는 데이터를 전송하는 방식에 따라 크게 RVOD(Real VOD)와 NVOD(Near VOD)로 분류된다. RVOD는 사용자가 원하는 시간에 비디오를 선택하여 볼 수 있는 서버와 클라이언트간의 1:1 서비스이다. 빨리 돌리기, 되돌리기, 일시정지 등과 같은 기능을 제공하여 비디오를 보는 것과 똑같은 방식의 시청이 가능한 대화형 서비스이다. 아무리 네트워크가 광대역화 되어간다고 해도 수만, 수십만 명에게 동시에 RVOD로 서비스하기에는 대역폭에 한계가 있다. 이런 경우 NVOD 서비스를 사용하게 된다. NVOD는 하나의 채널에 다수의 사용자들이 접속할 수 있는 브로드캐스팅 또는 멀티캐스팅 방식을 이용하여 전송하는 1:n 서비스로써, 동일 프로그램을 복수의 채널을 통하여 각 채널마다 일정 시간 간격을 두고 순차적으로 반복하여 방송하면 클라이언트는 지정된 시간대에 희망하는 프로그램을 볼 수 있는 서비스로써 서버의 비용을 획기적으로 줄인 방식이다. 대화형서비스를 지원하지 않고, 비디오를 시청하기 전에 대기시간을 약간 감수해야 하는 단점이 있다. 동일한 대역폭을 사용하더라도 NVOD 방식이 RVOD에 비해 더 많은 사용자에게 서비스 할 수 있다. NVOD가 RVOD에 비해 적은 채널을 필요로 한다는 것은 요청이 많은 비디오의 경우이다. NVOD는 고정적인 채널수만큼의 대역폭을 항상 점유하고 있어야 하기 때문에 수요가 적은 비디오의 경우에는 효과적이지 못하다.

따라서 요청이 많은 비디오에서는 NVOD서비스로, 요청이 적은 비디오의 경우는 RVOD 서비스로 자동 전환하여 서버의 대역폭을 관리할 수 있는 통합적인 프로토콜이 필요하다. 기존의 VOD 서비스를 제어하는 방식이나 프로토콜에서는 이러한 통합된 프로토콜이 존재하지 않았다. 이에 본 논문에서는 RVOD와 NVOD를 모두 제어 가능하고, 사용자의 수요 변화에 대응할 수 있도록 두 방식간의 전환이 가능한 프로토콜을 제안 한다.

본 논문의 구성은 다음과 같다. 2절에서는 관련 연구로서 지금까지 사용되었던 RVOD 프로토콜과 NVOD서비스 방법들에 대해 설명하고 각각의 장단점에 대해 언급한다. 3절에서는 본 논문에서 제안하는 RVOD/NVOD 통합 모드전환 프로토콜을 설명한다. 먼저 프로토콜을 메시지를 정의하고, 순차도와 시나리오를 통해 프로토콜의 메시지들이 어떠한 순서로 전송이 되는지 흐름에 따라 설명하며, 모드 전환의 예시를 보여준다. 4절에서는 모드 전환 프로토콜을 사용한 경우의 성능을 분석한 결과를 보인다. 끝으로 5절에서는 제안된 RVOD/NVOD 통합 모드 전환 프로토콜에 대해 결론을 내리고 향후 연구 과제에 대해서 언급한다.

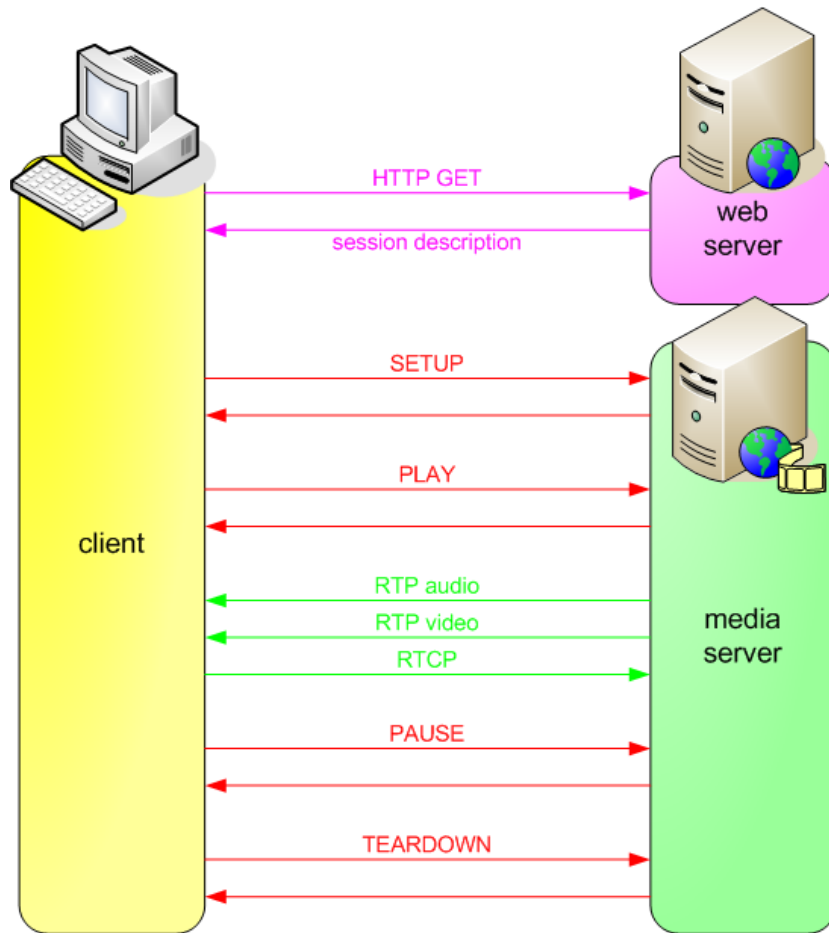
2. 관련 연구

RVOD는 사용자가 원하는 시간에 비디오를 선택하여 볼 수 있는 서버와 클라이언트간의 1:1 서비스이다. RVOD는 사용자가 원하는 시간에 원하는 비디오를 볼 수 있다는 장

점이 있지만 한명의 클라이언트가 하나의 전송채널을 사용하기 때문에 시스템 구축 및 운용비용이 증가하고 많은 수의 클라이언트가 접속했을 때 한계가 정해진 서버의 대역폭만으로는 감당할 수 없는 단점이 있다. 한계가 정해진 대역폭을 효율적으로 사용하기 위해 고안된 NVOD는 브로드캐스팅 또는 멀티캐스팅을 사용하여 하나의 채널에 여러 명의 사용자가 접속하여 비디오를 감상할 수 있도록 하는 1:n 서비스이다. 클라이언트의 숫자가 증가하더라도 서버의 대역폭에는 변동이 없는 장점이 있지만 접속된 클라이언트가 없더라도 항상 고정 대역폭을 사용하는 단점과 클라이언트가 비디오를 시청하기 위해서는 일정시간을 기다려야하는 단점을 가지고 있다. 이에 본 논문에서는 접속된 클라이언트의 수가 많을 경우에는 NVOD로, 접속된 클라이언트의 수가 적을 경우에는 RVOD로 모드 전환이 가능한 프로토콜을 제안한다. 본 프로토콜을 사용하면 클라이언트의 수에 따라 모드 변경이 가능하기 때문에 많은 수의 클라이언트가 접속한 경우에 서버가 폭주하는 현상을 나타내는 RVOD의 단점과 접속된 클라이언트가 없을 경우에도 항상 대역폭을 낭비하고 있는 NVOD의 단점을 서로 보완할 수 있으므로 서버의 대역폭을 효율적으로 사용할 수 있다. 모드 전환이 가능하기 위해선 먼저 두 방식을 모두 제어할 수 있어야 한다. RVOD와 NVOD를 통합하여 제어하며 모드 전환할 수 있는 프로토콜을 제안하기 위해 현재 사용되고 있는 RVOD의 대표적인 프로토콜과 지금까지 연구되어 오던 대표적인 NVOD 방식들을 살펴본다.

2.1 RTSP(Real Time Streaming Protocol)

사용자가 비디오 재생을 제어할 수 있도록 하려면, 미디어 플레이어와 서버는 재생 시작, 재생 중지, 이전이나 이후 시점으로 재생 재위치, 비디오의 급진진 재생, 비디오의 되감기 재생 등의 재생 제어 정보를 교환하는 프로토콜을 필요로 한다. 이런 기능은 DVD 비디오를 볼 때나 음악CD를 들을 때 CD 플레이어를 갖고 조작하는 것과 흡사하다. RFC 2326에 정의되어 있는 RTSP가 이런 프로토콜이다[1]. RTSP는 RVOD에 사용하는 대표적인 프로토콜이다. (그림 1)은 RTSP의 기본 동작을 나타낸 순차도(Sequence Diagram)이다. 클라이언트는 자신이 원하는 비디오를 보기 위해 웹에서 자신이 원하는 비디오를 선택하고, 선택한 비디오에 대한 정보를 웹서버로부터 수신한다. 수신한 정보를 통해 VOD를 서비스하고 있는 미디어 서버에 SETUP 메시지를 전송한다. SETUP 메시지는 비디오의 URI와 요청하는 미디어의 트랙번호를 기술한다. SETUP 메시지를 받은 서버는 그 응답으로 클라이언트에게 SessionID를 부여한다. RTSP에서는 재생시작시간 및 재생 종료 시간을 지정하여 PLAY메시지와 함께 전송할 수 있다. PLAY 메시지는 실제로 스트림을 전송을 요청하는 메시지이다. PLAY 메시지가 전송되면 RTP(Real-time Transport Protocol)를 통해 오디오 및 비디오 데이터를 전송한다. 이러한 RTP는 RTCP(Real-time Transport Control Protocol)는 이용하여 제어



(그림 1) RTSP의 기본 동작

할 수 있다. PAUSE 메시지를 통해 일시정지를 할 수 있으며 TEARDOWN 메시지를 통해 접속을 종료한다. RTSP는 재생 제어만을 목적으로 하는 프로토콜이기 때문에 오디오 또는 비디오를 압축하는 기법에 대해서는 정의 하지 않으며, 전송하는 프로토콜 역시 제약하지 않는다.

본 논문에서는 이러한 RTSP를 기반으로 여러 NVOD 방식도 수용 가능하며, RVOD와 NVOD간의 모드를 전환할 수 있는 확장 프로토콜을 제안한다.

2.2 NVOD(Near Video On Demand) 서비스 방법

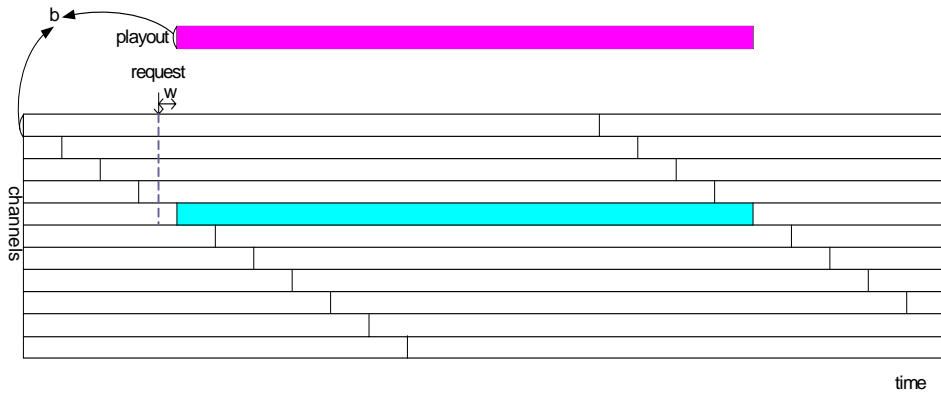
NVOD는 브로드캐스팅 또는 멀티캐스팅을 사용하여 하나의 채널에 여러 명의 사용자가 접속하여 비디오를 감상할 수 있도록 하는 1:n 서비스이다. 이러한 NVOD는 크게 시간 분할 방식과 데이터 분할 방식으로 나눌 수 있다. 동일한 비디오를 복수의 채널을 통해 일정한 시간간격을 두고 전송하는 시간분할의 대표적인 방법으로는 conventional broadcasting 방법이 있다[2]. 데이터 분할 방식은 비디오 데이터를 여러 부분으로 분할하고 여러 채널을 이용하여 분할된 데이터를 전송한다. 이러한 데이터를 분할하여 전송하는 방법으로는 pyramid broadcasting[3], fast broadcasting[4], flexible broadcasting[5] 등의 방법이 있다.

본 논문에서는 앞서 언급한 여러 가지 NVOD 방법들을 제어 및 모드 전환 할 수 있는 프로토콜을 제시한다.

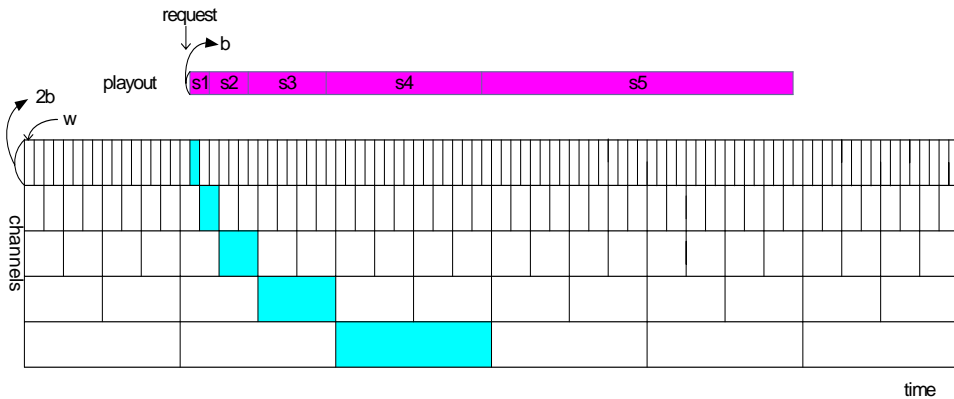
2.2.1 Conventional Broadcasting(CB)

RVOD는 사용자의 요청이 많을 때 엄청난 서버의 대역폭을 요구한다. 이러한 단점을 개선하기 위해 여러 NVOD 서비스 방법들이 생각되었다. 그중에서 전통적으로 많이 이용되고 있는 NVOD서비스 방법은 conventional broadcasting이다[2]. 이 방법은 전체 비디오를 일정한 시간 간격을 두고 여러 채널을 통해 전송하고, 각 채널을 브로드캐스트 또는 멀티캐스트의 방식으로 전송하므로 한 채널당 다수의 가입자들이 접속하여 동시에 비디오를 시청할 수 있도록 하는 방법이다.

(그림 2)에 conventional broadcasting의 시간-대역폭 그림이 있다. 윗부분이 재생영역이고 아랫부분이 브로드캐스팅영역이다. 서버는 그림에서와 같이 동일한 비디오를 시간 간격을 두어 여러 채널에 반복해서 전송을 한다. 클라이언트는 자신이 요청한 시간으로부터 가장 가까운 시간에 시작하는 채널에 접속하여 해당 스트림(그림 2에서 채널영역의 색깔된 부분)을 전송받아 바로 플레이하게 된다. 이때 최대 대기시간은 w이다. 예를 들어 120분 길이의 비디오를 그림



(그림 2) Conventional Broadcasting의 시간-대역폭 그림



(그림 3) pyramid broadcasting($\alpha=2$)의 시간-대역폭 그림

에서와 같이 12개의 채널을 통해 전송하면 각 채널의 시간 간격은 10분이 되므로 대기시간은 10분이 된다. 이 방법의 장점은 단말기에 재생을 위한 매우 작은 저장 공간만을 필요로 한다는 점이고 단점은 서버에서 활용하는 대역폭에 비해 너무 큰 대기시간을 갖는다는 점이다. 그러나 대기시간을 줄이면 서버 대역폭이 증가한다. 예를 들어 최대 대기시간을 1분으로 한다면 120개의 채널이 필요할 것이다.

2.2.2 Pyramid Broadcasting(PB)

Pyramid broadcasting은 비디오 데이터를 분할해서 전송한 최초의 방법이다. 비디오 데이터를 여러 부분으로 분할하고 여러 채널을 이용하여 분할된 데이터를 전송한다[3]. (그림 3)은 pyramid broadcasting($\alpha=2$)의 시간-대역폭 그림이다. 이 방법은 데이터의 분할, 서버 측의 전송알고리즘, 클라이언트 측의 알고리즘의 세부부분으로 나뉘어서 볼 수 있다.

① 비디오의 분할

첫 번째 분할(s_1)은 최대대기시간(w)의 α 배 길이의 크기로 분할한다. 그리고 다음 분할은 이전 분할의 α 배의 크기로 분할한다. 즉 i 번째 데이터 조각의 크기는 첫 분할의 α^i ($i=1, 2, 3, \dots, k$)배가 되고 $\alpha \geq 1$ 이다. 만약 비디오를 k 개로 분

할하였다면 k 개의 채널이 필요하게 된다. (그림 3)과 같이 α 를 2라고 하면 1, 2, 4, 8, 16, ...의 비율로 데이터를 분할한다.

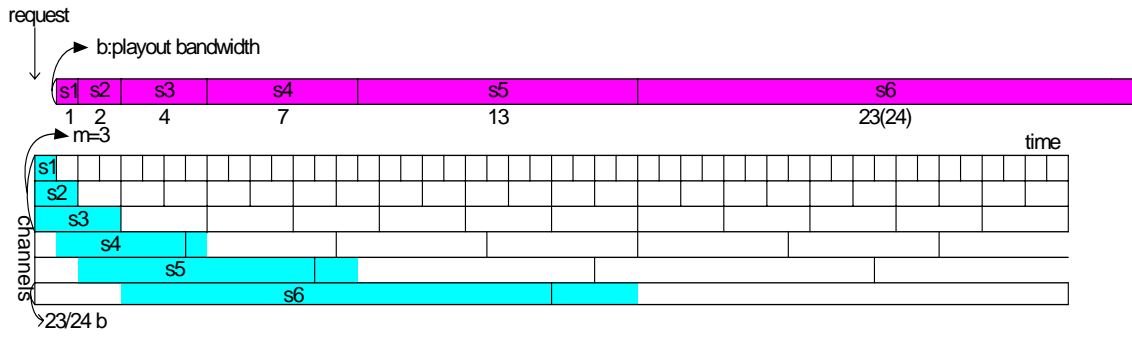
② 서버 측의 전송 알고리즘

- k 개의 채널을 만든다(k 는 분할의 개수) 각 채널의 대역폭은 $\alpha \times b$ (b 는 재생 대역폭)이다.
- s_i (i 번째 분할)을 i 번째 채널을 통해 반복 전송한다.

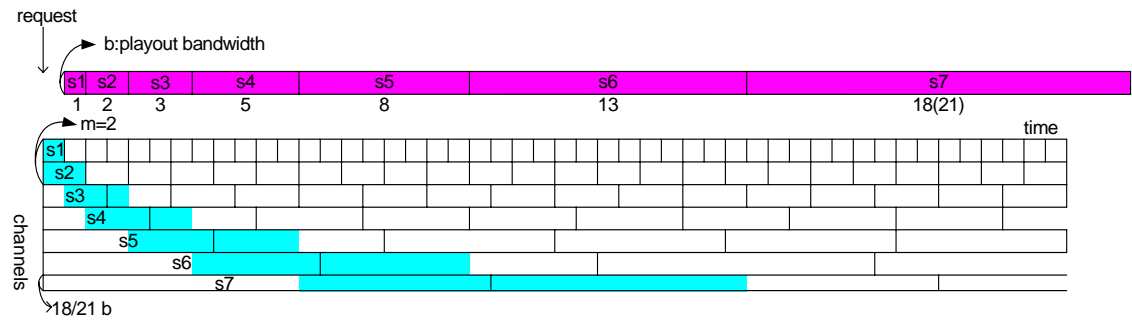
③ 클라이언트 측의 수신 알고리즘

- 첫 번째 채널을 통해 첫 번째 분할부터 전송받으며 동시에 재생을 시작한다.
- s_i 는 첫 채널부터 순서대로 전송받도록 하고 s_{i-1} 이 재생되기 시작한 이후로 가능한 빠른 시간에 전송받는다.
- 모든 분할은 분할의 시작부분부터 받는다.

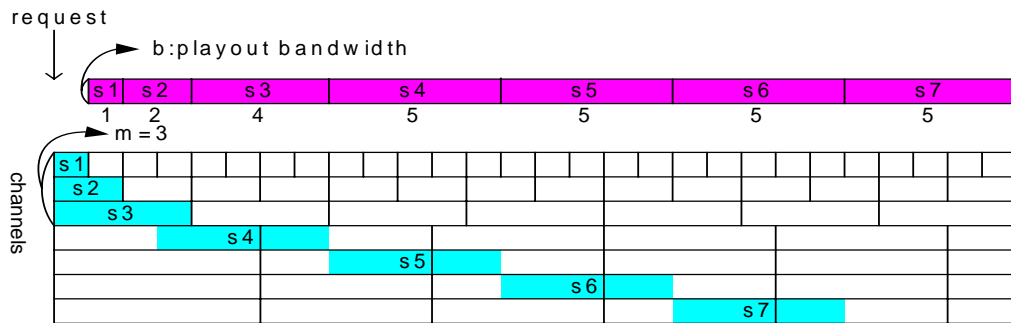
Pyramid broadcasting은 같은 대역폭을 이용할 경우 이전의 Conventional broadcasting에 비해 상당히 작은 대기시간을 필요로 한다. 이 말은 바꿔 생각하면 같은 대기시간을 가질 경우 상당히 적은 대역이 필요하다는 말이 될 수 있다. 하지만 비디오의 50% 이상을 저장할 수 있는 저장 공간이 필요하다는 것이 큰 단점이다.



(그림 4) 클라이언트의 네트워크 대역폭 및 버퍼의 제한이 없을 경우의 flexible broadcasting(m=3) 시간-대역폭 그림



(그림 5) 클라이언트의 네트워크 대역폭이 2b로 제한된 경우의 flexible Broadcasting(m=2) 시간-대역폭 그림



(그림 6) 클라이언트의 버퍼의 제한이 있는 경우의 flexible broadcasting(m=3) 시간-대역폭 그림

2.2.3 Flexible Broadcasting(FXB)

Flexible broadcasting은 서버의 대역폭의 효율과 클라이언트의 대기시간 감소에 초점을 두고, 클라이언트의 저장 공간에 제한이 있는 경우를 감안한 방법이다[5].

① 비디오의 분할방법

클라이언트의 저장 공간이 s_{max} 로 제한되지 않았다면 아래 수식중 위의 두 가지만으로도 비디오를 분할 할 수 있다. 만약 s_{max} 로 제한이 되었다면 s_{max} 보다 크게 분할되어야 하는 경우 해당 분할의 크기를 s_{max} 로 강제한다.

$$s_n^m = \begin{cases} 2^{n-1}, & (n \leq m) \\ \sum_{k=1}^m s_{n-k}^m, & (n > m) \\ s_{max}, & (s_n^m \geq s_{max}) \end{cases}$$

② 서버 측의 전송 알고리즘

- 먼저 분할의 개수 k개의 채널을 만든다.
- k번째 분할(마지막 분할)을 제외한 모든 분할을 각 채널

을 통해 네트워크 대역폭 b로 반복 전송한다.

- k번째 분할(마지막 분할)을 $\frac{s_k}{2^{k-1} \times b}$ 의 대역폭으로 반복 전송한다.

③ 클라이언트 측의 수신 알고리즘

- 앞 채널부터 순서대로 동시에 m개의 채널에 접속하여 데이터를 전송 받는다.
- i번째 분할이 재생되기 직전에 전송이 완료되도록 한다. 예를 들면 채널의 주기가 $5 \times w$ 라고하면 재생되기 $5 \times w$ 전에 수신을 시작한다.

(그림 4)는 클라이언트의 네트워크 대역폭과 비디오를 저장할 수 있는 버퍼가 충분한 경우를 나타낸 시간-대역폭 그림이고 아래의 (그림 5)는 네트워크 대역폭이 2b로 제한된 경우를 나타낸 그림이다.

마지막으로 (그림 6)은 클라이언트의 네트워크 대역폭은 제한이 없지만, 버퍼가 제한이 있는 경우를 나타낸 시간-대역폭 그림이다.

3. RVOD/NVOD 통합 모드 전환 프로토콜

이전에 연구되었던 VOD 서비스 방법들은 대부분 클라이언트의 수요를 고려하지 않은 방법들로써 수요의 변화에 대응하지 못하여서 서버의 네트워크 대역폭을 효율적으로 관리할 수 없었다. 또한 RVOD와 NVOD를 모두 제어할 수 있는 통합 프로토콜도 존재하지 않았다. 본 논문에서는 RVOD와 NVOD를 모두 제어할 수 있고, 클라이언트의 수요에 따라 RVOD와 NVOD간의 모드를 전환할 수 있게 함으로써 이전의 방법보다 효율적으로 서버의 네트워크 대역폭을 관리할 수 있도록 하는 RVOD/NVOD 통합 모드 전환 프로토콜을 제안한다.

3.1 프로토콜 정의

RVOD/NVOD 전환 프로토콜은 기본적으로 RTSP에 기반을 둔 대화형 프로토콜이다. RVOD/NVOD 전환 프로토콜은 동영상을 재생하는데 필요한 제어 메시지를 전송하는 응용계층의 프로토콜이다. RVOD와 NVOD를 모두 제어할 수 있으며, 필요에 따라 RVOD와 NVOD간의 모드를 전환할 수 있다.

사용되는 메소드는 RTSP와 흡사하나 많이 간략화 된 형태라 볼 수 있겠다. 본 논문에서 정의 하는 RVOD/NVOD 전환 프로토콜의 메소드는 다음의 4가지이다.

3.1.1 메소드 정의

① SETUP

RTSP의 SETUP은 URI와 요청하는 미디어의 트랙번호를 기술한다. 또한 클라이언트에서 지원하는 프로토콜을 나열하고 해당 프로토콜에 대한 포트를 지정해 서버가 참조할 수 있게 하는 역할을 한다. 반면 RVOD/NVOD 전환 프로토콜에서의 SETUP은 VOD서버에게 해당 파일에 대해 전송을 준비하라고 알리는 역할과 해당 서비스의 설명을 요청하는 역할을 한다.

다음은 SETUP 메소드와 그에 대한 응답의 예제이다.

```
C->S SETUP
CSeq : 1
File : test1.nvd
ContentId : 1
```

ContentId는 해당 파일이 VOD서버에서 서비스 중인 비디오의 고유 숫자이다.

```
S->C SETUP OK
CSeq : 1
Service Type : NVOD
NVOD Type : CB
State : Active
TotalTime : 3599 Sec 141666
MultiCasting IP : 224.88.40.1
StartPort : 1
ActivePort : 6
WaitTime : 10 min
```

```
SessionId : 01234567
```

위의 응답과 같이 요청한 비디오의 서비스 타입이 NVOD인 경우의 응답에는 서비스 타입(Service Type), NVOD 타입(NVOD Type), 현재 서비스의 동작 여부(State), 비디오의 전체 길이(TotalTime), 멀티캐스트 그룹 IP (MultiCasting IP), 시작 포트 번호(StartPort), 포트의 개수(ActivePort), 대기시간(WaitTime), 클라이언트를 식별하기 위한 세션 번호(SessionID)등의 정보를 포함한다. NVOD 타입의 경우 서버와 클라이언트 모두 지원하는 방식으로 서비스 되어야 한다.

```
S->C SETUP OK
CSeq : 1
Service Type : RVOD
State : Active
TotalTime : 3599 Sec 141666
SessionId : 01234567
```

한편, RVOD로 서비스 되고 있는 경우는 서비스타입, 서비스의 동작여부, 그리고 비디오의 전체 길이, 클라이언트 식별 번호만을 응답하면 된다.

② PLAY

PLAY 메소드의 경우는 RTSP와 유사한 역할을 한다. 하지만 RTSP에서는 시작 및 끝 시간을 조절할 수 있는 반면에, RVOD/NVOD 전환 프로토콜의 경우에는 RVOD로 서비스 중인 경우에만 재생시작시점을 서버에게 알릴 수 있다. 이 PLAY 메소드는 NVOD로 서비스 되고 있는 경우 멀티캐스트 IP를 통해 비디오 스트림을 받게 되어 사용이 불필요해 보이지만, 사용자의 수요를 파악하기 위해 VOD Server에 해당 비디오를 보고 있는지 알릴 필요가 있으므로 NVOD의 경우에도 사용한다. NVOD에서 사용할 경우 StartOffset은 사용하지 않는다.

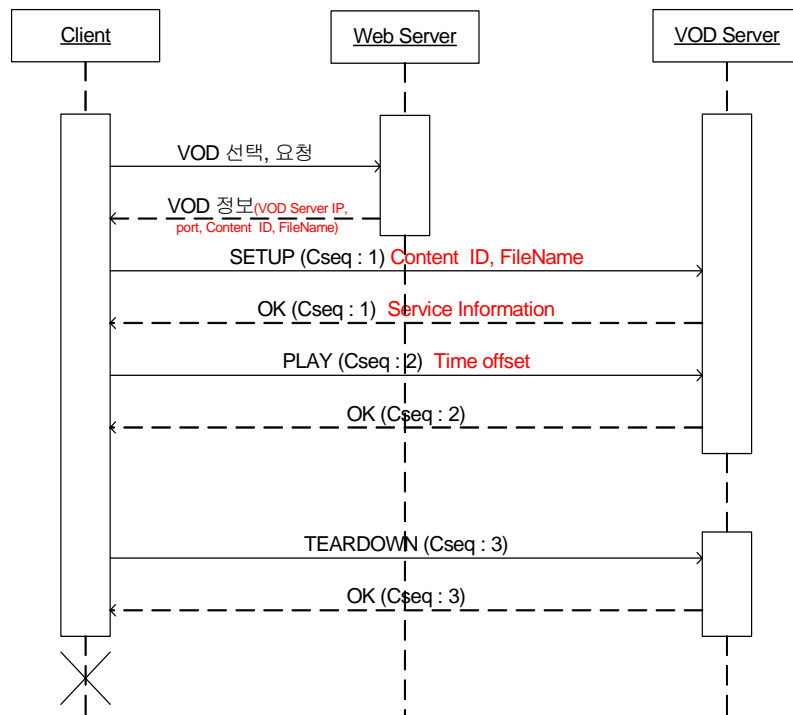
다음은 PLAY 메소드와 그에 대한 응답의 예제이다.

```
C->S PLAY
CSeq : 2
File : test1.nvd
StartOffset : 3 min 20 sec 3876
SessionId : 01234567
S->C PLAY OK
CSeq : 2
```

③ MODECHANGE

본 논문의 핵심이 되는 메소드이다. 서비스 중인 비디오의 요청이 급증하여 RVOD에서 NVOD로 전환이 필요하거나, 요청이 급감하여 NVOD로 서비스를 하는 것이 채널 낭비라고 판단이 되어 NVOD에서 RVOD로의 전환이 필요한 경우 해당 비디오를 서비스 받고 있는 모든 클라이언트에게 전송하게 되는 메소드이다.

다음은 NVOD에서 RVOD로 전환 시 전송되는 메소드의 예제이다.



(그림 7) 모드 전환이 발생하지 않은 경우의 순차도

S->C MODECHANGE

CSeq : 3
Target : RVOD
File : test1.nvd
ContentId : 1
ListenerPort : 2468

C->S MODECHANGE OK

CSeq : 3
SessionId : 01234567

접속 중인 모든 클라이언트로부터 모드 전환이 되었다는 OK 메시지를 받게 되면 서버는 전환되기 이전의 서비스를 중지할 수 있다. OK 메시지가 도착하지 않더라도 VOD 서버에 클라이언트의 접속이 유지되고 있는지를 모니터링 하기 위해 주기적으로 ping테스트를 하는 등, 접속 유지를 감별하는 인터페이스가 구현 되어 있다면 MODECHANGE 명령이 내려지기 이전에 접속한 모든 클라이언트가 연결이 끊어진 것이 확인이 된다면 기존의 서비스를 중지할 수 있다. RVOD에서 NVOD로 전환하는 경우는 Target에 NVOD가 들어가는 걸 제외하고는 위 예제와 동일하다.

④ TEARDOWN

서비스를 받거나 하는 것을 종료할 때 쓰인다. 클라이언트나 서버 어느 쪽에서도 사용이 가능한 메소드이다.

C->S TEARDOWN

CSeq : 4
SessionId : 01234567

S->C TEARDOWN OK

CSeq : 4

3.1.2 순차도 및 시나리오

RVOD/NVOD 모드 전환 프로토콜은 PLAY도중 서버에서 모드 전환이 발생하지 않은 경우와 발생한 경우의 두 가지 시나리오가 존재한다.

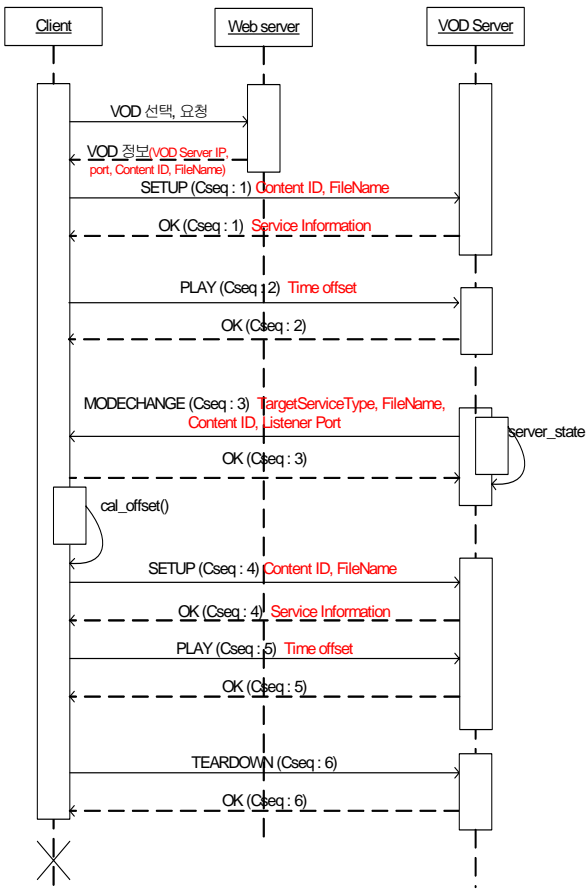
① 모드전환이 없는 경우

모드 전환이 발생하지 않은 경우의 순차도는 (그림 7)과 같고, 시나리오는 RVOD와 NVOD 두 경우 모두 다음과 같이 4단계로 나타낼 수 있다.

- i) 웹에 접속하여 VOD를 선택한 후, 선택한 VOD가 서비스되고 있는 VOD Server의 IP 와 프로토콜 메시지를 주고 받을 수 있는 포트번호, 그리고 해당 Content ID 및 파일명 등의 정보를 전송받는다.
- ii) 웹으로부터 받은 정보를 이용해 VOD서버에 SETUP 메소드를 전송하고, 그에 대한 응답으로 해당 비디오 콘텐츠의 서비스 정보를 받는다.
- iii) PLAY 메소드를 전송하여 비디오 스트림을 받아서 재생한다.
- iv) 재생이 끝나거나 도중에 정지할 경우, 그리고 서버 측에서 더 이상 서비스를 하지 않게 된 경우 TEARDOWN 메소드를 통해 연결을 종료한다.

② 모드전환이 있는 경우

모드전환이 일어난 경우의 순차도는 (그림 8)과 같다. 모드 전환이 발생한 경우의 시나리오는 i), ii), iii)까지는 모드전환이 발생하지 않은 경우의 시나리오와 동일하고 그 이후의 시나리오는 다음과 같다.



(그림 8) 모드전환이 발생한 경우의 순차도

iv) PLAY도중 서버에서 모드 전환이 일어날 경우 서버는 접속되어 있는 모든 클라이언트에게 모드전환을 요구하고 새로운 서비스를 구동한다.

v) 클라이언트는 MODECHANGE 명령과 함께 받은 파일명, ContentID, 프로토콜 포트 번호 등의 정보를 이용하여 새롭게 생성된 서비스(모드만 전환된 기존의 비디오)와의 새로운 연결 설정을 위해 VOD 서버에게 SETUP 메소드를 전송한다.

vi) 모드가 전환되기 이전에 미리 계산한 비디오의 시간 오프셋 값을 PLAY 메소드에 함께 실어서 해당 부분부터의

전송을 요청한다.

vii) 연결 종료시에는 TEARDOWN 메소드를 통해 종료한다.

3.2 모드 전환 예시

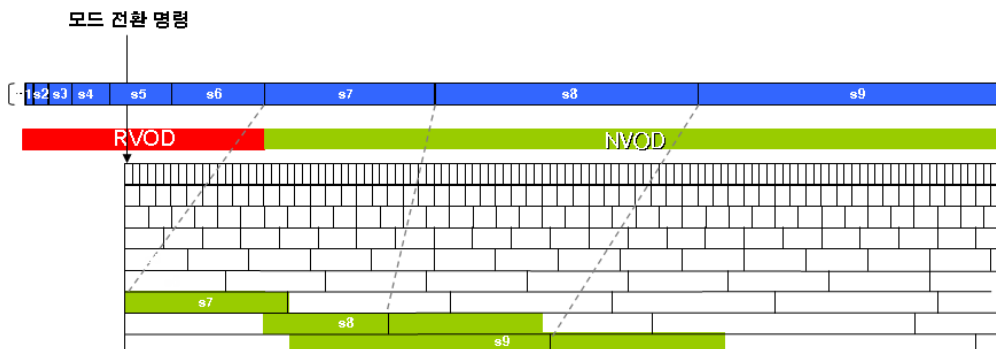
RVOD와 NVOD가 전환되는 예시를 시간-대역폭 그림을 통해 설명한다.

3.2.1 RVOD에서 NVOD로의 전환

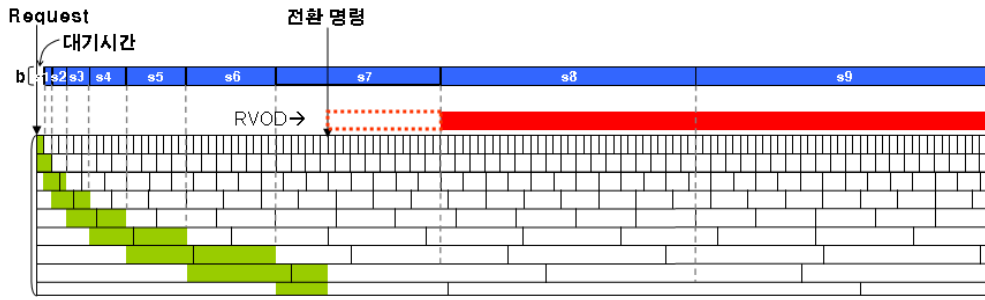
(그림 9)는 RVOD로 서비스 중인 VOD 서버에서 수요가 급증하여 NVOD로의 모드 전환 명령이 내려졌을 때 NVOD(Flexible Broadcasting)로 전환되는 타이밍을 나타낸 클라이언트의 시간-대역폭 그림이다. 이때 서버는 전환될 NVOD 서비스를 새롭게 생성하고 모드전환 메시지를 접속되어 있던 모든 클라이언트에게 전달한다. 모드 전환 메시지를 받은 클라이언트는 자신이 NVOD로 받아야 할 시점을 계산하여 해당 시간에 맞는 세그먼트를 전송해달라고 요청하게 된다. 예를 들어 Flexible broadcasting으로 서비스 중인 서버에 접속하여 s5에 해당하는 부분을 RVOD로 재생하고 있는 클라이언트가 있다고 가정하자. 이때 서버로부터 모드전환명령이 내려진다면, 서버에서는 새로운 NVOD 서비스를 시작하게 되며, (그림 9)와 같이 명령이 내려진 시점이 NVOD 전체 채널의 시작부분이 될 것이다. 만약 모드 전환명령이 내려진 시점의 바로 다음 세그먼트(그림 9의 s6)부터 버퍼에 저장한다면, s7이 재생되어야 할 시점이 되어서는 s7의 시작부분이 버퍼에 저장되어 있지 않아 재생이 중단되는 경우가 발생하게 된다. 이러한 상황을 방지하고자 Flexible broadcasting을 사용한 RVOD에서 NVOD로의 전환은 재생중인 세그먼트보다 2번째 뒤의 세그먼트부터 버퍼에 저장을 하도록 하여야 한다. Conventional broadcasting은 데이터를 분할하지 않은 방식이므로 RVOD로 서비스를 받던 클라이언트가 NVOD로 전환하여 스트림을 받을 수가 없고, 전환명령 이후에 접속하는 클라이언트에게만 NVOD로 스트림을 전송한다.

3.2.2 NVOD에서 RVOD로의 전환

(그림 10)은 NVOD로 서비스 중인 VOD 서버에서 수요가



(그림 9) RVOD에서 NVOD로의 모드 전환 예시 시간-대역폭 그림



(그림 10) NVOD에서 RVOD로의 모드 전환 예시 시간-대역폭 그림

급감하여 RVOD로의 모드 전환 명령이 내려졌을 때 NVOD(Flexible broadcasting)로 전환되는 타이밍을 나타낸 클라이언트의 시간-대역폭 그림이다. 이때 서버는 전환될 RVOD 서비스를 새롭게 생성하고 모드전환 메시지를 접속되어 있던 모든 클라이언트에게 전달한다. 모드 전환 메시지를 받은 클라이언트는 자신이 RVOD로 받아야 할 시점을 계산하여 해당 시점이 되었을 때 서버에게 전송 요청을 하게 된다. 예를 들어 (그림 10)과 같이 클라이언트가 NVOD로 서비스 받고 있고, s7을 재생 중일때 VOD 서버로부터 모드 전환명령이 내려졌다고 가정하자. 재생 초기부터 NVOD로 서비스를 받고 있었기 때문에 s7이 재생중이라면 버퍼에는 s7이 다 들어있다고 볼 수 있다. 그러므로 NVOD에서 RVOD로 전환할 때, 서버에서 클라이언트에게 RVOD를 전송하는 타이밍은 클라이언트의 버퍼에 들어있는 비디오의 재생이 다 끝난 시점, 즉 모드 전환 명령이 들어온 시점의 다음 세그먼트부터 RVOD로 전송을 하게 된다. Conventional broadcasting의 경우 버퍼에 따로 저장해 놓지 않기 때문에 모드 전환 명령이 들어올 경우 현재 재생시점을 서버에게 알려주고 바로 모드 전환을 할 수 있다.

위의 두 가지 전환 예시를 통해 프로토콜의 전환 방법과 실제 적용 예를 알아보았다. 여기서 알 수 있는 점은 RVOD로 서비스중인 서버에서 증가하는 클라이언트의 수요로 인해 한계에 가까워진 서버의 대역폭을 하나의 채널에 여러 명의 클라이언트가 접속할 수 있는 NVOD로 전환을 한다면 적은 대역폭으로도 많은 클라이언트에게 서비스를 할 수 있으므로 서버의 과부하를 방지하는 결과를 가져올 수 있고, NVOD로 서비스 중인 서버에서 클라이언트의 수가 급감하여 RVOD로 서비스하는 것이 서버의 대역폭을 아낄 수 있는 방안이라면 RVOD로 전환하여 대역폭의 낭비를 막는 결과를 가져올 수 있다는 것을 알 수 있다. 따라서 본 논문에서 제안한 프로토콜을 적용시키면 클라이언트의 수요에 따라 모드 전환을 통하여 서버의 대역폭을 효율적으로 관리할 수 있으므로 최적의 VOD서비스를 제공할 수 있다. 또한 RVOD/NVOD 통합 모드 전환 프로토콜은 RVOD 서비스와 다양한 방식의 NVOD 서비스에 적용이 가능하므로 기존의 연구 방법들을 그대로 활용하여 사용할 수 있다는 장점도 있다.

4. 성능 분석

본 논문에서 소개하는 프로토콜은 기존의 RVOD, NVOD 서비스들에도 적용할 수 있다. 따라서 클라이언트의 수신 요청 빈도, VOD 서버의 최대 대역폭 등 몇 가지 환경을 설정하고, 설정된 각 환경에서 기존의 방법들과 RVOD/NVOD 통합 모드 전환 프로토콜을 사용한 경우의 성능을 이론적으로 비교하고 분석하였다. 성능은 설정된 환경에서 서버에 필요한 네트워크 대역폭을 비교해서 분석하였다.

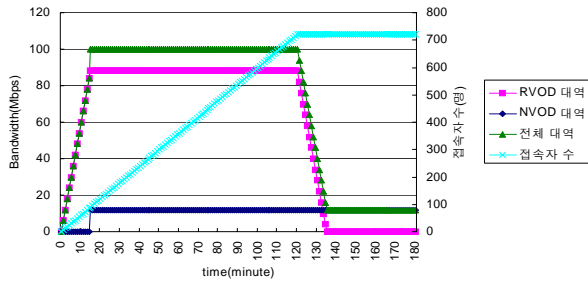
4.1 클라이언트의 수요가 많은 경우

본 절에서는 RVOD에서 NVOD로 전환하는 경우 중 시간 분할 방식인 Conventional broadcasting과 데이터분할 방식인 Flexible broadcasting 방식, 이 두 가지 NVOD 방식으로의 전환을 비교 분석한다.

4.1.1 Conventional broadcasting으로의 전환

(그림 11)의 그래프는 복수의 클라이언트가 RVOD에서 NVOD(Conventional broadcasting)로 전환되는 예시이다. (그림 11)을 참조하여 모드 전환에 따른 VOD 서버의 대역폭을 설명한다. (그림 11)의 그래프는 클라이언트로부터 VOD 서버에 VOD 서비스 요청이 10초에 하나씩으로 수요가 급증하는 경우를 나타낸 예시이다. (그림 11)에서 ‘*’ 모양의 그래프는 동일한 콘텐츠에 대한 VOD 서비스 접속을 요구하는 사용자 수를 나타내며 본 실험에서는 사용자 수가 10초마다 1명씩 계속적으로 증가하여 RVOD의 대역폭이 전체 사용자에게 대한 서비스를 수용할 수 없을 정도까지 증가시켰다. ‘■’ 모양의 그래프는 RVOD 모드로 서비스 되는 VOD 서버의 대역폭을 나타낸다. ‘◆’ 모양의 그래프는 Conventional Broadcasting 방식의 NVOD 모드로 서비스 되는 VOD 서버의 대역폭을 나타낸다. ‘▲’ 모양의 그래프는 RVOD와 NVOD 모드의 대역폭을 합친 전체 VOD 서버의 대역폭을 나타낸다.

VOD 서버의 최대 대역폭은 100Mbps이고, 전송률 1Mbps의 비디오 1개를 서비스하고 NVOD의 대기시간은 10분이라고 가정하였을 때, VOD 서비스를 요청하는 단말의 수가 증가하여 VOD 서버의 한계 대역폭에 다다른 경우(그림 11의



(그림 11) RVOD에서 NVOD(Conventional Broadcasting)로의 전환

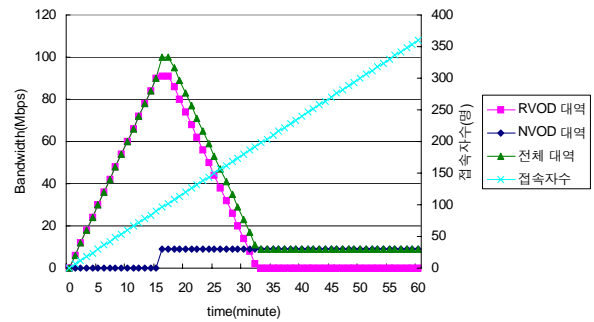
그래프에서는 14분 40초이다.) 모드 전환을 하지 않는다면 수 분 이내에 VOD 서버의 한계대역폭을 초과하여 서비스를 할 수 없는 상황이 될 것이다. 이러한 경우에 VOD 서버는 RVOD 모드에서 NVOD 모드로 전환하여 대역폭을 감소시킬 수 있다. (그림 11)에서 모드 전환은 14분 40초에 일어났으며 모드 전환 이후에는 더 이상 RVOD 접속이 없기 때문에 RVOD 대역폭은 증가하지 않는다. 그러나 모드 전환 이후 NVOD 채널이 새롭게 설정되어야 하기 때문에 전체 대역폭은 NVOD 채널 만큼 순간적으로 급증한다. 시간이 지나면서 접속자의 수는 계속 증가하지만, 새롭게 접속한 클라이언트는 NVOD로 접속되어서 전체 대역폭은 더 이상 증가하지 않는다. 일정 시간이 지난 후에 RVOD 모드로 서비스를 받던 복수의 클라이언트가 점차적으로 NVOD로 전환됨에 따라 VOD 서버에는 NVOD 서비스만 남게되어 전체적인 대역폭이 감소함을 알 수 있다.

4.1.2 Flexible Broadcasting으로의 전환

(그림 12)의 그래프는 네트워크 대역폭과 저장 공간이 충분 한 경우의 클라이언트가 RVOD에서 NVOD(Flexible broadcasting)로 전환되는 예시이다. (그림 12)의 그래프는 클라이언트로부터 VOD 서버에 VOD 서비스 요청이 10초에 하나씩 있고, VOD 서버의 최대 대역폭은 100Mbps, 전송률 1Mbps의 비디오 1개를 서비스하고 NVOD의 대기시간이 1 분이라고 가정한 상황에서 시뮬레이션 하였다.

(그림 12)에서 ‘*’ 모양의 그래프는 동일한 콘텐츠에 대한 VOD 서비스 접속을 요구하는 사용자 수를 나타내며 본 실험에서는 사용자 수가 10초마다 1명씩 계속적으로 증가시켰다. ‘■’ 모양의 그래프는 RVOD 모드로 서비스 되는 VOD 서버의 대역폭을 나타낸다. ‘◆’ 모양의 그래프는 Flexible Broadcasting 방식의 NVOD 모드로 서비스 되는 VOD 서버의 대역폭을 나타낸다. ‘▲’ 모양의 그래프는 RVOD와 NVOD 모드의 대역폭을 합친 전체 VOD 서버의 대역폭을 나타낸다.

대기시간은 4.1.1의 설정과 다르지만 서버의 대역폭의 효율을 보기 위함이므로 서버의 최대 대역폭과 서비스하는 비디오의 개수에 초점을 둔다. 시뮬레이션의 결과는 4.1.1의 결과와 마찬가지로 RVOD 모드에서 NVOD 모드로 전환됨에



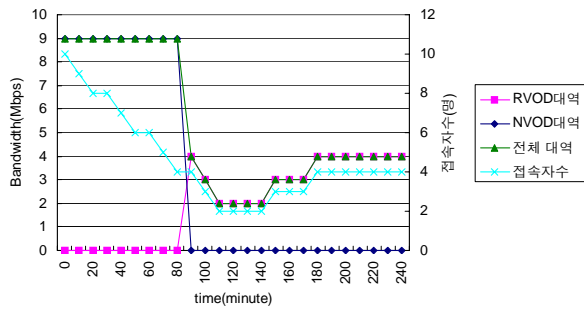
(그림 12) RVOD에서 NVOD(Flexible Broadcasting)로의 전환

따라 VOD 서버의 전체 대역폭이 감소되는 것을 확인 할 수 있다. VOD 서버에서 사용되는 대역폭이 줄어들기까지의 시간이 두 NVOD 방식 간에 차이가 있는 것을 확인할 수 있다. 이 차이는 Conventional broadcasting의 경우 시분할 방식이기 때문에, RVOD로 서비스를 받던 기존의 클라이언트가 모드 전환 이후 새롭게 서비스되는 NVOD 채널에 접속을 하지 못하므로 모드 전환 시점 이전에 접속한 클라이언트는 감상중인 비디오가 재생이 끝날 때까지 RVOD를 통해 서비스를 받아야 하는 데서 원인을 찾을 수 있다. 이를 통해 본 논문에서 제안한 프로토콜을 이용하여 RVOD에서 NVOD로 전환할 경우, 시간분할 방식의 NVOD 방식으로의 전환보다 데이터 분할 방식의 NVOD로의 전환이 보다 빠른 전환을 할 수 있고, 그를 통해 서버의 대역폭 감소를 빠르게 가져 올 수 있다는 것을 알 수 있다.

4.2 클라이언트의 수요가 적은 경우

(그림 13)의 그래프는 복수의 클라이언트가 NVOD (Flexible broadcasting)에서 RVOD로 전환되는 예시이다. 환경 설정은 4.1의 설정과 마찬가지로 비디오 길이 120분, 전송률 1Mbps의 비디오 한 개만 서비스하는 것으로 설정한 상태에서의 시뮬레이션이다. 4.1.2에서의 NVOD 설정과 같이 대기시간은 1분이며 이로 인해 9개의 채널이 만들어진다. 시뮬레이션 시작시에는 10명이 접속을 하고 있던 상태이고, 10분마다 1명씩 접속이 종료되며, 30분에 1명씩 접속이 들어 온다고 가정한 경우의 그래프이다. (그림 13)에서 ‘*’ 모양의 그래프는 동일한 콘텐츠에 대한 VOD 서비스 접속을 요구하는 사용자 수를 나타내며 처음에는 10명의 동시 사용자가 접속을 하고 있었으나 시간이 지남에 따라 사용자 수는 점차 줄어 2~4명 정도가 되도록 하였다. ‘■’ 모양의 그래프는 RVOD 모드로 서비스 되는 VOD 서버의 대역폭을, ‘◆’ 모양의 그래프는 Flexible Broadcasting 방식의 NVOD 모드로 서비스 되는 VOD 서버의 대역폭을, ‘▲’ 모양의 그래프는 RVOD와 NVOD 모드의 대역폭을 합친 전체 VOD 서버의 대역폭을 각각 나타낸다.

처음에 10명의 동시 사용자가 동일한 비디오를 보고 있어서 Flexible Broadcasting 방식의 NVOD 모드만으로도 VOD 서비스가 시작되어 9Mbps의 서버 대역폭을 차지하고 있었



(그림 13) NVOD(Flexible Broadcasting)에서 RVOD로의 전환

다. 그러나 사용자 수가 점점 감소하여 약 80분 뒤에는 남아있는 사용자 수가 4명으로 줄었다. 이 경우 NVOD로 계속 서비스 한다고 가정하면 서버에서는 9Mbps를 계속 할당하여야 하지만, RVOD로 전환할 경우 약 4Mbps의 대역을 서버에서 할당하게 되는 것을 볼 수가 있다. 이렇듯 클라이언트의 수요가 적은 상황에서는 RVOD로 서비스 하는 것이 대역폭의 낭비를 막을 수 있기 때문에 VOD 서버는 NVOD로 도중에(그래프에서는 90분에 모드 전환이 이루어짐) 모드 전환이 되어 서비스 하도록 하는 것이다. 모드 전환 이후에는 NVOD 서비스는 중단되고 RVOD 만으로만 서비스가 되기 때문에 서버의 대역폭은 RVOD에 접속된 사용자수에 따라 2~4Mbps를 유지하고 있는 것을 볼 수 있다.

(그림 13)의 그래프는 Flexible broadcasting에서 RVOD로의 전환을 나타낸 그래프이지만, RVOD에서의 재생은 재생시점을 포함한 PLAY메시지로 원하는 부분부터 곧바로 재생이 가능하기 때문에, NVOD에서 RVOD로의 전환은 서비스 하고 있는 NVOD의 방식에 관계없이 전환 명령이 내려진 시점에 바로 전환이 가능하다. 그러므로 다른 방식의 NVOD에서 RVOD로 전환하는 시뮬레이션은 생략하였다.

5. 결 론

이전에 연구 되었던 방법들은 비디오를 시청하는 클라이언트의 수요를 고려하지 않은 방법들이 많았다. 그에 따라 발생하는 문제를 해결하기 위해 본 논문에서는 효율적인 VOD 서비스를 위한 프로토콜에 관한 연구를 수행하였다. 기존의 VOD 서비스 알고리즘을 적용할 수 있는 새로운 프로토콜을 제안하고, 시뮬레이션을 통해 성능을 분석하였으며, 새로운 프로토콜을 적용시킨 서버/클라이언트 구조의 시스템을 개발하여 시뮬레이션의 결과를 직접 확인하였다.

본 논문에서 제안한 프로토콜의 특징으로는 RVOD, NVOD 모두 제어가 가능한 점, 기존의 RVOD, NVOD 방식을 적용할 수 있다는 점, 두 방식간의 모드 전환을 통해 효율적인 대역폭 관리를 할 수 있다는 점을 들 수 있다.

이와 같이 본 논문에서 제시한 RVOD/NVOD 통합 모드 전환 프로토콜은 기존의 VOD 전송에 관련된 알고리즘들을 활용하면서도 서비스 모드의 전환을 통해 서버의 대역폭을

클라이언트의 수요에 따라 조절할 수 있게 함으로써 VOD 인프라 구축비용 등의 사업비용을 절감할 수 있다. 또한 사업비용 절감으로 인해 잉여자본을 양질의 콘텐츠 확보에 투자할 수 있어서 VOD의 수요 또한 확대 시킬 수 있을 것으로 기대된다. 예를 들어 최근 주목받고 있는 UCC(User Created Contents)동영상과 같은 서비스의 경우, 본 프로토콜을 적용하면 서비스 업체에 파일을 올리지 않고도 네트워크 자원이 비교적 한정적인 일반 가정에서도 개인용 PC를 활용하여 효율적으로 사용할 수 있을 것이다.

향후 연구과제로는 모드 전환하는 최적의 타이밍을 정하는 기법, 수요 예측기법, RTSP와의 호환문제 등이 있으며, 관련연구에서 언급한 NVOD 방식 외의 기타 NVOD 방식에도 모두 적용시킬 수 있도록 프로토콜을 확장시켜야 한다.

참 고 문 헌

- [1] H. Schulzrinne, A. Rao, and R. Lanphier. 『Real time streaming protocol (rtsp)』, In RFC2326, Apr., 1998.
- [2] K. c. Almeroth and M. H. Ammar. "The use of multicast delivery to provide a scalable and interactive video-on-demand service," *IEEE Journal on Selected Areas in Communications*, 14(5):1110-22, Aug., 1996.
- [3] S. Viswanathan and T. Imielinski. "Pyramid Broadcasting for video on demand service". In *IEEE Multimedia Computing and Networking Conference*, Volume 2417, pp 66-77, San Jose, California, 1995.
- [4] L. Juhn and L. Tseng. "Fast data broadcasting and receiving scheme for popular video service". In *IEEE Transactions on Broadcasting*, 44(1):100-105, Mar., 1998.
- [5] 정민성. 『광대역통합망에서 다양한 단말환경에 대한 효과적인 NVOD 서비스』, 석사학위논문. 중앙대학교. 2005년 12월.
- [6] <http://www.koren21.net/> - 광대역통합 연구개발망
- [7] <http://www.videolan.org/> - VideoLan VLC media player
- [8] Michael K. Bradshaw, Bing Wang. "Periodic broadcast and patching services - implementation, measurement, and analysis in an internet streaming video testbed," In *ACM Multimedia 2001*: 280-290.
- [9] C. C. Aggarawal, J. L. Wolf, and P. S. Yu. "A permutation-based pyramid broadcasting scheme for video-on-demand systems," In *Proc. of the IEEE Int'l conf. of Multimedia Computing and Systems '96*, Hiroshima, Japan, June, 1996.
- [10] R. Fielding and et. al. 『HTTP:HyperText Transfer Protocol - HTTP/1.1』 RFC 2616, June, 1997.
- [11] A. Walsh and M. Bourges-Sevenier, 『MPEG-4 Jump Start』, Prentice-Hall, 2002.
- [12] Lain E.G Richardson, 『H.264 and MPEG-4』, Wiley&Sons Ltd.



김 명 훈

e-mail : rhymer9.kim@samsung.com
2005년 중앙대학교 전자전기공학부(학사)
2007년 중앙대학교 대학원 전자전기공학부
(공학석사)
2007년~현 재 도시바삼성스토리지테크
놀러지코리아 주식회사 연구원

관심분야 : 멀티미디어 스트리밍, 임베디드 소프트웨어 등



박 호 현

e-mail : hohyun@cau.ac.kr
1987년 서울대학교 계산통계학과(학사)
1995년 한국과학기술원 정보및통신공학과
(공학석사)
2001년 한국과학기술원 전자전산학과
(공학박사)

1987년~2002년 삼성전자 통신연구소 수석연구원
2003년~2006년 중앙대학교 전자전기공학부 조교수
2007년~현 재 중앙대학교 전자전기공학부 부교수
관심분야 : 멀티미디어 스트리밍, 멀티미디어 검색, LBS,
Ubiquitous Computing 등