

# 네트워크 부하 기반 프레임 생략 전송 알고리즘

정홍섭<sup>†</sup> · 박규석<sup>\*\*</sup>

## 요 약

다수의 사용자 요구에 의해 비디오 타이틀을 실시간으로 제공해야 하는 VOD 서비스는 클라이언트의 버퍼 안정화와 재생 질 보증을 위해서, 네트워크의 상태에 따라 드로핑(Dropping)이나 스킵핑(Skipping) 알고리즘으로 프레임을 전송하는 메커니즘이 필요하다. 본 논문에서는 네트워크의 부하에 따라 저장 장치에서 생략 인출한 MPEG 프레임(I, P, B 프레임)을 클라이언트에 전송하는 알고리즘을 제시한다. 또한 시뮬레이션을 통해 네트워크의 부하에 적응적으로 대처하여 네트워크의 부하를 줄이고 클라이언트의 수신 버퍼를 안정화시킬 수 있음을 검증한다.

## A Frame Skipping Transfer Algorithm based on Network Load

Hong-Seup Jeong<sup>†</sup> and Kyoo-Seok Park<sup>\*\*</sup>

## ABSTRACT

To guarantee client buffer stabilization and visual quality, the VOD service that provides real time video titles on requirements of numerous users, needs a mechanism which transfers frames with dropping or skipping algorithm by network condition. In this paper, we show an algorithm that transfers withdrawn skipped MPEG frames(I, P, B frame) from disk to client dependent on network load. Moreover, we verify through a simulation that adaptive dealing on network load can reduce the network load and stabilize client receiving buffer.

**Key words:** network load, skipping transfer, buffer stabilization

## 1. 서 론

VOD 서비스의 특성은 장시간의 전송 및 재생 시간을 요구하기 때문에, 고성능의 VOD 서버 및 고속의 네트워크를 통해 MPEG 비디오 프레임을 전송하더라도 프레임의 높은 전송 대역폭과 다수의 사용자, 네트워크의 과부하, 경부하 및 에러로 인하여 클라이언트는 일정치 않은 시간 간격으로 프레임을 수신하게 되거나 수신하지 못하게 되어 클라이언트의 수신 버퍼 상태도 수시로 변하게 된다. 이러한 버퍼 상태 변화로 인하여 클라이언트 버퍼에서는 스태이션

또는 오버런이 발생할 수도 있고, 최악의 경우 버퍼 넘침이 생겨 프레임을 분실하거나 프레임이 계속적으로 도착하지 않아 버퍼 고갈이 발생하여 재생 화상의 질이 저하되거나 단절되어 실시간 VOD 서비스의 원 목적을 상실할 수 있다[1]. 클라이언트의 버퍼 안정화와 재생 질 보증을 위해서는 네트워크의 상태를 주기적으로 모니터링하여 그 상태에 따라 디스크 배열에 저장되어 있는 프레임을 드로핑(Dropping)이나 스킵핑(Skipping) 알고리즘으로 프레임을 전송할 수 있는 메커니즘이 필요하다[2].

본 논문에서는 MPEG 부호화기[3]를 이용하여 일반 VHS 비디오 테이프의 내용을 손실 압축하여 디스크 배열에 저장해 두고, 디스크 배열에 압축 저장된 MPEG 프레임(I, P, B 프레임) 데이터를 네트워크의 부하에 따라 저장장치에서 생략 인출하여 클라이언트

본 논문은 2003년도 경남대학교 학술연구비 지원에 의해 수행되었음.

접수일 : 2003년 5월 9일, 완료일 : 2003년 8월 4일

<sup>†</sup> 정회원, 경남대학교 컴퓨터공학과 박사

<sup>\*\*</sup> 종신회원, 경남대학교 정보통신공학부 교수

언트에 전송하는 알고리즘을 제시한다.

또한 시뮬레이션을 통해 제안 프레임 생략 전송 알고리즘이 네트워크의 부하에 적응적으로 대처하여 클라이언트의 수신 버퍼를 안정화할 수 있음을 검증한다.

## 2. 관련연구

### 2.1 ACME의 생략 알고리즘

비디오 및 오디오와 같은 연속 미디어들은 일정한 속도로 처리되거나 출력이 요구된다. 즉, 연속적인 데이터 공급을 위해서는 서버의 저장 시스템 및 CPU, 통신 시스템, 각각의 출력 장치에서 비디오 데이터가 연속적으로 처리되어야 하는 문제점을 가지고 있다. 이러한 문제점을 End-to-End 연속성 보장 문제라고 하며, 이 문제를 해결하기 위해 ACME(Abstraction for Continuous MEdia)에서는 시간 제약을 어기는 경우, 전송될 프레임의 일부를 스킵핑(Skipping) 혹은 퍼징(Pausing)하는 기법을 사용하고 있다 [4,5].

이 기법은 LTS(Logical Time Stamp)와 논리장치의 타임 스탬프 사이에서 왜곡현상이 상한치를 넘지 않는 경우에는 현재 가지고 있는 프레임을 스킵핑(skipling)하여 생략함으로써 LTS를 따라 잡을 수 있다. 또한 왜곡현상이 계속적으로 상한치를 벗어날 경우 LTS 자체를 잠시 정지시켜 논리장치의 타임 스탬프를 동기화시킨 후 재 동작하는 기법을 사용하고 있다. 하지만 이 기법은 네트워크의 상태 변화에 따른 수신측 논리장치의 상태를 고려하지 않은 단점이 있다.

### 2.2 QoS에 의한 생략 전송 알고리즘

VOD 서비스가 시작될 때, 그 당시의 서버 시스템의 부하를 측정하여, 제공할 비디오의 타이틀에 따라 적절한 서비스 품질을 결정하고, 그 서비스 품질에 맞도록 서비스 파라미터를 결정해야 한다. 파라미터를 결정하는 방식으로는 서비스가 종료될 때까지 조정된 파라미터에 의해 서비스가 이루어지도록 하는 정적 협상 방식과, 서비스 수행 중간에 서비스 품질이 바뀔 수 있도록 하는 동적 협상 방식이 있다. 동적 협상 방식에는 QoS(Quality of Service) 관리자가 주기적으로 시스템의 부하상태를 점검하여, 필요한 경

우 서비스 관리자에게 서비스 품질 협상을 요구하는 운영체제에 의해 주도되는 방식과, 서비스 수행 중간에 QoS 관리자에게 시스템의 부하 상태를 질의하여, 상황에 따라 서비스 품질을 하향 혹은 상향 조정하는 서비스 관리자에 의해 주도되는 방식이 있다[6,7]. 이 알고리즘의 효율성은 서버가 지원하는 클라이언트의 수에 의해 결정된다는 제약조건이 따른다.

### 2.3 블록 드로핑에 의한 생략 전송 알고리즘

VOD 시스템과 같이 이미 압축 저장된 비디오를 전송함에 있어서 네트워크의 대역폭이 감소되는 경우, 서버에서 압축된 비디오의 비트율을 감소시켜 전송해야 할 경우가 종종 발생한다. 기존의 방법[9]에서는 먼저 비디오를 디코딩하고, 아주 낮은 비트율로 이것을 재압축하는 방식을 사용하고 있으나, MPEG은 디코더의 기능보다 인코더에 더 많은 처리 능력을 발휘하므로, 이 방법은 높은 비대칭성과 전송 가능한 비율의 파라미터 수를 제한해야 하는 단점을 가지고 있다. [8,9]에서는 압축된 도메인 상에서 단순히 비율 재조정(Rate Shaping)을 수행하고 있고, [10]에서는 사전 압축된 MPEG 비디오의 전송 비율 재조정을 위해 I, P, B 프레임 별로 블록을 생략 전송하는 블록 드로핑 알고리즘에 대해 제안하고 있다. 여기서 블록은 프레임을 구성하는 매크로 블록을 의미하며, 블록의 생략 전송시 흰 블록을 모두 전송하면 클라이언트 측에서 검은 블록을 복원하는 방법을 사용하고 있다.

### 2.4 HeiTS의 생략 전송 알고리즘

HeiTS(Heidelberg Transport System)[5]의 미디어 스케일링(media scaling) 알고리즘은 전송단계(transport level)에서 수행되며, 현재의 네트워크 상태를 파악하여 유용한 대역폭으로 트래픽을 적용시키는 네트워크의 응답성에 중점을 두고 있다. 즉 얼마나 빨리 트래픽을 가용 대역폭에다 적용시키느냐 하는 것이 문제이며, 네트워크 부하 정도에 따라 스케일링 업(scaling up)과 스케일링 다운(scaling down)을 수행한다. 모든 데이터 스트림은 HeiTS에 연결되어 전송되기 때문에 이 시스템의 QoS 파라미터 셋트에 의해 데이터 스트림의 전송율을 알 수 있다. 여기서 데이터 스트림의 전송율은 HeiTS의 수신측에서 매 패킷(packet)들의 도착 상태를 모니터링하기 위해 사용된다. 그 예로서 어떤 패킷이 수신측에

늦게 도착한다는 것은 네트워크가 과부하 상태에 있음을 나타낸다. 이 경우 수신측은 네트워크의 과부하 상태를 전송측에 알리면 전송측은 스트림을 스케일링 다운시켜 수신측에 전송하고, 과부하 상태가 끝나면 다시 스트림을 스케일링 업시켜 스트림을 전송한다. 그러나 이 기법은 미디어 스케일링 알고리즘을 너무 늦게 적용할 경우 수신측 시스템은 오버로드 상태에 있을 수 있으며, 반면에 너무 빨리 적용할 경우는 네트워크의 일시적인 지연에도 불구하고 스케일링을 행하여 미디어 스트림의 손실을 유발하는 단점을 가지고 있다.

3. 제안 생략 전송 알고리즘

제안 알고리즘을 위한 VOD 시스템의 구성도는 그림 1과 같으며 단일 서버와 다수의 클라이언트들이 고속의 네트워크를 통해 서로 연결되어 있고 저장장치로는 수많은 타이틀을 저장할 수 있으며 높은 전송 대역폭을 제공하는 디스크 배열을 사용한다.

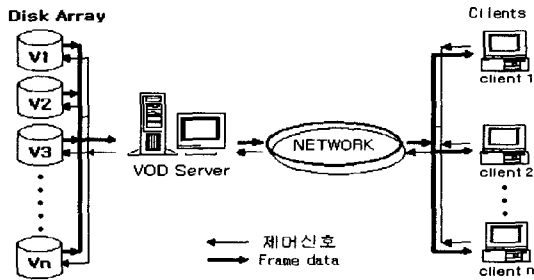


그림 1. VOD 시스템 구성도

본 논문의 구성을 위해, 먼저 그림 2와 같이 일반 VHS 비디오 타이틀을 MPEG으로 GOP가 15(N=15, M=3)인 프레임은 생성하여 각 디스크에서의 각 프레임이 같은 장소에 위치되도록 동기화되어 디스크 배열에 저장하였음을 전제로 한다.

이와 같은 작업들은 서비스 마감시간 전까지 진행되고 마감시간이 완료되면 적재된 버퍼의 내용들이 라운드 로빈 알고리즘에 의해 각 클라이언트에 전송된다. 서비스 중에 신규 요청이 접수되면 위의 작업과 마찬가지로 승인여부의 판단과 승인이 이루어졌다면 다음 주기의 마감시간에 도달한 다른 클라이언트들과 함께 서비스가 이루어진다.

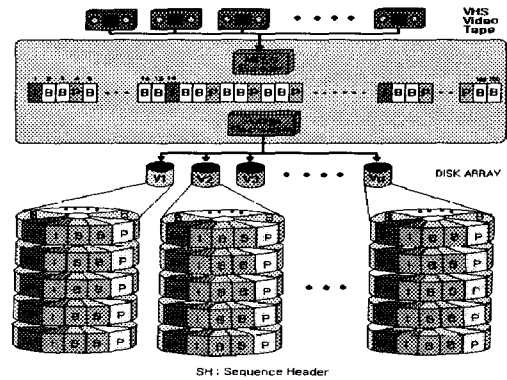


그림 2. MPEG 프레임의 저장 상세도

서버의 내부 구성은 그림 3과 같으며 서버는 클라이언트의 서비스 요청에 대해 서비스 제약사항을 위반하지 않는다면 그 요청을 수락한다. 다음으로 서버 스케줄러는 디스크 관리기로부터 저장되어 있는 비디오 정보를 관리하는 비디오 인덱스 레코드를 검색하여 비디오 타이틀의 위치를 확인하여 서비스할 비디오 프레임은 디스크 배열에서 서버 버퍼로 인출한다.

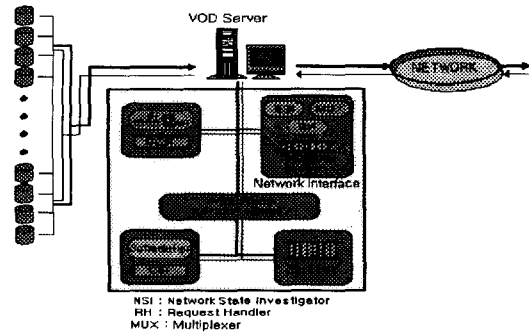


그림 3. 서버의 내부 구성도

디스크 관리기(Disk Manager)는 각 클라이언트에서 요청한 비디오 타이틀의 저장여부 판단과 디스크 제어 레코드(Disk Control Record : DCR)에 현재 서비스 중인 비디오와 해당 디스크의 섹터의 정보와 검색회수를 관리한다.

서버 스케줄러는 제어 신호를 접수하여 디스크 관리기에 알리고, 디스크 관리기는 접수된 제어신호에 따라 디스크 제어 레코드에 기록하여 주기마다 이 디스크 제어 레코드의 내용에 따라 디스크 배열 저장장치에서 프레임을 서버 버퍼에 인출한다.

네트워크 인터페이스(Network Interface)는 IP 주

소(Address) 및 프레임 전송 등에 대한 내용을 관리하고, 주기적으로 네트워크의 부하정도를 측정하여 서버 스케줄러에 알린다. 여기에는 네트워크 상태 검사기(Network State Investigator), 요구 처리기(Request Handler), 전송 제어 다중화기(MUX) 및 네트워크 버퍼가 있다. 네트워크 상태 검사기(NSI)는 클라이언트의 패킷 수신량 정도에 대한 정보를 분석하여 서버 스케줄러에 알린다. 이때 네트워크의 부하가 60%이상이면 서버 스케줄러는 프레임 생략 전송 알고리즘에 따라 생략 전송할 프레임을 제외한 프레임을 버퍼에 인출하여 전송한다. 요구 처리기는 서비스 중인 클라이언트의 IP 주소를 관리하고, 현재 서비스 중인 네트워크의 과부하로 클라이언트의 버퍼 고갈 시 다른 라우터 및 스위치의 경로를 설정하여 해당 클라이언트에 프레임을 전송하는 역할을 한다. 네트워크 버퍼는 주기마다 서비스 마감시간이 지난 서버 버퍼의 프레임을 읽어와 전송 제어 다중화기에 보낸다. 전송 제어 다중화기는 네트워크 버퍼에 있는 프레임을 각 클라이언트에 분배하는 역할을 한다.

### 3.1 네트워크의 부하에 의한 프레임 인출

#### 3.1.1 네트워크 상태 검사기의 신호

네트워크의 상태변화를 주기적으로 감시하는 네트워크 상태 검사기는 해당 클라이언트의 버퍼에 초당 수신되는 프레임의 양과 최대 수신량으로 부하를 측정한다<sup>1)</sup>.

이 부하에 따라 제어 신호를 서버의 디스크 제어 레코드에 기록한다. 서버 스케줄러는 디스크 제어 레코드(Disk Control Record : DCR)의 내용에 따라 해당 디스크에서 프레임을 인출하여 서버 버퍼에 보낸다.

각 클라이언트에 연결되어 있는 네트워크의 상태를 검사한 후 그 부하가 60%이상일 때는 서버 스케줄러에 1001(Net1)의 신호를 보내고, 부하가 70% 이상일때는 1010(Net2)의 신호를 보낸다. 또한 부하가 80% 이상일때는 1011(Net3)의 신호를 보내고, 부하가 90% 이상일 때는 1110(Net4)의 신호를 보낸다. 신호를 수신한 서버 스케줄러는 그 내용을 디스크 제어 레코드의 네트워크 상태(Net-state) 필드에 기록한다. 표 1은 네트워크 상태 검사기의 제어 신호를 나타낸다.

1) Visual c++로 작성된 네트워크 상태 감시기에 의함

표 2는 디스크 제어 레코드의 예로서 C<sub>1</sub>, C<sub>3</sub>, C<sub>4</sub>, C<sub>n-1</sub>, 그리고 C<sub>n</sub>은 클라이언트에 연결된 네트워크의 상태를 보여 준다. 즉 C<sub>1</sub> 클라이언트는 네트워크의 부하가 70% 이상에 속한 경우이고, C<sub>3</sub> 클라이언트는 네트워크의 부하가 80% 이상에 속한 경우이다.

표 1. 네트워크 상태 감시기의 제어 신호

제어 신호	표시	용도
1001	Net1	60% 이상의 부하 (일의 자리가 3의 배수 (3,6,9,...)인 B 프레임 생략)
1010	Net2	70% 이상의 부하 (모든 B프레임 생략)
1011	Net3	80% 이상의 부하 (모든 B 프레임과 홀수 P 프레임 생략)
1110	Net4	90%이상의 부하 (B, P 프레임 모두 생략)

표 2. 디스크 제어 레코드 상태

Client	C1	C2	C3	C4	Ci	Cn-1	Cn
Net-state	Net2		Net3	Net1		Net1	Net4

그리고 C<sub>4</sub>과 C<sub>n-1</sub> 클라이언트는 네트워크의 부하가 60% 이상에 속한 경우이고, C<sub>n</sub> 클라이언트는 네트워크의 부하가 90% 이상인 상황을 보이고 있다.

#### 3.1.2 프레임 인출 알고리즘

네트워크의 부하에 의한 프레임 인출 알고리즘은 표 2의 디스크 제어 레코드의 상태에 따라 표 3처럼 프레임을 생략하여 인출한다. 표 3에서 GOP 파라미터는 N=15이고 M=3인 1개의 GOP에서 네트워크의 부하에 따라 프레임이 생략되어 인출되는 상태를 보여준다. 여기서 N과 M은 생략할 프레임의 번호가 되는 파라미터 또는 인출하지 않아도 될 주기의 파라미터로서 활용할 수 있다.

표 3에서 네트워크의 부하가 60%이하일 경우는 정상적이므로 프레임을 생략하지 않고 모든 프레임을 매 주기마다 저장장치에서 인출하여 클라이언트에 전송하고, 네트워크의 부하가 60% 이상일 경우에는 M=3이 되는 주기에 해당되는 프레임과 3의 배수가 되는 주기마다 B 프레임을 저장장치에서 생략하여 인출한다. 따라서 해당 클라이언트에는 3의 배수가 되는 B 프레임을 제외한 모든 프레임이 인출된다.

또한 네트워크의 부하가 70% 이상일 경우에는 모든 B 프레임을 생략 인출하고, 80% 이상인 경우에는 홀수 파라메터를 가진 P 프레임을 인출한다. 마지막으로 네트워크의 부하가 90% 이상일 경우에는 모든 B 프레임과 P 프레임을 생략하고 I 프레임만 인출한다.

표 3. 프레임 생략 인출 방법(N=15, M=3)

네트워크 부하(%)	생략 프레임 선택(주기별 선택)
60이하	I <sub>1</sub> B <sub>2</sub> B <sub>3</sub> P <sub>4</sub> B <sub>5</sub> B <sub>6</sub> P <sub>7</sub> B <sub>8</sub> B <sub>9</sub> P <sub>10</sub> B <sub>11</sub> B <sub>12</sub> P <sub>13</sub> B <sub>14</sub> B <sub>15</sub>
60이상	I <sub>1</sub> B <sub>2</sub> - P <sub>4</sub> B <sub>5</sub> - P <sub>7</sub> B <sub>8</sub> - P <sub>10</sub> B <sub>11</sub> - P <sub>13</sub> B <sub>14</sub> -
70이상	I <sub>1</sub> - - P <sub>4</sub> - - P <sub>7</sub> - - P <sub>10</sub> - - P <sub>13</sub> - -
80이상	I <sub>1</sub> - - P <sub>4</sub> - - - - - P <sub>10</sub> - - - - -
90이상	I <sub>1</sub> - - - - - - - - - - - - -

디스크 제어 레코드를 검사하여 서버 스케줄러가 디스크에서 프레임을 서버 버퍼에 인출한 후의 상태는 그림 4, 5, 6, 7과 같다.

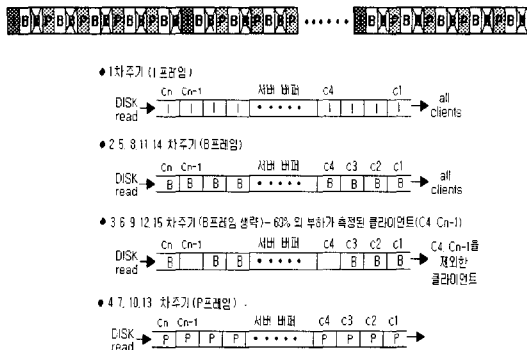


그림 4. 네트워크의 부하가 60% 이상인 경우의 인출

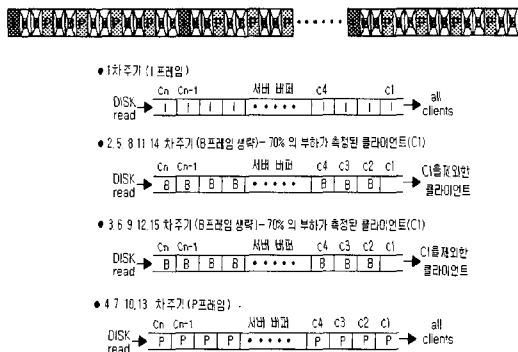


그림 5. 네트워크의 부하가 70% 이상인 경우의 인출

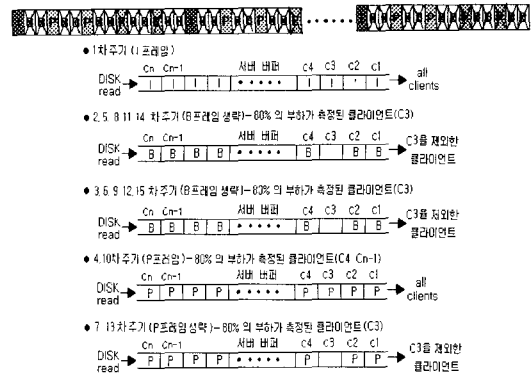


그림 6. 네트워크의 부하가 80% 이상인 경우의 인출

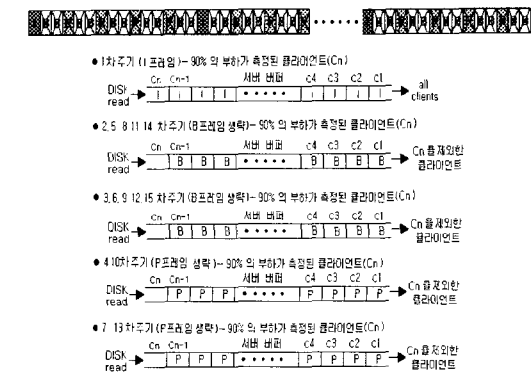


그림 7. 네트워크의 부하가 90% 이상인 경우의 인출

그림 4, 5, 6, 7에서의 디스크 인출과정은 그림 8과 같으며, 15주기(한 GOP)에 동안의 디스크 인출과정이다. 그림 우측 부분의 네트워크 상태 표시는 i번째 주기에서 네트워크의 상태가 변화되어 인출될 프레임이 변환된 것을 보여준다.

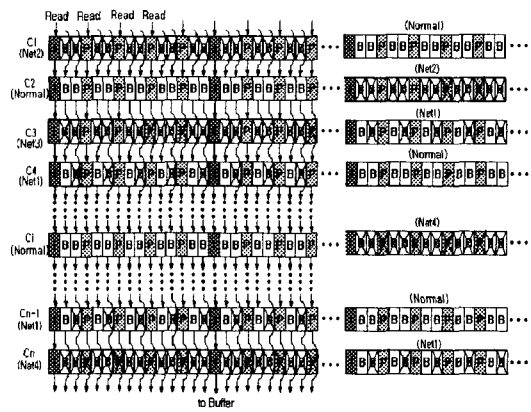


그림 8. 각 디스크에서의 프레임 생략 인출

그림에서 화살표의 진행과정은 라운드로빈 방식으로 디스크를 액세스하는 과정을 보여주며, 매 주기마다 각 디스크에서 생략되는 프레임은 읽기를 하지 않아 버퍼로 인출이 되지 않는다.

최종적으로 각 클라이언트의 네트워크 부하에 따라 디스크로부터 프레임을 버퍼에 인출한 서버 버퍼의 상태는 그림 9와 같다.

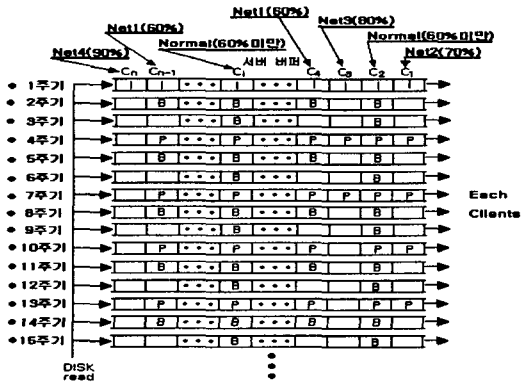


그림 9. 서버 버퍼 상태

● 알고리즘

```

main {
While { // 한 주기 동안의 서비스
Read Disk Control Record(DCR)
case DCR = Net0 call Net0-stat()
case DCR = Net1 call Net1-stat()
case DCR = Net2 call Net2-stat()
case DCR = Net3 call Net3-stat()
case DCR = Net4 call Net4-stat()
} // endwhile
transfer server's buffer to each client
} // endmain

Net0-stat(){ // 네트워크 상태가 정상인 경우
read Disk..frame[i]
Buffer[i] = Disk..frame[i]
}

Net1-stat(){ // B frame number 중 3의 배수가
아닌 프레임만 인출
read Disk..frame[i].info
if Disk.frame[i].info = Disk.frame[B].info then
    
```

```

if mod(Disk.frame[B].number / 3) <> 0 then
Buffer[i] = Disk.frame[i];
else Buffer[i] = Disk.frame[i];
}

Net2-stat(){ // 모든 B frame 생략 인출
read Disk..frame[i].info
if Disk.frame[i].info <> Disk.frame[B].info then
Buffer[i] = Disk..frame[i];
}

Net3-stat(){ // 모든 B frame과 홀수 P frame 생
략 인출
read Disk..frame[i].info
if Disk.frame[i].info <> Disk.frame[B].info
then if mod(Disk.frame[P].number / 2) <> 0
then Buffer[i] = Disk.frame[i];
}

Net4-stat(){ // 모든 B, P frame 생략 인출
read Disk..frame[i].info
if Disk.frame[i].info = Disk.frame[B].info then
Buffer[i] = Disk..frame[i];
}
    
```

4. 시뮬레이션

제안 알고리즘에 대한 시뮬레이션은 PC-Scheme으로 작성된 DEVS(Discrete Event Simulation)를 기반으로 수행하였으며, DEVS는 Petri-Net이나 Finite State Machine에서 모델링할 수 없는 시간 개념을 유효 적절하게 도입하여 시뮬레이션 하였으며 [11], 시뮬레이션 환경은 그림 10과 같다.

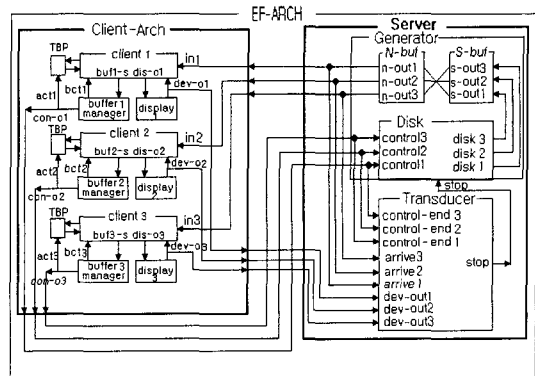


그림 10. 시뮬레이션 환경

그림 10에서 서버의 N-buf, S-buf는 네트워크 버퍼와 서버 버퍼이고, 프레임 생성기(Generator)는 1 주기에 해당하는 하나의 프레임을 디스크 전송속도에 따라 각 디스크에서 프레임을 인출하여 S-buf에 보내고, 서비스 마감시간이 지나면 N-buf에 보낸다. 회선 분배기(DeMux)는 네트워크의 부하에 따른 트래픽으로 인한 수신율 차를 보정한다. 그리고 매체 재생기로 송출되는 데이터율을 적응시키기 위해 버퍼를 두고, 버퍼 감시기에 의해 버퍼의 상태를 모니터링 한다.

프레임 생성기(Generator)는 전송 감시기(Transducer)에서 발한 시작 신호를 수신하여 각 디스크에서 프레임을 인출하여 서버 버퍼에 보내고, 네트워크 버퍼를 통해 3개의 클라이언트에 전송하고, 전송했다는 신호를 전송 감시기에 보낸다.

전송 감시기(Transducer)는 프레임 생성기에 구동 시작과 끝 신호를 보내고, 프레임 생성기의 전송 신호와 클라이언트에서 보낸 스트림의 수신 신호와 버퍼 제어기에서 발한 신호를 수신하여 이들의 상태를 파악하여 다음 주기에 전송할 프레임의 비율을 프레임 생성기에 알린다.

4.1 시뮬레이션 파라미터

시뮬레이션에 필요한 파라미터는 아래와 같다. 표 4의 네트워크의 상태에 따른 전송 평균 비트율은 [12]에서 참조한 것이다.

표 4. 네트워크의 상태에 따른 전송 평균 비트율

비트율	패턴	평균 비트 (KB/s)	제안 알고리즘 (KB/s)
50%	IB <sub>1</sub> B <sub>2</sub> P <sub>1</sub> B <sub>3</sub> B <sub>4</sub> P <sub>2</sub> B <sub>5</sub> B <sub>6</sub> P <sub>3</sub> B <sub>7</sub> B <sub>8</sub> P <sub>4</sub> B <sub>9</sub> B <sub>10</sub>	155	100
60%	IB <sub>1</sub> P <sub>1</sub> B <sub>3</sub> P <sub>2</sub> B <sub>5</sub> P <sub>3</sub> B <sub>7</sub> P <sub>4</sub> B <sub>9</sub>	138	89
70%	IP <sub>1</sub> P <sub>2</sub> P <sub>3</sub> P <sub>4</sub>	122	70
80%	IP <sub>1</sub> P <sub>2</sub>	90	48
90%	I	51	24

클라이언트의 버퍼 크기, 오버런 구역, 스타베이션 구역, 동기 제어 구간은 아래의 계산에 의해 구해진다.

- 클라이언트 수 = 3 (회선 수 : 3)
- 디스크 수 = 3  
(단, 한 디스크에 하나의 비디오 타이틀이 저장되어 있음)
- 서버 버퍼의 크기  
= I 프레임 평균 크기 × 3개의 디스크 × 3 회선  
= 12635 Bytes × 9 = 113715 Bytes(≈111KBytes)  
(단, 서버 버퍼의 크기는 클라이언트 수(회선 수)에 따라 크기가 달라진다)
- 전송 프레임 크기  
= 서버버퍼의 크기 ÷ 회선 수  
= 111KBytes ÷ 3  
≈ 37KBytes  
(클라이언트 버퍼의 크기는 스타베이션을 방지하기 위해 동기 제어 구간을 설정하여 그 크기를 정해야 하므로[13], 1초 동안(2 개의 GOP) 30 프레임 분량의 버퍼 메모리를 설정하였다)
- 본 논문의 클라이언트 버퍼 크기(한 클라이언트)  
= 전송 프레임 크기 × 30 프레임  
= 37KBytes × 30 = 1110KBytes

표 5는 [14]에서 제공되는 MPEG 스트림 데이터의 통계적 특성을 보여주며, 부호화 형식은 하나의 GOP인 IBBPBBPBBPBBPBB(N=15, M=3)를 따른다.

표 5. MPEG 스트림에서 각 픽처의 크기(N=15, M=3)

형태	크기(bytes)	셀(제안 알고리즘)
seccion header	20	
group header	22	
I frame	12429	258.94
P frame	5138	107.05
B frame	2198	45.79
B frame	2234	46.54
P frame	4954	103.21
.....	.....	

그리고 네트워크에서의 지연 정도에 대한 시뮬레이션 파라미터는 ATM을 기준으로 하여 1~4 ms로 사용한다. 이 데이터는[15]에 있다.

최종적으로 본 논문에서의 시뮬레이션 파라미터는 표 6과 같다.

표 6. 시뮬레이션 파라미터

파라미터 속성	단 위
클라이언트 버퍼의 크기	1 Mbytes
프레임 크기	I : 12635 Bytes P : 5991 Bytes B : 1770 Bytes
전송시간	0.033(1/30) 초 0.011 초(하나의 클라이언트)
네트워크의 지연 시간	1~4ms
네트워크의 부하에 따른 생략 전송 프레임 비트율	표 4 참조

4.2 시뮬레이션 결과

네트워크 부하에 대한 클라이언트의 상태 변화는 그림 11, 12, 13, 14와 같으며, MPEG 데이터는 표 4의 프레임별 평균치를 가지고 900초동안 수행한 결과이다.

그림 11은 네트워크 부하가 60% 이상인 경우의 결과를 보여 주며 생략 프레임의 크기가 작은 관계로 그래프의 오실레이션 진폭의 정도가 거의 없는 형태로 진행이 되며 버퍼의 수위가 점진적으로 증가함을 보여 준다.

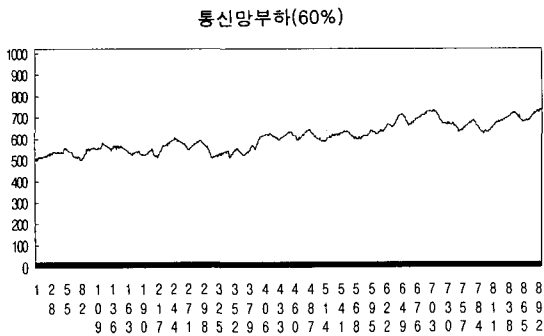


그림 11. 네트워크 부하 60% 이상인 경우

그림 12는 네트워크 부하가 70% 이상인 경우의 결과를 보여 주며 생략 프레임의 크기가 조금 작은 관계로 그래프의 오실레이션 진폭의 떨림 정도가 조금 있는 상태로 진행이 되지만 버퍼의 수위가 안정적으로 유지됨을 보인다.

그림 13은 네트워크 부하가 80% 이상인 경우의 결과를 보여 주며 생략 프레임의 크기가 조금 많은 관계로 그래프의 오실레이션 진폭의 떨림 정도가 심한 상태로 진행이 되어 버퍼의 수위가 점진적으로 낮아

져 수 분내에 버퍼 스타베이션 임계구역에 도달하리라 예상된다.

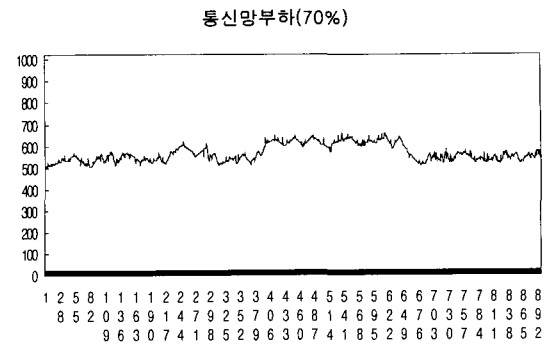


그림 12. 네트워크 부하 70% 이상인 경우

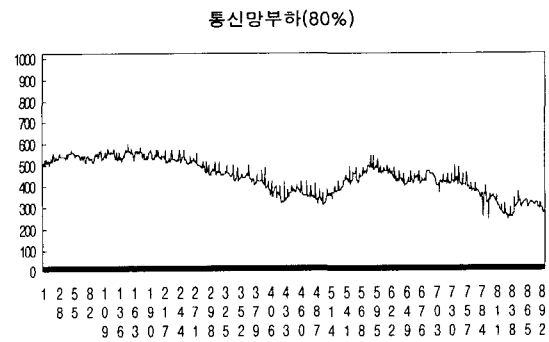


그림 13. 네트워크 부하 80% 이상인 경우

그림 14는 네트워크 부하가 90% 이상인 경우의 결과를 보여 주며 생략 프레임의 크기가 많은 관계로 그래프의 오실레이션 진폭의 떨림 정도가 상당히 심한 상태로 진행이 되어 버퍼의 수위가 상당히 낮아져 수분 내에 버퍼의 스타베이션 임계구역에 도달하리라 예측된다.

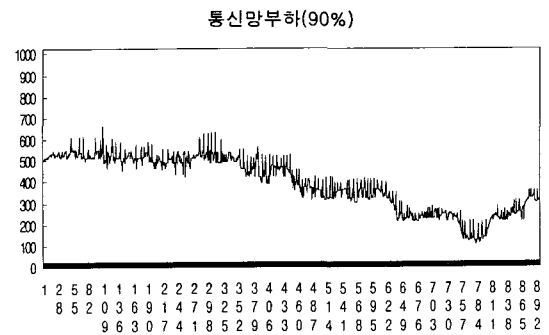


그림 14. 네트워크 부하 90% 이상인 경우



시뮬레이션 결과를 종합해 보면 네트워크의 부하 정도가 60%미만에서는 클라이언트 버퍼의 상태가 점진적으로 증가함을 보인다. 이러한 버퍼의 증가는 네트워크의 경부하 또는 고품질의 MPEG 데이터를 계속적으로 전송한 것으로 추정된다. 네트워크의 부하 정도가 60%이상인 경우는 MPEG 데이터의 생략 정도를 많이 행하여 클라이언트 버퍼의 상태가 점진적으로 감소함을 보인다. 이 경우 네트워크의 부하를 줄일 수 있고, 생략되어 전송되는 프레임의 크기가 너무 많아 화질의 저하를 가져오지만, 재생 단절을 피할 수 있는 장점은 있으나, 네트워크의 과부하가 지속적인 경우에는 클라이언트의 버퍼 상태가 스타베이션 임계구역을 벗어나 재생 단절이 우려되므로 제어 요구를 발하여 서버의 제어를 받는 것이 필요하다.

## 5. 결 론

VOD 서비스는 다수의 사용자 요구를 만족시켜야 하고, 대량의 비디오 데이터를 압축저장하고 있다가 서버의 제어에 의해 실시간으로 전송이 이루어져야 하고, 가입자 측에서는 수신한 비디오 데이터를 압축 복원하여 재생이 이루어져야 한다. 또한 장시간의 비디오 프레임을 전송하는데 있어서 네트워크의 대역폭 및 부하정도를 계속적으로 모니터링하여 적절히 대처해야 한다. 따라서 VOD 서비스는 이러한 요구사항에 의해 여러 가지 제반 메카니즘의 설계가 요구되고, 그 지원 서비스의 설계도 전통적인 텍스트와 수치 데이터를 지원하는 서비스의 설계와는 달라야 한다.

본 논문에서는 이러한 요구사항을 바탕으로 해서 VOD 시스템을 구현하기 위해 VHS 비디오 타이틀을 MPEG으로 압축하여 다수의 디스크에 저장해 두고 네트워크의 부하 상태를 파악하여 디스크에 저장된 프레임을 생략 인출하여 클라이언트에 전송할 프레임을 조정하는 알고리즘을 제시하였다.

시뮬레이션 결과 장시간의 재생을 요구하는 VOD 서비스에서 클라이언트 버퍼의 상태는 VHS 비디오 타이틀의 화질, 압축된 MPEG 데이터의 프레임 크기, 그리고 네트워크의 부하 정도에 따라 민감하게 변화함을 알 수 있었다. 따라서 지속적으로 네트워크의 상태를 감시하여 네트워크의 부하에 적응적으로 대처하여 프레임을 전송한다면 네트워크의 부하를

줄일 수 있으며, 클라이언트의 수신 버퍼를 안정화시킬 수 있음을 알 수 있었다.

앞으로의 연구방향은 MPEG 프레임의 효율적인 저장 방법과 셀 생략 전송 알고리즘과 VOD 서비스의 구현에 필요한 제반 메카니즘의 개발 등에 대한 계속적인 연구가 요구된다.

## 참 고 문 헌

- [1] S.S. Lam, S. Chow, and D.K.Y. Yau, "An Algorithm for Lossless Smoothing of MPEG Video", *In to Appear in the Proceedings of ACM SIGCOMM'92*, London, 1992.
- [2] H. J. Chen, "A Disk Scheduling and MPEG Data Layout Policy Interactive Video Access from a Single Disk Storage Device", Ph. D. Thesis, Boston University, Boston, 1995.
- [3] 정제창, "그림으로 보는 최신 MPEG", 교보문고, 서울, 1995.
- [4] L. Delgrossi, C. Halstrick, D. hehmann, R. G. Herrtwich, O. Krone, J. Sandvoss, and C. Vogt, "Media Scaling for Audiovisual Communication with the Heidelberg Transport System", *Proceedings ACM Multimedia*, 1993.
- [5] J. Sandvoss, J. Winkler, and H. Witting, "Network Layer Scaling : Congestion Control in Multimedia Communication with Heterogeneous Networks" IBM European Networking Center, Heidelberg, 1994.
- [6] H. M. Vin. Alok. Goyal, Anshuman Goyal, and Pawan Goyal, "An Observation-Based Admission Control Algorithm for Multimedia Servers", *In Proceedings of the IEEE International Conference on Multimedia Computing and Systems (ICMCS '94)*, Boston, 1994.
- [7] H. M. Vin, P. Goyal, and A. Goyal' "A Statistical Admission Control Algorithm for Multimedia Server", *In Proceedings of the ACM Multimedia'94*, SanFrancisco, 1994.
- [8] A. Eleftheriadis, D. Anastassiou, "Constrained and general dynamic rate shaping of compressed digital video", *In Proceedings of the*

*IEEE International Conference on Multimedia Image Processing.*, vol. 3, pp. 396-399, Oct., 1995.

- [9] R. J. Safranek, C. R. Kalmanek Jr., and R. Garg, "Methods for matching compressed video to ATM networks", *In Proceedings of the IEEE International Conference on Multimedia Image Processing*, vol.1, pp. 13-16, Oct., 1995.
- [10] W. Zeng, B. Liu, "Rate Shaping by Block Dropping for Transmission of MPEG Pre-coded Video over Channels of Dynamic Bandwidth", *Multimedia 96 Processing, The Fourth ACM International Multimedia Conference*, Boston. Ma. pp. 129-140. 1996.
- [11] D. H. Batly, J. C. Jansen, and D. W. Woods, "PC SCHEME User's & Language Reference Manual", The MIT Press, 1990.
- [12] E. T. Chang, "Storage and Retrieval of Compressed Video", Ph. D. Thesis, California University, Berkeley, 1996.
- [13] 장영, "특집: 비디오 서버의 개념", 전자과학지, 통권 430호, 제 37권, pp. 234-257, 3월, 95년
- [14] "ftp://mm-ftp.cs.berkeley.edu/pub/mpeg/movies/"
- [15] R. Rooholamini, V. Cherkassky, "ATM-Based Multimedia Servers", *IEEE Multimedia*, pp. 39-52, 4월, 1995.



정 홍 섭

1997년 경남대학교 대학원 컴퓨터 공학과(석사)  
2002년 경남대학교 대학원 컴퓨터 공학과(박사)

관심분야 : 멀티미디어, 암호화 설계, 영상처리, VOD, 정보보호



박 규 석

1980년 중앙대학교 대학원 전자계산학과(석사)  
1988년 중앙대학교 대학원 전자계산학과(박사)  
1982년~현재 경남대학교 정보통신공학부 교수  
1992년~1996년 경남대학교 전산정보원 원장

1995년~1996년 한국 정보과학회 이사, 영남지부장  
1999년~2002년 경남대학교 정보통신연구소 소장  
2002년~현재 경남대학교 산업대학원 원장  
2002년~현재 한국 멀티미디어 학회 회장  
관심분야 : 분산처리 시스템, 정보통신 소프트웨어, 멀티미디어 시스템

교신저자

정 홍 섭 631-701 경남 마산시 월영동 449 경남대학교 공과대학 컴퓨터공학과