

---

# 윈도우 기반 동적 대역폭 평활화 방식을 이용한 자원 할당 및 전송 제어 기법

김형진\* · 고성현\*\* · 나인호\*\*

## Resource Allocation and Transmission Control Scheme using Window-Based Dynamic Bandwidth Smoothing Method

Hyoung-Jin Kim\* · Sung-Hyun Go\*\* · In-Ho Ra\*\*

---

이 논문은 2005년도 군산대학교 전자정보공학부 BK21 산학협력팀의 연구비를 지원받았음

---

### 요 약

초고속 통신망을 통해 다양한 서비스 품질을 요구하는 멀티미디어 데이터의 서로 다른 실시간 전송 제약을 보장 할 수 있는 스트림 전송 기법에 대한 연구가 활발히 진행되고 있다. 각 멀티미디어 응용의 데이터 전송 요구에 따라 차별적으로 자원을 예약하는 기법을 설계하고 할당되지 않고 남아 있는 자원의 활용률을 높일 수 있는 대역폭 할당 기법과 유연한 실시간 전송을 제공할 수 있는 파이프 라이닝 기법을 제시하였다. 또한 멀티미디어 데이터의 고유 특성으로 인한 동기성을 보장하기 위해서 수신 버퍼를 기반으로 한 피드백 전송 제어 기법을 적용하여 실시간 전송이 가능하도록 하였다. 그리고 네트워크 폭주로 인한 전송 경로상의 병목현상이 발생하였을 때 최소한의 서비스 품질을 보장하면서 에러 허용률 범위안에서 데이터 전송량에 대해 네트워크 자원 요구량을 저하시킬 수 있는 전송 제어 기법을 제안하였다. 마지막으로 송·수신자 간에 지연에 민감한 비디오 스트림이 연속적으로 전송 될 수 있도록 각 비디오 스트림이 요구하는 최대 대역폭을 평활화 할 수 있는 동적 대역폭 평활화 기법을 제안하였다.

### ABSTRACT

Recently, many of researches on stream transmission for satisfying each of different real-time transmission condition of the multimedia data that demands various service quality through high-speed networks have been studied actively. In this paper, we design a scheme that discriminately reserves the network resources for the transmission of each multimedia application and propose a bandwidth allocation scheme for improving the utilization ratio of free resources. And we also propose a pipelining scheme for providing flexible real-time transmission. The proposed schemes can be used to support a real-time transmission by applying feedback transmission control method based on receiving buffer for guaranteeing the synchronization conditions requested by the multimedia data. Moreover, we propose a transmission control scheme that can take the amount of network resources down to the minimum amount within the range of permissible error-range under the guarantee with no quality degradation simultaneously when the bottleneck is caused by the network congestion. Finally, we propose a dynamic bandwidth smoothing scheme that can smooth the maximum bandwidth to the demand of each video steam for giving continuous transmission to the delay sensitive video steam between senders and receivers.

### 키워드

Resource Allocation, Transmission Control Scheme

---

\* 익산대학 정보통신과

\*\* 군산대학교 전자정보공학부

## I. 서 론

최근 컴퓨터, 고속 네트워크, 데이터 압축 기술 분야의 급속한 발전으로 인하여 멀티미디어 서비스의 실현이 가속화되고 있다. 따라서 다양한 미디어 데이터와 미디어 형식을 지원하는 멀티미디어 서비스를 제공하기 위해서는 오디오와 비디오 같은 연속 미디어 스트림의 연속성 및 동기성을 보장할 수 있어야 한다. 연속미디어 스트림의 충분한 서비스 품질을 보장하기 위해서는 동기화, 흐름제어, 버퍼관리 기술들이 개발되어야 한다[1][2].

본 논문에서는 상이한 미디어 스트림의 전송 특성으로 가변적인 전송속도(VBR)를 요구하는 데이터에 대한 자원 예약 기법과 네트워크 상황에 따라서 전송속도를 제어할 수 있는 기법에 대하여 제안하였다.

본 논문의 구성은 다음과 같다. II장에서는 시스템 모델, 서비스 모델, 제안 기법을 기술한다. III장에서는 제안된 기법을 성능평가하고, 마지막으로 IV장에서 결론을 맺는다.

## II. 본 론

### 2.1 시스템 모델

본 논문에서는 VoD나 실시간 비디오 화상회의와 같이 실시간 데이터를 전송 및 수신하는 경우 발생된 세션에 자원 할당과 효율적으로 관리하는 기법에 관한 연구이다. 또한 자원 부족이나 네트워크 폭주와 같은 사건이 발생하였을 때 수신자에게 지속적인 서비스를 제공할 수 있도록 하기 위해서 전송 속도를 제어하도록 하였다. 그림1은 요구된 데이터에 대한 전송 경로의 지연 시간을 나타내고 있다[3].

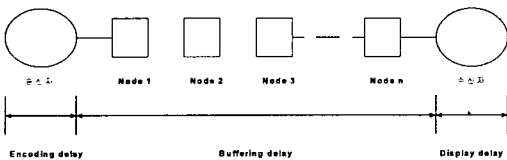


그림 1. 전송 경로 상의 지연 시간 모델  
Fig. 1 Delay time model on a transmission channel

전송 경로 상에서 발생하는 지연시간을 세 가지 성분으로 구분하면 다음과 같다. 요구된 데이터를 준비하는 지연시간(encoding delay)과 네트워크를 통하여 소비되는 버퍼링 지연시간(buffering delay), 클라이언트에서 디스플레이 장치로 보내는 디스플레이 지연시간(display delay)으로 구분된다. 여기서, 버퍼링 지연시간은 데이터가 전송 버퍼, 네트워크, 수신 버퍼를 점유하는 시간이다.

### 2.2 서비스 모델

멀티미디어 압축 기술과 다양한 응용의 등장으로 다양한 미디어 형태들이 출현되고 있다. 일반적으로 멀티미디어 서비스는 일정한 라운드 단위로 전송프레임을 스케줄링 한다. 각 라운드는 보통 16ms의 프레임으로 알려져 있다. 다시 말해서 서버는 매 16ms마다 전송이나 각 미디어 요구를 스케줄 한다. 그림2는 각 프레임의 내부구조를 설명하고 있다. 전송 스케줄링에 따라 요구된 데이터를 제공하기 위해서 전송될 패킷을 일정한 시간 구간의 윈도우 단위로 우선순위를 고려하여 전송 프레임에 구성한다. 각 프레임에 대한 윈도우의 순서는 우선순위에 따라서 전송될 데이터를 할당한다. 즉, 데이터 형태에 따라 실시간 데이터를 위한 큐, 빠른 응답성 데이터를 위한 큐, 대기큐, 일반 텍스트 데이터를 위한 큐 순으로 우선순위를 설정하여 주기적으로 라운드 로빈 방식으로 서비스한다.

전송 프레임의 구조는 다음과 같다.

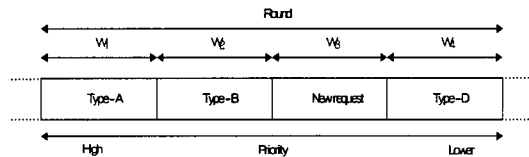


그림 2. 전송 프레임 구조  
Fig. 2 Transmission frame structure

한 라운드 동안에 연결된 세션에서 전송되는 패킷의 전송율은 라운드 시간동안에 전송되는 패킷의 크기로 계산된다. 각 사용자에 대한 전송율은 세션 설정단계에서 패킷의 크기를 결정하여 할당한다. 네트워크 폭주 또는 수신측 버퍼의 상태에 따라서 전송을 조절에 대한 정보가 사용자로부터 피드백 되었다면 생성되

는 패킷의 크기를 조정하여 전송율을 조정한다[4].

### 2.3 제안된 기법

멀티미디어 서비스는 실시간성을 보장할 수 있는 자원을 예약하고 네트워크 폭주와 같은 상황이 발생할 경우 최소한의 실시간성을 보장할 수 있도록 전송 제어를 통해 각 비디오 스트림이 요구하는 최대 대역폭을 평활화 할 수 있는 기법을 제안한다.

#### 2.3.1 자원예약 기법

송신측과 수신측의 세션이 결정되기 전에 실시간성을 보장할 수 있는 네트워크 자원을 할당해야 한다. 그러한 실시간성을 보장할 수 있는 전송 속도를 *target rate*라고 한다. 이 전송 속도는 네트워크의 가용 대역폭과 응용의 자원 예약 능력에 따라서 결정된다. *target rate*는 CBR 예약 방법을 사용하여 자원을 예약한다. 본 논문에서는 세션이 유지되는 동안 연속 미디어 스트림의 출력 연속성을 보장하도록 할 것이다.

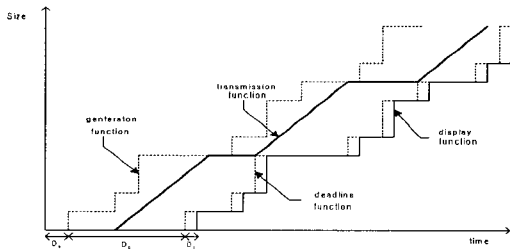


그림 3. 실시간 비디오에 대한 정보량 평활화 과정  
Fig. 3 Smoothing process of an amount of the information for a real-time video

그림 3은 시간에 따라 연속 미디어 데이터가 생성되고 소비되는 과정을 나타낸 것이다. 여기에서, 비디오 데이터와 같은 프레임 단위로 생성되는 생성 함수와 클라이언트에서 출력되는 디스플레이 함수를 나타내고 비디오 출력의 연속성을 보장할 수 있는 전송속도( $R$ )로 전송되는 전송함수를 보이고 있다. 전송함수는 가변 비트율의 전송속도를 갖는 비디오 데이터와 네트워크 상태로 인하여 발생하는 지터 때문에 다음 두 구간으로 나눌 수 있다. 그 첫 번째는 수신측에서 요구한 전송 속도로 전달하는 전송 구간과 송신측의 버퍼에 충분한 데이터가 채워지지 않아서 전송을 할

수 없는 유향 구간으로 이루어져 있다.

전송속도( $R$ )는 서버 측에서 패킷의 크기를 결정하는데 이용된다. 요구된 패킷을 전송하기 위해서 네트워크의 가용 대역폭을 할당한다. 만약 가용 대역폭이 부족하면 전송 속도( $R$ )를 보장할 수 없게 된다. 이와 같은 경우에는 다음과 같은 정책을 수행한다.

① 실시간 응용이 허용하는 범위 내에서 버퍼링 지연시간을 최대화시킨다.

② 양자화 제어

실시간 응용이 허용하는 범위 내에서 허용된 버퍼링 시간으로 결정된 후에도 연속적인 출력을 보장할 수 없다면 전송 프레임 크기를  $R(\text{target rate})$ 과 버퍼링 지연시간( $D_b$ ), 인코딩 시간( $D_e$ ), 버퍼레벨( $B$ )을 근거로 최대 프레임의 크기를 계산하여 엔코더에 의해서 프레임 생성하도록 한다.

$$f_{\max} = \min(R \times (D_b + D_e) - B, R \times D_b) \quad (1)$$

세션이 설정되기 전에 계산된 전송속도와 엔코더에 의해서 생성되는 전송 프레임 크기는 네트워크 자원의 상태에 따라서 정확하지 않을 수도 있다. 이러한 경우에는 실시간 응용의 연속성을 보장할 수 없다. 따라서 자원예약의 부정확성으로 인하여 발생하는 연속 미디어의 불연속성을 보정하기 위해서는 세션 중에 자원할당 및 해제를 수행하도록 한다. 세션 중에 전송 속도를 변경하기 위해서는 다음 두 가지 단계로 수행한다. 첫째, 현재 버퍼 수준과 프레임 크기의 예측값을 기반으로 새로운 전송율을 탐색하는 단계를 수행한다. 둘째, 전 단계에서 검출된 새로운 전송율로 전송을 시작하며 전송 중에 가용 프레임 정보로부터 새로운 전송율의 변경이 요구될 때 변경 여부를 결정한다. 다음 전송 프레임부터 변경된 전송율로 데이터를 전달한다. 이와 같이 두 단계로 나누어서 속도 변경을 조정하면 전송을 변경요구의 횟수를 줄이면서 실시간 전송을 유지할 수 있다. 또한 피드백 정보로 인한 전송 오버헤드를 줄일 수 있다.

#### 2.3.2 파이프라이닝

파이프라이닝은 요구된 데이터가 클라이언트 측에 도착한 이후 일정한 시간 간격으로 수신되는 과정을

말한다. 이러한 전송 경로 상의 데이터 흐름을 파이프 라인이라 한다. 연속 미디어 데이터의 실시간성을 보장할 수 있는 파이프라이닝은 두 가지 방법이 있다. 첫째, 사용자가 요구한 데이터 전송율을 전송 경로 상에 있는 모든 링크의 전송 대역폭으로 예약하는 방법과 전송 경로 상에서 가용한 모든 전송 대역폭을 할당하는 방법으로 분류할 수 있다. 전자는 데이터를 요구한 사용자 측의 어떠한 디스크 버퍼도 요구하지 않는 장점을 갖는다. 후자는 네트워크 자원에 대한 활용율을 높일 수 있다는 장점을 갖는 반면 사용자 측의 디스크 버퍼와 같은 부가적인 자원을 필요로 한다.

본 논문에서는 전송 링크 상에서 사용되지 않은 대역폭을 활용하여 짧은 시간 내에 데이터를 전송할 수 있도록 하였다. 데이터의 자원 점유 시간을 줄여 다른 사용자들이 자원을 이용할 수 있도록 하여 자원 활용율을 높이고 충분한 자원 할당으로 비디오와 같은 데이터의 연속성을 초기에 확실히 보장한다. 이러한 기법은 다음과 같은 단계로 수행된다.

- ① 모든 링크 상에 사용하지 않고 남아 있는 자원이 있는지 검사한다.
- ② 전송 경로 상에 병목 노드를 검출한다.
- ③ 각 병목 노드에서 요구된 버퍼의 크기를 계산한다.
- ④ 각 링크와 병목 노드에서 활용될 대역폭을 계산한다. 결과적으로 병목 노드는 버퍼링 노드로써 동작한다.
- ⑤ 객체의 일정한 부분이 어느 버퍼링 노드에서 저장되어야 하는지를 지정하는 검색 계획을 생성한다.
- ⑥ 요구된 자원을 사용자에게 할당하고 전송을 초기화한다.

여기서, 전송 경로 중에서 발생할 수도 있는 병목 노드를 검출하기 위해 다음과 같은 기법을 적용한다. 어떤 노드에서 입력되는 데이터와 다음 노드에 전송하는 데이터가 같아야 추가적인 자원을 요구하지 않는다. 만약 그 노드에서 입력된 데이터보다 출력되는 데이터가 적다면 출력되지 않고 남아 있는 데이터를 위한 버퍼가 필요하다. 이러한 추가적인 버퍼를 요구하는 노드로 인하여 병목현상이 발생한다. 그 병목 노드

는 전송되지 않고 남아 있는 데이터를 저장하기 위해 버퍼를 할당해야 한다.

임의노드  $i$ 에서 다음 노드로 전송할 수 있는 가용 대역폭을  $f_{out}$ 이라 한다면 사용자에게 할당할 수 있는 전송대역폭은 다음 식 (2)와 같다.

$$f_{\max}(N_i) = \text{Min}_{j=i+1}^s f_{out}(N_j) \quad (2)$$

다음과 같이 조건을 갖는 노드  $N_i$ 는 병목노드로 설정한다.

$$f_{\max}(N_i) > f_{out}(N_i) \quad (3)$$

이와 같은 병목 노드의 버퍼 요구량은 식 (4)와 같이 계산된다.

$$S_{buffer}(N_i) = \frac{f_{achievable}(N_i) - f_{out}(N_i) * S_{object}(k)}{f_{achievable}} \quad (4)$$

단,  $f_{achievable}(N_i) = \text{Min}(f_{\max}(N_i), BW(N_i) + f_{out}(N_i))$

### 2.3.3 수신자 기반의 전송 제어 기법

본 논문에서는 수신자 기반의 버퍼 상태 정보를 기반한 전송 속도를 변경하는 제어 방법으로 송신자와 수신자간의 동기성을 유지 할 수 있도록 하였다. 이를 위해서 수신 버퍼는 정상 레벨, 상위 임계 레벨, 상위 제어 레벨, 하위 임계 레벨, 하위 제어 레벨의 5가지의 상태를 갖게 된다. 다시 말해, 5가지 수준의 임계값(threshold)을 갖는다. 수신 버퍼에 도착한 데이터가 5가지 수준의 임계값에 이를 때 전송율 조정 정보를 피드백 한다. 이러한 버퍼 임계값과 전송 속도의 사용은 연속 미디어 스트림의 실시간성을 보장하기 위한 것이다. 버퍼 임계값들 간의 간격은 응용 특성에 따라서 결정한다. 그 간격 폭을 너무 넓게 결정한다면 전송 속도를 변경하는 제어 정보의 전송 요구는 줄어들지만 버퍼의 언더플로우나 오버플로우로 인하여 데이터 손실률이 높아진다. 또한, 그 폭을 너무 좁게 결정하면 수신 버퍼에서 발생하는 데이터 손실률은 낮아지지만 전송 속도를 조정하기 위한 제어 정보의 요구가 빈번하게 발생하여 네트워크의 부하를 가중시키게 된다. 따라서 클라이언트에서 요구한 미디어의 특성에 따라 버퍼 임계값들 간의 간격을 알맞게 결정하여 연속미디어 스트림의 최소한의 품질을 보장하고 네트워크의 폭주

가 발생하지 않는 범위로 결정한다. 따라서 수신자 기반 전송 제어 알고리즘은 다음과 같다.

표 1. 수신자 기반 전송 제어 알고리즘  
Table. 1 Transmission control algorithm based on receiver

```

// flowout(i) : the amount of available bandwidth on
// link connecting i node
// flowin(i) : the amount of available bandwidth from
// sending node to i node
// flowachievable(i) : the maximum amount of
// bandwidth without data loss on i node
// R : a feasible rate for real-time transmission
//  $\alpha$  : the specific value of real-time application

when a receiver request a session
  identify all the bottleneck nodes
  compute the size of buffer required at each bottleneck
  node
  if bottleneck node on transmitting path then
    determines the bandwidth of flowachievable(i)
    buffering node is computed the size of the
    portion as(4)
  generate a retrieval plan
  allocate required resources and start the delivery for a
  session is established
  if R is a feasible rate then
    continue to delivery the rate
  otherwise
    step 1 : relaxing within delay bound
    step 2 : request to change a frame size send
    threshold level and  $\alpha$  to sender
end session
    
```

### 2.3.4 윈도우 기반의 동적 대역폭 평활화 기법

기존의 적응형 종단간 제어 기법은 네트워크의 폭주문제에 대하여 고려를 하지 않을 뿐만 아니라 데이터 압축으로 인하여 미디어 스트림에 나타날 수 있는 트래픽 밀집성에 대한 영향에 대해서도 고려하지 않았다. 압축으로 인한 트래픽 밀집성 때문에 스트림을 전송하는 동안에 가변적인 대역폭을 요구할 수밖에 없다. 가변 스트림으로 인하여 전송 대역폭은 계속해서 변화하기 때문에 자원을 할당을 어렵게 하고 적응형 기법의 설계를 어렵게 한다. 이러한 문제점들을 해결하기 위해서 본 논문에서는 임의의 수의 프레임들을 하나의 윈도우로 묶음으로써 대역폭의 요동성을 완화시키도록 하였다. 일반적으로 한 윈도우즈 내에서 이러한 대역폭의 요동성을 완화하는 것을 평활화라고 한다. 본 논문에서 사용되는 평활화 기법은 한 윈도우 단위로 필요한 대역폭의 평균값으로 데이터를 전송하도록 하였다. 패킷의 전송율을 평활화 하기 위해서 평균 전송율은 다음과 같이 계산한다.

$$\Delta a = \frac{\Delta W}{N_{current}} \quad (5)$$

여기서  $\Delta a$ 는 평활화된 패킷 전송율이고  $\Delta W$ 는 윈도우 크기이고  $N_{current}$ 는 현재 윈도우 간격에서의 패킷 수를 의미한다.

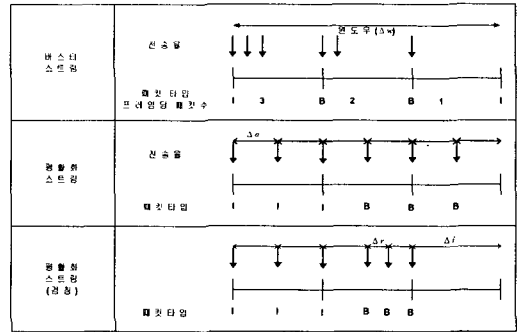


그림 4. 대역폭 평활화 과정  
Fig. 4 Bandwidth smoothing process

평활화된 패킷 전송율로 전송구간(interval)동안에 전송하면 수신측은 전송구간 동안에 최소의 대역폭과 일정한 전송율을 보장할 수 있다. 이러한 방법으로 데이터를 전송할 때 수신측에 의해서 전달된 피드백 메시지가 현재 또는 과거의 대역폭 요구에만 적용되다는 단점이 있다. 미래에 보다 높은 대역폭을 요구하는 수신측에 대해서는 고려하지 않았다. 그런 높은 대역폭은 네트워크가 처리할 수 없을 것이다.

그림 4는 한 윈도우 내에서 여러 미디어 유닛을 그룹화하여 대역폭의 변화폭을 순화시키는 과정을 설명하고 있다. 또한 검침 기법을 사용하여 다음 윈도우의 미디어 유닛의 전송 간격을 예측하도록 하였다.

그림 4와 같이 패킷의 전송을 위한 임계간격( $\Delta$ )을 예측하였다면 패킷 전송을 위한 대역폭 요구량은 평활화 기법과 검침 기법에 의해서 트래픽 밀집성을 감소시켜 최적화할 수 있다. 현재 윈도우에서 예측된 임계간격은 다음 윈도우간에서 패킷의 전송간격으로 제공된다. 임계간격을 위한 최소 대역폭을 계산하기 위해서 일정한 간격으로 균일화한 패킷의 분포를 나타내는 평균화 알고리즘(averaging algorithm)을 적용한다. 임계 간격에 대한 패킷 간 전송 간격은 다음과 같이 계산한다.

$$\Delta r = \frac{\Delta W}{N_{current}} \quad (6)$$

여기서,  $N_{critical}$ 은 임계간격에서의 패킷의 개수를 의미한다. 따라서 다음 윈도우에 대한 최소 대역폭은 다음과 같다.

$$BW_{min} = \frac{S_{packet}}{\Delta r} \quad (7)$$

여기서,  $BW_{min}$ 은 최소 대역폭이고  $S_{packet}$ 은 패킷의 크기를 의미한다.

따라서 본 논문에서는 수신측에서 요구한 전체 데이터에 대한 사전 정보를 이용하여 전송 스케줄을 작성하지 않고 네트워크 상태에 따라 일정한 범위의 전송 프레임에 대한 정보를 가지고 전송스케줄을 작성할 수 있도록 재생 지연 시간, 수신측 버퍼, 송신측 처리 능력, 전송 프레임 추정 기법들을 통합적으로 고려한 평활화 모델을 설계하고 지연시간에 덜 민감한 응용에 적용할 수 있는 윈도우 기반의 평활화 기법이다. 알고리즘은 표 2와 같다.

표 2. 윈도우 기반 평활화 알고리즘  
Table. 2 Window-based smoothing algorithm

```

// s : Transmission speed at server
// W : Dynamic window size
// Bs : Smoothing region size of client buffer
// Θ : Maximum increment limit of smoothing region of client buffer
// δ : Increment of transmission speed at server
s' = s;
W' = W;
REPEAT
    status = read_feedback
    switch(status) {
        case INITIAL:
            W = initial_input_value;
            if(s > s') s = s';
            break;
        case CONGESTION:
            W = W/2;
            s = s + s * δ;
            break;
        case SLOW-START:
            W = W + Δ;
            break;
        case STEADY:
            if(W < W' * (Bs + Θ)/Bs)
                W = W + W*(Θ/Bs);
            break;
    }
UNTIL
    
```

### III. 실험 및 평가

본 논문은 급격히 변화하는 전송폭을 완화하기 위해 본 논문에서 제안한 자원 할당 및 전송 제어 기법을 네트워크 전송 상태와 버퍼 레벨 제어를 위한 방법

으로 윈도우 기반 동적 대역폭 평활화 기법을 이용하여 전송시간의 급격한 변화를 줄이게 하였다.

시뮬레이터는 NS-2를 사용하고[5], 실험결과를 출력하기 위해 Xgraph틀을 사용하여 출력하였다.

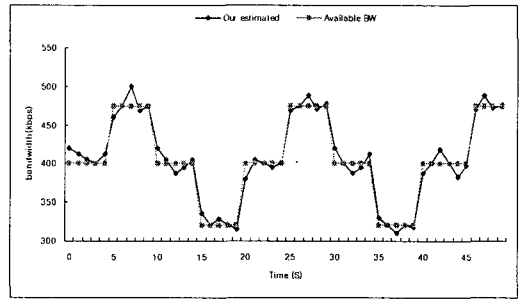


그림 5. 제안된 전송제어 기법을 이용한 가용 대역폭 추적 결과

Fig. 5 The results of the appropriate bandwidth detection using the proposed transmission control scheme

그림 5는 본 논문에서 제안한 기법을 기반으로 네트워크 대역폭 변화를 추적할 수 있는지를 보여주고 있다. 네트워크 대역폭이 320Kbps~480Kbps 사이로 변할 때 네트워크 상태 변화에 네트워크 대역폭 적응의 효율성을 증명하기 위해 제안된 전송 제어 기법을 이용하여 가용 대역폭의 추적 결과를 보인 것이다. 따라서 본 논문에서 제안한 네트워크 상태 변화에 따른 전송 제어 기법은 가용 네트워크 대역폭을 우수하게 추적하고 있음을 보여주고 있다.

그림 6은 비디오 데이터를 33ms마다 전송한 결과를 보여주고 있다. 비교적 불규칙한 전송시간이 소요되는 것을 볼 수 있다. 그림 7에서 오디오 데이터를 전송한 결과로서 비디오 데이터의 전송 시간보다 낮은 것으로 나타났다.

그림 8은 그림 6, 그림 7에서 불규칙하게 전송되는 대역폭을 완화하기 위하여 제안된 윈도우 기반 평활화 기법을 적용한 것이다. 따라서 결과적으로 전송 시간의 변동이 적게 나타남을 볼 수가 있다. 제안된 평활화 기법의 경우, 지연 변동이 불규칙하지 않기 때문에 패킷 손실률 과 중복되는 패킷 발생이 많이 줄어드는 것을 볼 수 있었으며, 무엇보다도 갑작스러운 변화가 없으므로 변화하는 네트워크 전송 속도에 적응하기 쉽다.

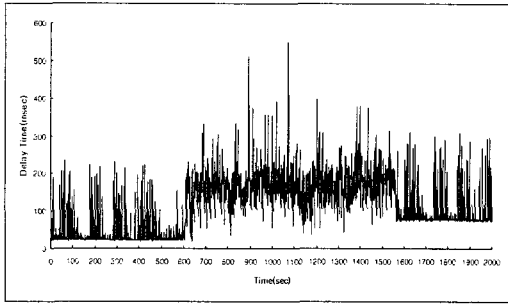


그림 6. 비디오 패킷에 대한 전송 지연  
Fig. 6 The transmission delay for the video packets

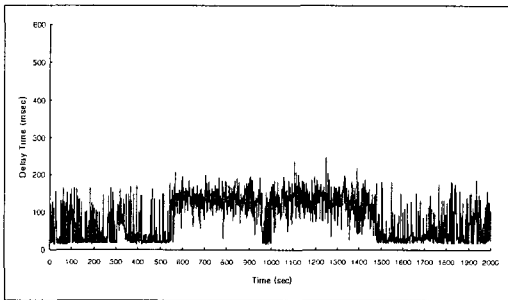


그림 7. 오디오 패킷 지연시간  
Fig. 7 The audio packets delay time

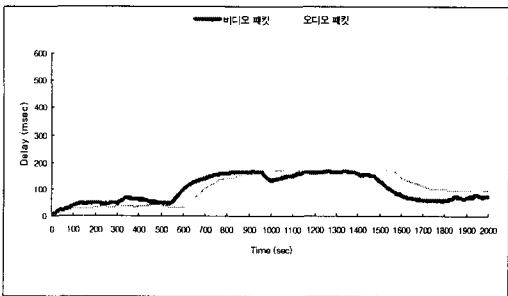


그림 8. 오디오 패킷과 비디오 패킷의 평활화 기법에 의한 전송  
Fig. 8 The transmission using the smoothing scheme of the video and audio packets

#### IV. 결 론

본 논문에서는 멀티미디어 데이터를 전송하기 위한 자원 예약 기법과 자원의 이용율을 높일 수 있는 파이프 라이닝 기법에 대하여 제안하였다. 또한 다양한 응용에 맞게 송수신자간의 동기성을 보장하기 위해서 다중 버퍼 수준을 이용하여 적응적인 전송 제어 기법을 제안하였다. 또한 시뮬레이션을 이용한 성능평가를 통해 제안된 윈도우 기반 대역폭 평활화 기법을 사용하여 지연에 민감한 오디오/비디오 스트림의 최대 대역폭의 요구량을 완화시키고 전체적으로 대역폭 변화율을 평활화시킬 수 있음을 보였다.

#### 참고문헌

- [1] J. Y. Hui, E. Karasan, J. Li, J. Navas, J. Zhang, "Client-Server Synchronization and Buffering for Variable Rate Multimedia Retrievals", Proc. Intl. Workshop Multimedia Synchronization, May, 19, 1995.
- [2] S. Cyrus, H. A. Mohammand, W. Shimen, "A Buffering Policy Distributed Continuous Media Servers", Integrated Media Systems Center and Computer Science Departement Univ. of Southern California, Los Angeles, California 90089.
- [3] R. Guerin, S. Kamat, V. peris, R. Rajan, "Scable QoS Provision Through Buffer Management", Proc. ACM SIGCOMM, Sep. 1998.
- [4] K. W. Lee, S. Ha, J. R. li, V. Bharghavan, "An Application-level Multicast Architecture for Multimedia Communications", In Proceedings of the 8th ACM International Conference on Multimedia, October, 2000
- [5] E. Amir et al. UCB/LBNL/VINT Network Simulator - ns2 (version 2.26). Available at <http://www.isi.edu/nsnam/ns>.

저자 소개



김형진(Hyung-Jin Kim)

1997년 호원대학교 전자계산학과 졸업  
1999년 군산대학교 정보통신공학과 석사  
2004년 군산대학교 정보통신공학과 박사

2004. 9 ~ 2005. 3 군산대학교 전자정보공학부 계약교수  
2005. 4 ~ 현재 익산대학 정보통신과 전임강사  
※관심분야 : 멀티미디어 DBMS, 멀티미디어 시스템



고성현(Sung-Hyun Go)

1998년 군산대학교 전과공학과 졸업  
2000년 군산대학교 전과공학과 석사  
2003. 3 ~ 현재 군산대학교 전자정보공학부 박사 수료  
1997. 11 ~ 현재 이지시스템(주) 대표이사

※관심분야 : 멀티미디어 시스템, 전파통신, 초고속통신망



나인호(In-Ho Ra)

1988년 울산대학교 전자계산학과 졸업  
1991년 중앙대학교 전자계산학 석사  
1995년 중앙대학교 전자계산학 박사  
1995. 9 ~ 현재 군산대학교 전자정보공학부 부교수

※관심분야 : 멀티미디어 통신시스템, 초고속 통신망