

# 분산 멀티미디어 스트림상의 PULL 방식 데이터 전송기법 설계 및 구현

김정훈\*, 이승현\*, 박용희, 이조원, 임영환\*\*

\*송실대학교 대학원 컴퓨터학과

\*\*송실대학교 정보과학대학 컴퓨터학부

E\_mail : -jhkim@hanmail.net

## A Design and Implementation of PULL-based Data Transfer Method on Distributed Multimedia Streams

Jeonghun Kim\*, Seunghyun Lee\*, Yonghee Park\*, Jowon Lee\*, Younghwan Lim\*\*

\*Dept. of Computing Soongsil Univ.

\*\*School of Computing, Soongsil Univ.

### 요 약

기존의 MuX 시스템은 분산환경에서 멀티미디어 데이터를 전송하기 위해서 PUSH 방식의 데이터 전송기법만을 제공한다. PUSH 방식의 데이터 전송기법은 자료 발생매체 중심의 데이터 전송기법으로 멀티미디어 데이터가 발생할 때 바로 자료 사용 매체로 전송하는 기법이다. PUSH 방식 데이터 전송 기법은 네트워크 대역폭이 크고 시스템 자원이 풍부한 경우 간단히 연출 QoS를 만족시킬 수 있다. 그러나 네트워크의 대역폭이 좁은 시스템에서는 PUSH 방식 데이터 전송기법으로 연출 QoS를 만족시키는데 한계가 있다. 이러한 문제를 해결하기 위해 PULL 방식 데이터 전송기법을 제안하였다. PULL 방식 데이터 전송기법은 자료 사용 매체 중심의 데이터 전송 기법으로 자료 사용 매체의 요구시점에 요구한 만큼의 데이터를 발생시켜 그 매체로 전송하는 방법이다.

### 1. 서 론

멀티미디어 데이터를 사용한 연출에 있어 한정된 자원과 네트워크 대역폭의 한계로 인해 일정한 수준의 연출 QoS를 만족시키는데는 어려움이 많다.

이러한 문제를 해결하기 위해 기존엔 멀티미디어 데이터를 압축하여 네트워크를 통해 전송함으로써 적은 네트워크 대역폭 상의 멀티미디어 연출을 가능하도록 하였다. 하지만 멀티미디어 데이터를 압축하더라도 연속된 멀티미디어 스트림을 네트워크를 통해 전송하는 경우 그 처리의 방법에 따라 연출 QoS를 만족할 수 없는 경우가 발생한다.

따라서 본 논문에서는 낮은 네트워크 대역폭 상에서 멀티미디어 스트림의 연출 QoS를 만족시키기 위한 PULL 방식 데이터 전송 기법을 제안하고, 이를 구현하였다.

PULL 방식 데이터 전송기법을 제공하기 위해 스트림 제어기 상에 버퍼관리자와 매체 스케줄러를 두어 네트워크 및 시스템 자원인 빈약한 경우 멀티미디어 스트림의 연출 QoS를 만족시키기 위한 방법을 제공한다.

### 2. PUSH/PULL 방식 데이터 전송기법

#### 2.1 PUSH 방식 데이터 전송기법

PUSH 방식 스트림 제어기법은 자료발생지점의 시스템의 입장에서 발생한 데이터를 빠르게 자료사용지점으로 전송하는 것을 목적으로 한다. 이는 자료발생지점 중심의 스트림 제어기법이라 말할 수 있다.

본 논문에서는 PUSH 방식 스트림 제어기법을 자료발생지점에서 생성된 데이터를 중간에 아무런 처리없이 자료사용지점의 시스템으로 무조건 전송하는 것으로 정의한다.

PUSH 방식 스트림 제어기법은 자료발생지점에서 생성된 데이터를 최대한 빠르게 자료사용지점으로 전송하는 것을 의미하므로, 자료발생지점의 시스템 과부하 및 자료사용지점의 시스템 환경에 대해 고려하지 않는다. 이러한 방식은 자료발생매체 및 자료사용매체의 시스템 환경이 좋은 경우 및 네트워크를 통한 데이터 전송속도가 빠른

경우에 연출 QoS를 쉽게 만족시킬 수 있다.

## 2.2 PULL 방식 데이터 전송기법

PULL 방식 데이터 전송기법은 자료발생매체 중심이다. PUSH 방식의 데이터 전송기법은 실시간으로 발생하는 데이터를 처리하는데 있어 효율적이지만, 자료사용매체 및 자료발생매체의 시스템 환경이 빈약하거나 네트워크를 통한 데이터 전송속도가 느린 경우 멀티미디어 연출에 있어 일정한 QoS를 보장할 수 없다.

자료사용매체에서 멀티미디어 데이터의 원활한 연출을 위해 자료사용매체 중심의 데이터 전송기법을 PULL 방식 데이터 전송기법이라 정의한다.

PULL 방식 데이터 전송기법은 자료사용매체가 필요한 데이터를 자료발생매체에 요구하고 자료발생매체는 자료사용매체가 요구한 양의 데이터를 생성해 자료사용매체에 전달하는 기법을 말한다.

자료사용매체 중심에서 필요한 양의 데이터를 자료발생매체에 요구하고 자료발생매체에서 생성한 데이터를 전달받아 이를 버퍼에 적체하여 연출 시 버퍼에서 데이터를 가져와 사용한다.

자료사용매체로의 데이터 전송량과 사용량을 고려하여 최적 크기의 버퍼를 할당하고 이 버퍼에 최적 크기의 데이터를 선인출하여 일정한 수준의 QoS를 제공할 수 있다. 만일 연출 중에 데이터의 전송량과 사용량이 변화한다면 이에 맞추어 선인출 해야 하는 데이터의 크기를 변화시켜 QoS를 일정하게 유지할 수 있다.

멀티미디어 연출 시 자료사용매체의 시스템 환경, 멀티미디어 매체의 데이터 요구량 및 네트워크를 통한 데이터 전송량을 계산하여 최적의 버퍼를 할당하고, 일정한 양의 데이터를 선인출하여 연출에 사용한다. 목적지 시스템의 환경이 변한 경우 선인출 데이터 크기를 재설정하여 시스템 환경변화에 따른 동적 대응이 가능하다.

특히 자료사용지점의 시스템자원이 빈약하거나, 시스템 환경이 불규칙적으로 변화하는 경우 데이터의 요구량, 버퍼에 선인출해야 할 데이터의 크기를 동적으로 변경하여 일정한 QoS를 유지한다.

## 3. PUSH 방식의 문제점 및 해결방안

PUSH 방식의 데이터 전송기법은 화상회의 및 실시간 데이터의 전송 및 처리에 있어 효율적이다. 하지만 분산 환경에서 멀티미디어 연출 시에 PUSH 방식의 데이터 전송기법 만으로는 만족스러운 연출 QoS를 제공하는데 한계가 있다.

이와 문제를 해결하기 위해서는 스트림 내부에 데이터

의 전송 속도 및 데이터 사용속도를 고려한 최적의 버퍼를 할당하고, 이 버퍼에 스트림의 재생시간 이전에 일정량의 데이터를 선인출(Prefetch)하여 초기지연을 제거할 수 있다. 자료 사용매체의 환경이 변화거나 네트워크를 통한 전송 속도가 변한 경우 이에 맞게 선인출하는 데이터의 크기를 조정하여 분산환경의 멀티미디어 연출에서 일정한 수준의 QoS를 제공할 수 있다.

멀티미디어 데이터의 원활한 연출을 위해서는 버퍼관리기법 뿐만 아니라 자료 발생매체 및 자료 사용매체의 일관된 관리를 위한 스케줄링(Scheduling) 방법이 필요하다.

본 논문에서는 매체간의 스케줄링 기능 및 PUSH/PULL 방식 데이터 전송기법을 제공하는 같은 분산 멀티미디어 스트림 서버를 구현하였다. 스트림 제어기는 분산 컴퓨팅 환경에서 연출 QoS의 만족을 위해 버퍼관리기능 및 중앙 집중적인 매체 스케줄링기능, 연출 데이터의 선인출 기능을 제공한다.

## 4 버퍼 관리 기법 및 매체 스케줄링 기법

### 4.1 버퍼관리 기법

#### 4.1.1 선인출 알고리즘

##### 4.1.1.1 선인출 데이터 크기 ( x ) 결정

$$x = n(a - b).....식 (1)$$

a : 초당 재생되는 데이터량(bytes/sec)

b : 통신망을 통해 전송되는 데이터량(bytes/sec)

n : 스트림의 연출시간(sec)

여기서 x 만큼 선인출 되어 있을 때 시간별 데이터량 변화는 [표 4-1]과 같다.

시간 (sec)	1	2	3	...	N
데이터량 (byte)	x-a+	x-2a+	x-3a+	...	x-na
	b	2b	3b	...	+nb

[표 4-1] 보유 데이터 량 변화표

n초 후에는 실행이 종료되고 보유한 데이터량이 0이 된다. 따라서 조건  $x - na + nb = 0$  을 만족하는 선인출할 데이터 량 x의 값은  $n(a - b)$  이 된다.

##### 4.1.1.2 선인출 시작 시점( t )결정

$$t = start\_time - x/b.....식 (2)$$

start\_time : stream의 실행 시작 시간(sec)

$x$  : 선인출할 데이터량 (byte)

$b$  : 초당 통신망을 통해 전송되는 데이터량 (bytes/sec)

위의 식에서 초당 데이터의 사용량이 64Kbps 인 H.261 비디오 스트림의 56Kbps의 모뎀을 통해 전송하여 1시간 동안 멀티미디어 연출을 하는 경우 선인출 해야할 데이터의 크기 및 선인출 시작시점을 계산해보면 다음의 [표4-2]와 같다.

- 네트워크를 통한 데이터 전송량 : 5600bps (b)
- 자료 사용 매체에서 데이터의 사용량 : 6400 bps (a)
- 멀티미디어 스트림의 연출 시간 : 3600 sec (n)

선일출해야 할 데이터 량(x)

$$= 3600 \text{ sec} * (6400\text{bps} - 5600 \text{bps})$$

$$= 2880000 \text{ bit} = 360 \text{ KByte}$$

선일출 시점(t) =

$$- 2880000 \text{ bit} / 5600 \text{ bps} \approx - 514 \text{ sec}$$

∴ 연출 시점의 약 514초 전에 연출데이터의 선인출 이 시작되어야 한다.

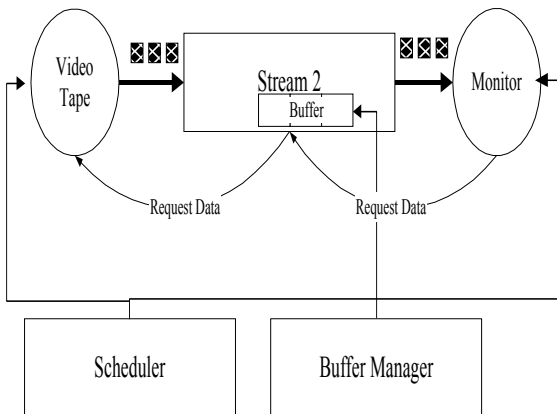
[표 4-2] 데이터의 선인출 시점 결정 예제

#### 4.1.2 버퍼관리기법

버퍼는 스트림 상에 존재하며, 버퍼관리자가 이 버퍼를 관리한다.

버퍼관리자는 4.1.1에서 설명한 데이터 선인출 알고리즘 및 선인출 시기 결정 알고리즘에 따라 스트림 상에 일정한 크기의 버퍼를 할당하고, 할당된 버퍼를 관리한다.

#### 4.2 매체 스케줄링 기법



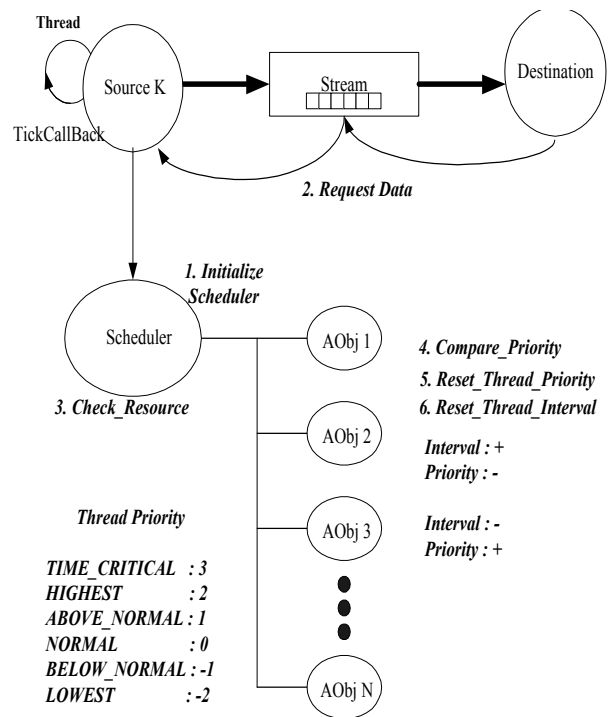
[그림 4-1] 스케줄러의 구성

[그림 4-1]에서 자료 발생매체 및 자료 사용매체를 스케줄러가 통합적으로 관리한다.

자료 사용매체에서 일정한 수준의 연출 QoS를 만족하기 위해서는 일정한 수준의 데이터 전송속도를 유지해야 한다. 자료 사용매체는 스트림상의 버퍼에서 일정한 속도로 데이터를 인출한다. 자료사용매체는 버퍼관리자를 통해 버퍼의 상태변화를 인식하여, 필요한 양의 데이터를 스트림을 통해 자료 발생매체로 요청한다.

스케줄러는 이러한 요청을 받아 자료 발생지점의 데이터 발생량을 조정하여 이를 스트림상의 버퍼에 전송하도록 한다.

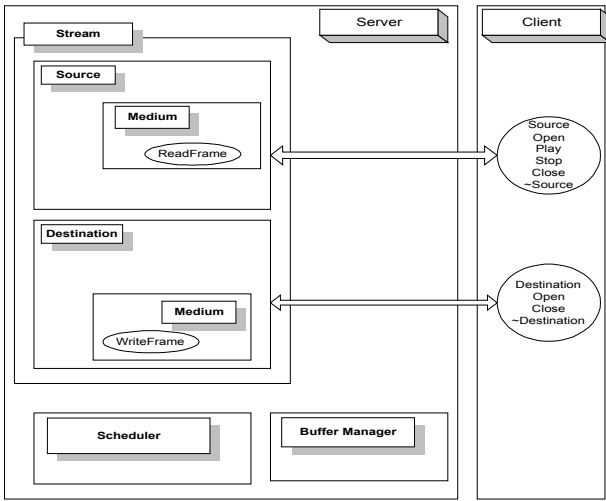
#### 4.2.1 매체 스케줄링 방법



[그림 4-2] 매체 스케줄링 방법

[그림 4-2]은 스트림 제어기 상의 매체 스케줄링 기법을 도식으로 표현한 것이다. 버퍼 관리자에 의해 자료 사용 매체에서의 연출에 이상이 있음을 판단하면, 버퍼관리자는 최적 버퍼 크기와 선인출 데이터의 크기를 계산하여 이를 자료 사용매체 및 스케줄러에 알린다. 이 때 스케줄러는 요구받은 데이터량을 자료 사용 매체로 전달하기 위해 자료 발생 매체의 자료발생주기를 조절한다. 자료 발생 주기가 변경되면, 그 주기마다 자료 발생 매체를 스케줄링 하여 자료 발생 매체에서 멀티미디어 데이터를 발생시켜 자료 사용 매체로 전송할 수 있도록 한다.

#### 5. 구현



[그림 5-1] 스트림 제어기 클래스 계층도

[그림 5-1]은 PULL 방식의 데이터 전송기법을 제공하기 위한 스트림 제어기의 클래스 계층도이다.

### 5.1.1 스트림 객체

- 자료 발생매체에서 생성된 데이터를 자료 사용매체로 전달한다.
- PULL 방식 데이터 전송기법을 제공하기 위해 스트림 상에 버퍼를 가지고 있으며, 자료 발생매체는 멀티미디어 데이터를 생성하여 스트림 상의 버퍼에 전송하고, 자료 사용매체는 스트림상의 버퍼에서 데이터를 가져와 이를 사용해 멀티미디어 연출을 수행한다.

### 5.1.2 버퍼관리자

- 자료 사용매체의 데이터 사용량 및 네트워크를 통한 데이터의 도달량을 계산하여, 스트림 상에 최적 크기의 버퍼를 할당한다.
- 할당된 버퍼에 일정량의 데이터를 선일출하여 연출시 일정한 QoS를 유지할 수 있도록 한다.
- 스트림의 연출 중에 자료 사용매체의 자료 사용량 및 데이터의 도달량을 주기적으로 모니터링하여 버퍼에 선일출 해야하는 데이터량을 계산하고, 이 값이 일정한 QoS 변수를 만족하지 않는 경우 이를 자료 사용매체 및 스케줄러에 알린다.

### 5.1.3 스케줄러

- 자료 사용매체 및 자료 발생매체를 통합적으로 관리하는 객체이다.
- 버퍼관리자를 통해 스트림 상에서 자료 발생 매체의 자료발생주기가 변경 요구가 들어온 경우, 자료 발생 매체의 자료발생주기를 변경하고 변경된 주기에 따라

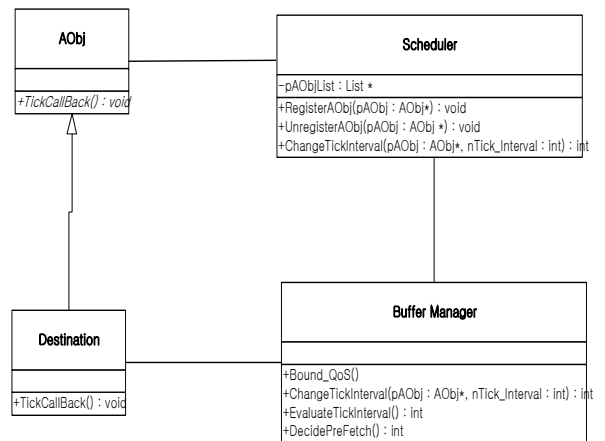
서 자료 발생매체에서 멀티미디어 데이터를 발생시킨다.

- 버퍼의 데이터량이 아주 적은 경우에는 자료사용주기를 늦출 수 있다. 버퍼에 데이터가 적정한 수준에 도달하면 자료사용주기를 정상적인 수준으로 변경한다.

### 5.1.4 자료 사용 매체 및 자료 발생 매체

- 스트림 상에서 멀티미디어 데이터를 발생시키고, 이를 사용하는 매체를 말한다.
- 자료 사용 매체 및 자료 발생 매체는 스케줄러에 의해 통합적으로 관리된다.
- 매체는 그 각각이 고유의 자료 사용(발생) 주기를 가지고 있다.

## 5.2 스트림 제어기의 클래스 Diagram



[그림 5-2] 스트림 제어기의 클래스 Diagram

[그림 5-2]는 PULL 방식 스트림 제어기법을 제공하기 위한 개략적인 클래스 다이어그램이다.

### 5.2.1. AObj(Active Object) Class

MuX 시스템에서 스케줄의 단위가 되는 클래스로 기존엔 자료 생성 매체만이 AObj를 상속받아 구현되어 있지만 본 스트림 제어기에서는 자료 사용 매체도 AObj를 상속받아 구현되었다.

- TickCallBack : 가상함수, 매체가 Tick을 받아 활성화되었을 때 수행된다.

### 5.2.2. Scheduler Class

매체들을 통합적으로 관리하고, 매체들의 고유의 Tick 주기에 따른 스케줄링을 담당한다.

#### 5.2.2.1. Attribute

- pAObjList : 스케줄러에서 관리하는 AObj의 List를 말한다.

### 5.2.2.2. Method

- RegisterAObj : pAObjList에 하나의 AObj를 등록한다.
- UnregisterAObj : pAObjList에 하나의 AObj를 삭제한다.
- ChangeTickInterval : 등록되어 있는 AObj의 TickInterval을 변경한다.

### 5.2.3. Buffer Manager Class

초기에 자료 사용 매체의 프리패치 크기를 결정하고, 지속적으로 시스템을 감시하여 시스템 환경에 따른 Tick 주기의 변화 및 프리패치 크기의 변경한다.

- Bound\_QoS : 시스템의 환경을 지속적으로 감시하고 그 결과가 일정한 수준의 QoS를 만족하는 지 검사한다. 검사결과에 따라 자료 사용 매체의 Tick 주기 및 프리패치 크기를 변경한다.
- ChangeTickInterval : 자료 사용 매체의 Tick 주기를 변경한다.
- EvaluateTickInterval : Bound\_QoS에서 Tick 주기를 변경해야 하는 경우에 최적의 Tick 주기를 계산하기 위해 사용된다.
- DecidePreFetch : 자료 사용 매체에서 프리패치 해야할 크기를 결정한다.

### 5.2.4. Destiantion Class

자료 사용 매체 클래스이다. 기존의 MuX 시스템에서 자료 사용 매체는 Obj(Object) 클래스를 상속받아 구현되어, 스케줄링 될 수 없는 수동적인 매체였으나, AObj를 상속받아 구현함으로써 자료 사용 매체가 동적 스케줄링의 단위가 될 수 있다.

- TickCallBack : 한번의 Tick 주기마다 일정량의 데이터를 자료 생성 매체에서 가지고와 버퍼에 넣는다.

### 5.3. 스트림 제어기 사용자 API

PULL 방식 스트림 제어 기법을 사용자가 사용하기 위해서는 다음과 같은 API를 사용해야 한다.

함 수	기 능
[ 표 5-1 ] 사용자 API	
6. 결 론	기존의 Stream 생성자와 같은 역할을 수행하지만, mode 변수를 설정함으로써 Push/ Pull 방식을 정할 수 있다.
Stream	사용자가 기존의 생성자를 사용하면 push 방식으로 설

분산 환경에서 기존 PUSH 방식을 사용한 연속된 멀티미디어 데이터 스트림의 연출은 네트워크 대역폭의 영향을 많이 받았다. 네트워크 자원을 무한정 사용할 수 있다면 어떠한 연속 매체라도 원만한 연출이 가능하지만, 네트워크 대역폭의 한계가 있어 낮은 대역폭을 사용하는 경우 멀티미디어 연출 QoS를 만족시킬 수 없다.

이에 본 논문에서는 PULL 방식의 데이터 전송기법을 제안하고, 연구하였다.

PULL 방식 데이터 전송 기법은 자료 사용매체 중심의 데이터 전송 기법으로 자료 사용 매체의 요구 시 요구한 양의 데이터를 자료 발생매체에서 생성해 자료 발생매체로 전송하는 기법이다.

PULL 방식 데이터 전송기법을 사용함으로써 네트워크 대역폭이 적은 경우에도 원만한 멀티미디어 연출이 가능하다. PULL 방식 데이터 전송기법을 제공하기 위하여 스트림 제어기 상에 버퍼 관리자를 두어 최적 크기의 버퍼를 할당하고, 멀티미디어 스트림의 연출 전에 일정한 양의 데이터를 선인출 한다. 연출 중에는 시스템의 상태에 따라 선인출 해야하는 데이터의 양을 계산하여 시스템 환경 변화에 대응한다.

PULL 방식 데이터 전송 기법을 사용함으로써 네트워크 대역폭이 적은 시스템에서도 멀티미디어 스트림의 연출 QoS를 만족시킬 수 있다.

### 참고문헌

- [1] 임영환, "ComBiStation : 분산 멀티미디어 컴퓨팅 환경을 위한 컴퓨터 플랫폼," 정보과학회 논문지, 제 2권, 제 1호, 1996, pp.160-181.
- [2] Baker R. A. Dowing, K.Finn, E.Rennison, D.H.Kim, and Y.H. Lim, "Multimedia Processing Model for a Distributed Multimedia I/O System," Proceedings of 3rd International Workshop on Network and Operating Systems for Digital Audio/Video, 1993, pp.233-239.
- [3] Rennison. E, R.Bker, D.H.Kim, and Y.H.Lim, "MuX : An X Co-Existant Time-Based Multimedia I/O Server," The X Resoure, Issue 1, 1992, pp.213-233.
- [4] C.W.Mercer and H.Tokuda, "The ARTS Real-Time Object Model," IEEE Real-Time System Symposium, 1990, pp.2-10.
- [5] K.Jeffay, D.L.Stone and D.E.Poirier, "YARTOS:Kernel Support for Efficient, Predictable Real-Time Systems," Proceedings of

- IFAC, Workshop on Real-Time Programming, 1991.
- [6] D.P.Anderson, "Meta-scheduling for distributed continuous media," ACM Transaction on Computing Systems, Vol.11, No.3, 1993, pp.226-252.
- [7] M. Salmony and D. Shepherd, "Extending OSI to Support Synchronization Required by Multimedia Applications," computer Communications,13,1990,pp.399-406.
- [8] 함재욱 외 4명, "멀티미디어 프리젠테이션 동기화를 위한 선행 스케줄링 기법," 정보과학회 논문지, 제21권, 제12호, 1994, pp.2187-2197.
- [9] Mon-Song Chen, Dilip D. Kandlur, Philip S. Yu, "Optimization of the Grouped Sweeping Scheduling (GSS) with Heterogeneous Multimedia Streams," ACM Multimedia 93, 1993, pp.235-242.
- [10] T.Tsang, R.Lai, "Time-Estelle : An Extended Estelle Capable of Expressing Multimedia QoS Parameters," 1997, pp.311-318.
- [11] Doo-Hyun Kim, Sang-Hwan Kung, and Chee-Hang Park, "Empirical Delay Comparisons of Push and Pull Implementation Models for Local Audio and Video Streams," IEEE PROMS-MmNet97, Santiago, Chile, 1997, pp.24-26.
- [12] 임영환 등, 멀티미디어 컴퓨터공동연구개발, 연구보고서 1994. 7.