

# 분산 환경에서의 멀티미디어 프리젠테이션 모델

## (A Multimedia Presentation Model in Distributed Environments)

최 속 영\*  
(Sook-Young Choi)

**요 약** 분산 멀티미디어 문서 시스템은 네트워크로 연결된 여러 서버에 있는 미디어들은 검색하여 제시된 시간 관계에 따라 미디어들을 프리젠테이션한다. 효과적인 프리젠테이션을 위해서는 동기화가 지원되어야 하며, 특히, 분산 환경에서의 프리젠테이션은 네트워크 대역폭과 지연시간 등에 의해 영향을 받기 때문에, 그러한 요소들이 고려되어 동기화가 지원되어야 한다. 본 연구에서는 미디어들이 각 서버로부터 전송될 때 네트워크의 상태와 자원을 검사하여 그에 따른 변화를 서버에 피드백시켜, 서버로부터 전송되는 데이터의 양을 조절함으로써 동기화를 지원하는 분산 멀티미디어 프리젠테이션 모델을 제안한다.

**Abstract** In the distributed multimedia document system, media objects distributed over a computer network are retrieved from their sources and presented to users according to specified temporal relations. For effective presentation, synchronization have to be supported. Furthermore, presentation in distributed environments is influenced by network bandwidth and delay, so they have to be considered for synchronization. This paper propose a distributed multimedia presentation model that supports synchronization effectively. In this model, when media objects are transferred from servers to a client, network situation and resources are monitored and thus a feedback message for change of them is sent to server. Then, the server adjusts the data sending rate to control synchronization.

### 1. 서 론

초고속 네트워크의 등장과 더불어 미디어 압축 기술의 발달, 저장장치 용량 등의 증가 등으로 여러 가지 다양한 미디어를 통합 처리하는 멀티미디어 시스템의 발전은 광고, 교육, 안내, 국방, 의학, 정보처리 관련 등 여러 분야에 적용되고 있다. 특히, 최근에는 네트워크 기술의 발전으로 분산 멀티미디어 응용 시스템들이 연구 개발되고 있다.

멀티미디어 프리젠테이션(presentation)은 실시간(real time)에 생성되거나 저장 장치에 저장된 미디어 데이터를 처리하여 사람이 인식할 수 있는 형태로 보여주는 것을 의미한다. 멀티미디어 프리젠테이션에서는 미디어간의 시

간 및 공간 정보를 효율적으로 표현하고 처리할 수 있는 기법이 요구된다[6]. 시간 정보는 여러 미디어간의 시간 관계 또는 시간 동기화(synchronization) 정보를 의미하며, 이것을 모델링하고 제어하기 위한 여러 연구들이 있어왔다[2,3,4,5,7].

본 연구는 선행 연구로서 동기화 정보를 표현하기 위해 프리젠테이션 동기화 트리 PST (Presentation Synchronization Tree)를 제안하고, 이를 이용하여 효과적으로 프리젠테이션하는 수행 모델을 설계하였다[14].

그런데, 이러한 멀티미디어 프리젠테이션을 구성하는 객체들은 협동 작업을 수행하는 시스템들에 분산되어 저장될 수 있다. 예를 들어, 분산 멀티미디어 문서 시스템과 분산 비디오 서버 등과 같은 응용들에서는 네트워크를 통해 연결된 여러 서버로부터 객체들을 검색하여 제시된 시간 관계에 따라 프리젠테이션 되어야 한다[8,9]. 그런데 여러 서버로부터 객체를 검색하는 것은 객체들의 프리젠테이션

\*우석대학교 컴퓨터교육과 조교수

레이션 시작 시간, 수행시간(duration), 네트워크 대역폭(bandwidth)에 의해서 영향을 받는다[8,9,13]. 특히 대역폭과 버퍼(buffer) 크기와 같은 네트워크 자원의 제약성 때문에 데이터를 서버로부터 클라이언트에 전송하는데 동기화 문제가 발생됨으로써 미디어들의 프리젠테이션 시간과 순서에 불일치가 일어난다. 또한 생성지에서 연속적이고 일정한 시간 간격으로 생성된 미디어들이 네트워크를 통해 전송될 때 일정하지 못한 네트워크 전송 지연(delay)과 전송 에러로 인해 목적지에서 연속적인 재생을 받지 못하는 경우가 발생된다[12,13].

본 연구에서는 선행 연구로서 개발된 PST에 기초한 프리젠테이션 모델을 확장하여, 클라이언트-서버 분산 환경에서 효과적으로 프리젠테이션을 수행할 수 있도록 동기화 기법을 제안하고, 이를 지원하는 프리젠테이션 시스템을 제안한다. 이를 위해 본 연구에서는, 네트워크 상태와 자원을 검사하여, 그에 따른 변화를 서버에 피드백(feedback)시켜 서버로부터 전송되는 데이터의 양을 조절함으로써 동기화를 제어하고 있다.

본 연구의 구성은 다음과 같다. 2장에서는 본 연구에서 기초로 하고 있는 프리젠테이션 모델로서 PST와 그를 이용한 동기화 기법을 간단하게 설명한다. 3장에서는 분산 환경에서 프리젠테이션을 위한 고려 사항으로 본 연구에서 제안하고 있는 네트워크의 동적 검사 방법에 대해 기술하고, 이에 기초한 분산 환경에서의 프리젠테이션 모델을 4장에서 설명한다. 5장에서, 본 연구에 영향을 준 관련 연구들의 장·단점을 파악하고 본 연구와 비교 분석한다. 마지막으로, 6장에서 결론을 맺는다.

## 2. PST에 기반한 프리젠테이션 동기화 모델

본 장에서는 선행 연구로서 개발된 프리젠테이션 동기화 모델을 간략하게 설명한다. 본 모델은 단일 서버 모델을 가정으로 개발되었으며, PST(Presentation Synchronization Tree)를 기반으로 하여 동기화를 지원하고 있다.

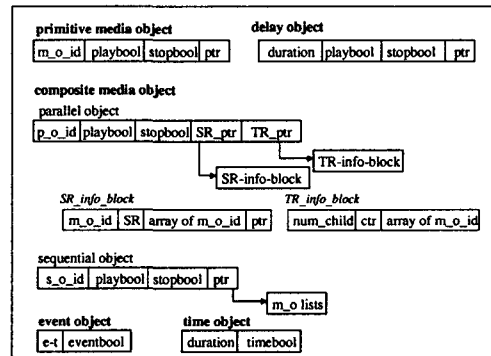
### 2.1 모델 정의

프리젠테이션 동기화 트리 PST는 (N, L, PF)로 정의된다.

- N : PST 노드들의 집합
- $N \subseteq \{PO \cup CO \cup DO \cup EO\}$
- L : 노드 연결 링크의 집합

$$L \subseteq \{PO \times CO\} \cup \{CO \times PO\} \cup \{CO \times CO\} \cup \{PO \times PO\} \cup \{PO \times DO\} \cup \{DO \times PO\} \cup \{CO \times DO\} \cup \{DO \times CO\} \cup \{PO \times TO\} \cup \{DO \times TO\} \cup \{PO \times EO\} \cup \{DO \times EO\}$$

PST를 구성하는 노드 N은 원형 미디어 객체(primitive media object) PO와 복합 미디어 객체(composite media object) CO, 그리고 미디어간의 동기화를 위해 생성되는 지연 객체(delay object) DO로 구성된다. L은 노드들을 연결하는 링크이며, 시간 객체(time object) TO와 사건 객체(event object) EO가 원형 미디어 객체들과 링크를 형성하게 된다. PF는 playing-firer로서 프리젠테이션 수행시 동기화를 맞추면서 노드들을 운행하게 된다.



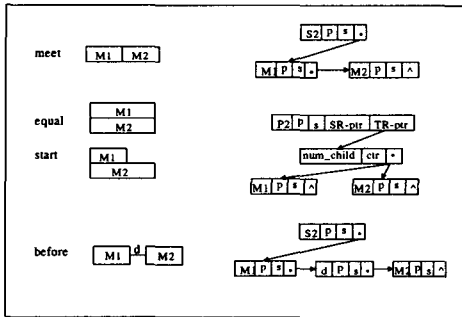
<그림 1> 미디어 객체들의 구성 형태

PST에서 정의하고 있는 객체들의 종류와 그 객체들을 구성하는 필드의 구성은 <그림 1>과 같으며 내용은 다음과 같다. 원형 미디어 객체는 프리젠테이션 될 각 미디어들을 나타내며, 객체의 프리젠테이션 상태는 playbool과 stopbool 필드로서 나타낸다. 지연 객체는 프리젠테이션 동기화를 위한 지연 시간을 주기 위해 필요한 객체이다. 복합 미디어 객체는 원형 미디어 객체간의 시간관계 및 공간관계를 표현하기 위한 객체로서 병렬적 혹은 순차적 관계에 의하여 서로 연관된 모든 원형 미디어 객체들의 집합으로 구성되며, 각각 병렬 객체(parallel object)와 순차 객체(sequential object)로 분류된다. 순차 객체에 연결된 하위 미디어 객체들은 링크를 따라 순차적으로 수행되며, 마지막 객체의 수행이 끝났을 때 순차객체의 수행이 끝나게 된다. 병렬 객체는 병렬 객체를 구성하고 있는 각 미디어 객체들이 동시에 프리젠테이션 되며, 프리젠테이션 지속시간이 가장 긴 미디어 객체의 수행이 끝났을 때 병렬 객체의 수행이 끝나게 된다. 시간 객체는 프리젠테이션 될 객체의 지속시간을 전달받아 외부장치인 시스템 타이머와의 링크를 수행하기 위한 객체이다. 사건객체

는 프리젠테이션 상태를 변경하는 사용자 입력을 처리하기 위한 목적으로 사용되며, 사용자 입력 형태로 일시정지(FZ), 재수행(RS), 전진방향 스킵(F\_SK), 후진방향 스킵(B\_SK), 프리젠테이션 속도 조절(PS) 등을 지원한다.

## 2.2 PST를 이용한 시간관계 명세

임의의 두 미디어 객체간에는 Allen이 제안한 13가지 시간 관계[4]가 존재한다. 이 시간 관계들을 PST로 표현한 것들 중 일부분을 <그림 2>에서 보여주고 있다.



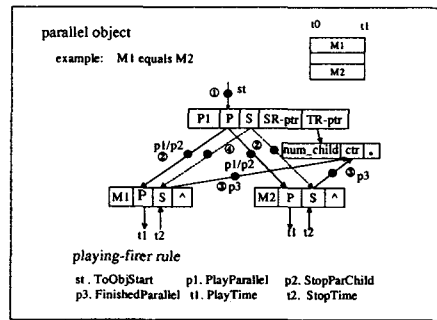
< 그림 2 > Allen의 시간관계에 대한 PST

## 2.3 Playing-firer를 이용한 동기화

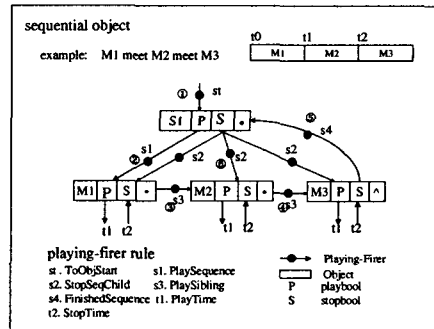
프리젠테이션시 객체들 간의 동기화를 위해 PST는 playing-firer를 운영한다. Playing-firer는 PST의 각 노드들을 방문하면서 각 노드의 playbool을 세트(set)시킴으로써 미디어를 플레이(play)시킨다. 순차 객체에 연결된 미디어 객체들을 플레이 하는 경우에는 한 미디어 객체가 플레이를 마친 후 stopbool이 세트될 때 다음 미디어 객체를 방문하게 된다. 이들 playing-firer는 객체들 간의 동기화를 제어하는데 아주 중요한 역할을 수행하게 된다. <그림 3>은 각 객체에 대한 playing-firer의 수행 규칙(rules)들 중 일부분만을 보여주고 있다.

- st. ToObjStart()
  - set playbool of start object to T
  - to sequential object
- s1. PlaySequence()
  - if playbool(sequential) = T then
    - set playbool(1st media object) to T
  - endif
- s2. StopSeqChild()
  - if playbool(sequential) = F then
    - set stopbool(all media objects) to T
  - endif

- s3. PlaySibling()
  - if stopbool(media object) = T and not end of child then
    - set playbool(next media object) to T
  - endif
- s4. FinishedSequence()
  - if stopbool(last media object) = T then
    - set stopbool(sequential) to T
  - endif



<그림 3> playing-firer의 수행 규칙



<그림 4> PST의 수행 과정 (복합 객체)

<그림 4>는 복합 미디어 객체(순차객체, 병렬객체)의 PST에서 playing-firer를 운행하면서 프리젠테이션 하는 것을 보여주고 있다. 여기서, 아크 번호는 프리젠테이션시 동기화에 관련된 playing-firer의 수행 규칙들로서 <그림 3>에서 기술되었다.

## 3. 분산환경에서 프리젠테이션을 위한 고려사항

멀티미디어 프리젠테이션에서 미디어들을 원하는 시간에 원하는 순서대로 디스플레이(display) 하기 위해서는 효과적인 동기화 기법이 요구된다. 분산환경에서 프리젠테이션을 수행하기 위해서는 여러 서버에 분산되어 저장된 미디어 객체들을 클라이언트로 전송받아 디스플레이해야 한다. 이때, 네트워크를 통해 데이터들이 전송되기 때문에 네트워크의 대역폭과 통신 지연시간은 분산 환경에서 프리젠테이션을 수행하는데 중요한 변인이 된다. 예를 들어, 네트워크에 지연이 발생되면 미디어내(intra-media) 동기화 관점에서 볼 때 미디어들이 자연스럽게 디스플레이되지 않으며, 미디어(inter-media)의 동기화 관점에서는 미디어간의 시간 관계를 위배하는 경우가 발생되어 원하지 않은 프리젠테이션이 될 수 있다. 또한, 네트워크 지연 시간과 관련해 미디어 데이터들이 자연스럽게 디스플레이 되기 위해서는 클라이언트에 있는 버퍼가 효과적으로 유지되어야 하며, 이때 사용 가능한 버퍼가 존재하는지에 대한 검사가 요구된다.

본 연구에서는 동기화를 지원하기 위해서 네트워크 상태를 동적으로 모니터(monitor)하여, 현재의 네트워크 상태에 따라 서버로부터의 전송되는 데이터의 양을 조절함으로써 동기화를 지원하고자 한다.

### 3.1 네트워크 상태 검사

본 연구에서는 네트워크 상태를 모니터하고, 그 현재의 네트워크 상태를 반영하기 위해서 한 서버로부터 패킷을 보내는 시간과 패킷이 클라이언트에 도착하는 시간의 차이 DSA(time difference between sending and arrival packets)를 정의하여 사용한다. 패킷이 전송될 때마다 DSA를 구하여, 현재 네트워크 상태를 반영하는 최적의 DSA를 결정함으로써 네트워크 밀집 현상을 취하고 데이터 손실률을 줄이고자 한다. 즉, 동적으로 최적인 DSA를 모니터하여, 만약 현재의 DSA와 최적의 DSA 사이의 차가 어떤 한계치를 넘으면, 서버에 이 네트워크 상태변화 값을 피드백 시켜 준다. 그러면 서버는 이 정보를 받고 데이터의 품질(quality) 낮추어 전송되는 데이터의 양을 조절함으로써 동기화를 지원하게 된다.

#### 1) 현재 DAD의 계산

CDSA(Current DAD)는 현재 DSA이며 한 클라이언트에 패킷이 도착한 시간에서 서버에서 그 패킷을 전송한 시간을 뺀 값을 의미한다.

$$CDSA(t) = Tarrival - Tsent$$

Tsent : 서버에서 한 패킷이 전송되는 시간

Tarrival : 클라이언트에 한 패킷이 도착하는 시간

#### 2) 최적의 DSA의 계산

한 미디어의 패킷이 클라이언트에 도착하면 현재 DSA를 구한 후, 최적의 DSA를 구하기 위해 패킷이 늦게 또는 빨리 도착했는지와 버퍼가 다 찼는지를 체크하여 데이터 손실률을 계산하게 된다. 패킷이 늦게 도착했는지를 검사하기 위해 현재 구한 DSA값이 허용될 수 있는 최대 네트워크 전송 지연시간보다 큰 경우나, 허용될 수 있는 최소 네트워크 전송 지연 시간보다 작은 경우에는 그 패킷은 버려지고 데이터 손실률로 계산된다. 큰 경우에는 그 패킷이 너무 늦게 도착한 경우이며, 작은 경우에는 너무 빨리 도착한 경우이다. 또한 DSA 값이 허용가능한 값 범위에 포함될 경우, 다음으로 클라이언트의 버퍼에 빈 공간이 있는지 체크하여, 있는 경우에는 버퍼에 집어넣고 없는 경우에는 그 패킷은 버려지고, 데이터 손실률로 계산된다. <그림 5>는 최적의 DSA를 구하는 알고리즘을 나타낸다.

데이터 손실률은 <그림 6>과 같은 알고리즘을 이용하여 동적으로 계산된다. 초기에 데이터 손실률은  $DL(0)=0$ 이다.  $t'$ 는 이전의 LR 계산 시간이고,  $t$ 시간에 한 패킷이 도착하였을 경우, 그 패킷이 늦게 혹은 빨리 도착하거나 버퍼가 다 차 데이터 손실이 발생되었을 경우 데이터 손실률은  $DL(t) = \alpha \times LR(t') + (1 - \alpha)$ 이며, 그렇지 않을 경우에는  $LR(t) = \alpha \times LR(t')$ 이다. 여기서  $\alpha$ 는 주어진 파라미터(parameter) 값이다.

클라이언트에 도착한 각 패킷의 DSA에 대해 데이터 손실률을 구한 후 가장 적은 손실률 값을 갖는 DSA가 BDSA로 된다. 또한 각 패킷의 CDSA 값과 BDSA 값의 차이 값  $\Delta t$ 가 임계치를 넘으면 동기화를 위해  $\Delta t$  값을 서버에 피드백시켜 준다.

#### Procedure Compte\_Bset\_DSA()

```

/* BDSA : current DSA, CDSA : best DSA
DL : Data Loss Rate */
BDSA = CDSA
while new packet arrives do
  compute the current DSA
  CDSA(t) = Tarrival - Tsent
  compute the difference between CDSA and
  BDSA
  Δt = CDSA - BDSA
  if |Δt| > δ then
    send a feedback for Δt to server
  end if

```

```

compute the data loss rate of the packet
CALL Compute_data_loss()
compare DL of CDSA with DL of BDSA
if DL of CDSA < DL of BDSA then
    BDSA = CDSA
end if
end while

```

<그림 5> 최적의 DSA를 구하는 알고리즘

```

Procedure Compute_data_loss()
/* DL : Data loss rate
   Dmax_allow : allowable maximum network delay
   Dmin_allow : allowable minimum network delay
*/
DL(0) = 0
if (CDSA > Dmax_allow) then
    discard the packet
    DL(t) =  $\alpha \times LR(t') + (1 - \alpha)$ 
else if CDSA < Dmin_allow then
    discard the packet
    DL(t) =  $\alpha \times LR(t') + (1 - \alpha)$ 
else if buffer is not available
    discard the packet
    DL(t) =  $\alpha \times LR(t') + (1 - \alpha)$ 
else
    put the packet into buffer
    DL(t) =  $\alpha \times LR(t')$ 
end if
<그림 6> 데이터 손실률을 구하는 알고리즘

```

### 3.2 미디어내(Intra-media) 동기화와 미디어간(Inter-media) 동기화

본 연구에서는 미디어내 동기화와 미디어간 동기화를 제공하고 있다. 미디어내 동기화는 한 미디어 객체를 구성하는 단위들이 시간적 순서를 유지하며 일정하게 디스플레이 되는 것을 의미하며, 미디어간 동기화는 미디어간의 시간 관계와 공간 관계를 맞추는 것을 나타낸다[13].

미디어내 동기화를 위해 타임스탬프(time stamp)를 사용한다. 각 서버가 데이터를 전송시 타임 스탬프를 함께 보낸다. 각 클라이언트에서는 전송된 데이터의 타임스탬프를 검사하여, 그 순서에 따라 각 미디어 단위들을 플레이시킨다. 이러한 미디어내 동기화를 보장하기 위해서는 클라이언트에 버퍼가 유지되며, 먼저 도착한 미디어들은 이 버퍼에 저장된다. 3.1에서 설명한 바와 같이, 허용

가능한 지연 시간의 범위 안에 도착한 패킷들만이 버퍼에 저장되며, 그 범위 밖에 도착한 패킷들은 버려진다.

본 연구에서의 미디어간 동기화는 미디어 사이에 정의된 시간관계와 공간관계에 따라 미디어들을 플레이시킴으로써 이루어진다. 미디어들의 시간관계와 공간관계는 2장에서 기술된 PST에 의해 표현하고 있으며 이 PST를 수행함으로써 시나리오를 구성하는 미디어들이 프리젠테이션 된다. 이러한 미디어간의 동기화는 클라이언트에 있는 PST 관리기에 의해서 제어된다. 또한 네트워크 관리자는 네트워크 상태 변화를 모니터링하여 이를 반영하기 위해 서버에 피드백 정보를 보내, 전송되는 데이터의 양을 조절함으로써 동기화를 제어한다.

## 4. 분산 프리젠테이션 시스템

본 장에서는 분산 환경에서 성공적인 멀티미디어 프리젠테이션을 위한 시스템 구조에 대해 설명하고, 프리젠테이션을 위한 각 단계별 과정을 기술한다.

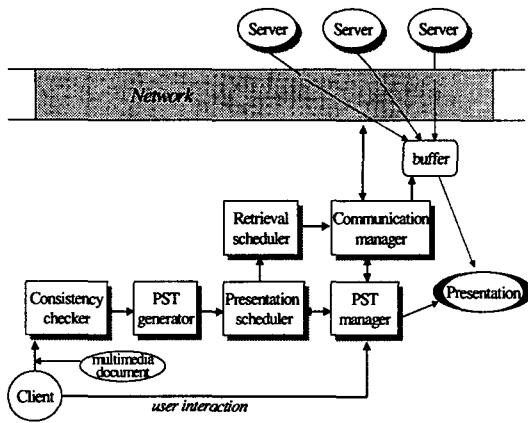
### 4.1 시스템 구조

<그림 7>에서 볼 수 있는 바와 같이, 본 시스템은 사용자에게 의해 입력된 멀티미디어 문서에서 미디어들의 시간관계에 대한 일관성을 체크하는 일관성 유지 검사기(consistency checker), 이 과정에서 검증된 미디어들의 시.공간 관계 정보를 사용하여 프리젠테이션 동기화를 위한 내부 표현 형태인 PST를 생성하는 PST 생성기(PST generator), 실행시간에 미디어들의 프리젠테이션되는 시작 시간을 결정하는 프리젠테이션 스케줄러(presentation scheduler), 각 미디어들이 존재하는 서버로부터 미디어를 검색하는 시간을 결정하는 검색 스케줄러(retrieval scheduler), 현재의 네트워크 상태와 자원을 체크하는 통신 관리자(communication manager), 프리젠테이션 스케줄에 따라 PST를 운행함으로써 프리젠테이션을 수행시키는 PST 관리자(PST manager)로 구성된다.

### 4.2 PST와 프리젠테이션 스케줄의 생성

PST 생성기는 일관성 유지 검사기를 거쳐 생성된 미디어 객체들과 미디어 객체들의 시간 및 공간 정보를 입력받아 프리젠테이션을 위한 PST를 생성한다. 즉, 입력된 시간 및 공간 정보와 미디어 객체들의 프리젠테이션 지속시간을 이용하여 미디어 객체들의 시간 순서에 따라 객체들 간의 링크 및 외부 객체와의 링크를 구성함으로써

PST를 생성하게 된다. 이에 대한 자세한 알고리즘은 [14]에서 참조될 수 있다.



<그림 7> 분산 환경에서의 프리젠테이션 시스템

프리젠테이션 스케줄러는 입력된 시간관계 및 미디어 객체의 프리젠테이션 지속시간을 이용하여 두 미디어 간의 프리젠테이션 시작 시간을 결정한다. 구해진 시작 시간은 미디어 객체 테이블에 저장되어 관리된다. 또한 프리젠테이션 스케줄러는 미디어들의 정적인 시작 시간 값을 기본으로 PST 관리기에 의해 PST가 운행됨에 따라, 시스템에 의해 발생하는 내부 이벤트 큐 및 실시간으로 발생하는 사용자 상호작용에 의한 외부 이벤트 큐를 조사하여 해당되는 객체의 실행 상태 및 객체 수행시간 등을 고려하여 처리한다.

시간 관계가 의미하는 바에 따라 각 미디어의 시작 시간을 결정하는 기본 방법은 <그림 8>와 같다.

Procedure Scheduling\_Proc()

/\* Ps\_time(m) : presentation start time for a media object m

p\_time(m) : presentation duration for a media object m

d\_time(r) : delay time for the temporal relation r \*/

check temporal relation

case before or meet

$Ps\_time(m2) = Ps\_time(m1) + p\_time(m1) + d\_time(r)$

case overlap

$Ps\_time(m2) = Ps\_time(m1) + d\_time(r)$

case during

$Ps\_time(m2) = Ps\_time(m1) - d\_time(r)$

case start or equal

$Ps\_time(m2) = Ps\_time(m1)$

case finish

$Ps\_time(m2) = Ps\_time(m1) - (p\_time(m2) - p\_time(m1))$

<그림 8> 프리젠테이션 시작시간 결정 알고리즘

두 객체 모두의 시작 시간이 결정되어 있지 않은 경우에는 먼저 프리젠테이션 되어야 할 미디어 객체의 시작 점을 0으로 하고, 그 둘 간의 시간 관계에 따라 나머지의 미디어 객체의 시작 시간을 결정한다. 이미 존재하던 미디어 객체와 시간 관계를 갖게 되면 그 미디어 객체의 프리젠테이션 시작시간은 이미 결정되어 있으므로 그것을 이용하여 새로운 미디어 객체의 프리젠테이션 시작 시간을 결정할 수 있다. 두 객체 모두 이미 결정된 시작 시간을 가지고 있는 경우는 문서를 구성하되 서로 독립적으로 존재하는 즉, 어떠한 시간 관계에 의해서도 연결되지 않고 있던 두 컴포넌트(component)를 연결하는 시간 관계가 입력된 것이다. 그러므로 두 번째 미디어 객체의 프리젠테이션 시작 시간을 첫 번째 미디어 객체의 시작 시간을 기준으로 삼아 재조정하고 두 번째 미디어 객체가 속한 컴포넌트의 모든 미디어 객체들의 시작 시간도 재조정하여 준다.

### 4.3 검색 스케줄의 생성

각 미디어들의 검색시간은 네트워크를 통해 전송되는 시간을 고려하여 검색 스케줄러에 의해 결정된다. 이때 한 객체에 대해 검색시간을 계산하는 식은 다음과 같이 표현될 수 있다.

$Rs\_time(o) = Ps\_time(o) - Sinit(o) / Thr(o) - \Delta t$

Rs\_time(o) : 한 미디어 객체의 검색 시작시간

Ps\_time(o) : 한 미디어 객체의 프리젠테이션 시작 시간

Sinit(o) : 프리젠테이션에 요구되는 초기 객체 단위의 크기

$\Delta t$  : 초기 지연시간(클라이언트가 요구 신호를 보내어, 서버로부터 응답을 받는데 걸리는 전파시간)

Thr(o) : 미디어 객체 o를 전송할 때 사용되는 네트워크 처리량

#### 4.4 네트워크 상태 변화와 사용자 상호작용에 따른 동기화 제어

네트워크 관리자는 3.1절에서 기술된 DSA값의 변화에 따라 네트워크 상태 변화를 감시한다. 이를 위해, 전송되는 패킷에 대한 CDS의 값과 BDS의 값을 계산하여 그 차이가 임계값을 넘으면 동기화를 위해 이 값에 대한 정보를 서버에 피드백시킨다. 즉, 현재 네트워크 상태에서 지연이 발생되면 서버에게 데이터의 품질을 낮춰서 전송하도록 하기 위해 CDS 값과 BDS 값의 차이  $\Delta t$ 을 보내게 된다. 서버는 클라이언트로부터 피드백된  $\Delta t$  값을 가지고 데이터 전송률을 조절하기 위해 현재의 생성 주기를 고려하여 데이터의 품질을 낮춘다.

PST 관리자는 사용자 상호작용(user interaction)에 의해 프리젠테이션의 순서가 동적으로 변환하는 경우에 각 미디어 객체의 상태를 관리하고 이를 효과적으로 처리해준다. 본 모델에서 지원하고 있는 사용자 상호작용에는 일시 정지(freeze), 재실행(resume) 전진/후진방향 스킵(forward/backward skip) 등이 있다[14]. 예를 들어 프리젠테이션 중에 전진 방향 스킵이 발생되면, 현재 수행중인 모든 미디어 객체들을 중지시키고 서버에게 현재 전송되는 미디어 패킷들의 전송을 중지하라는 신호를 보낸다. 또한 스킵 시간 후에 플레이되어야 할 미디어들을 체크하여, 그 미디어를 전송하는 서버에게 스킵 시간을 전송하여 스킵 시간 후에 전송될 미디어 패킷을 체크하여 전송하도록 한다.

#### 5. 관련연구

멀티미디어 프리젠테이션의 동기화 문제를 해결하기 위한 여러 연구들이 그 동안 수행되어 왔다. 그러한 연구들은 하나의 단일 시스템에서 멀티미디어 프리젠테이션을 위한 관점으로, 미디어들의 시간 관계를 표현하고 동기화를 제어하고자 하는 연구들[2,3,6,7,16]과 분산 환경으로 확장하여 프리젠테이션에 포함되는 미디어들을 서버로부터 전송을 하여 클라이언트에서 프리젠테이션을 하는 경우에 동기화 문제를 다루는 연구들로 구분될 수 있다 [5,8,9,11,12].

미디어들의 시간 관계를 패트리넷(petri-net) 모델에 기초하여 표현하고 동기화를 제어하고자 하는 연구들이 있어왔다. 특히 [2]에서, Little과 Ghafoor는 시간 패트리넷 모델의 수정 형태로서 OCPN에 기초하여 시간 간격(time interval)을 이용한 모델을 제안하였고, Prabhakar와 Raghavan[3]은 사용자 상호작용을 제어하기 위해 OCPN을 확장하였다.

[9]의 연구에서는 분산 환경에서 프리젠테이션의 동기화를 위해 글로벌 프리젠테이션 그래프(global presentation graph)를 제안하고 있으며, 이 그래프는 시간 파라미터(time parameter)와 하이퍼미디어 링크의 개념을 추가하여 패트리넷 표현을 확장하였다. 또한 서버로부터 데이터 전송시 발생가능한 네트워크 지연을 고려할 수 있도록 하기 위해, 제어 시간(control time) 개념을 사용하고 있다. 그러나, 이 연구는 실제로 서버로부터 미디어 데이터가 네트워크 상으로 전송될 때 발생할 수 있는 네트워크 상태 변화를 적절하게 반영할 수 없다는 단점이 발생한다.

[8]에서는 차이 제약(difference constraints)에 기반한 융통성 있는 시간 관계 모델을 제안하고 있으며, 프리젠테이션 스케줄과 검색 스케줄을 생성하고 있다. 또한 생성된 검색 스케줄을 검증하기 위하여 매 세그먼트마다 요구되는 처리량과 버퍼량을 계산하고 네트워크 제공자가 제공하는 최대 네트워크 처리량과 버퍼량을 초과하는지 검사하여 만족하지 않을 때는 프리젠테이션 스케줄과 검색 스케줄을 조정하도록 한다. 그러나, 이 연구는 데이터 전송시 클라이언트에 늦게 또는 빠르게 도착한 패킷들을 처리하거나, 미디어내 동기화를 위한 기법들을 제공하지 않고 있다.

[10]에서는 트래픽에 기초한 네트워크 지연을 예측하여 동기화를 지원하기 위한 방법을 제안하고 있다. 이를 위해 지연 모델을 정의하고 LAP라는 스케줄러를 지원하고 있다. LAP는 지연을 예측하고 이에 따른 검색 스케줄을 생성해 준다. 그런데 이 연구는 동적인 네트워크 상태를 적절히 반영하지 못하고 있으며, 지연에 따른 동기화를 맞추기 위해 클라이언트에서 객체의 해상도를 낮추는 방법을 사용하기 때문에, 이미 데이터들이 지연이 발생한 상태에서 제어되기 때문에 효과적이지 못하다.

[12]에서는 실시간 버퍼 관리에 초점을 두고 있으며, 네트워크 지연과 클라이언트에 도착한 데이터들의 잘못된 순서를 다루기 위한 버퍼 관리 기법을 제안하고 있다. 네트워크 상태에 따라 서버로부터의 데이터 전송 속도를 조절하고 있다. 그런데, 이 연구는 미디어내 동기화를 제어하기 위해 버퍼를 관리하는 기법이 복잡하다. 또한 네트워크 상에 지연이 발생되어 서버의 데이터의 전송 속도를 높이고자 할 때, 대부분 현재 네트워크 대역폭을 최대한 사용하는 것을 가정하기 때문에, 적합하지 않을 수 있다.

본 연구는 위와 같은 연구들의 장단점을 분석 보완하여 다음과 같은 특징을 갖는 동기화 기법을 제안하고 있다. 먼저, 본 연구의 동기화 기법은 동적인 네트워크 상태를 효과적으로 반영하고 있으며, 이를 위해 서버로부터 전송되는 데이터의 품질을 낮추어, 데이터의 양을 조절함으로써 동기화를 제어하고 있다. 또한 미디어내 동기화와

미디어간 동기화를 모두 제공하고 있다.

## 6. 결 론

본 연구에서는 클라이언트-서버 분산 환경에서 여러 서버에 분산되어 있는 미디어들을 사용하여 프리젠테이션을 수행시 가장 중요하게 고려되는 동기화 기법에 대해 제안하였다. 여러 서버로부터 미디어들을 전송받아야 하는 경우, 네트워크 상태에 따라 전송 시간이 달라지기 때문에, 이 네트워크 상태 변화를 검사하여 이를 서버에 피드백시킴으로서 전송되는 데이터의 양을 조절하도록 하여 동기화를 제어하고 있다. 또한 본 연구는 미디어 단위들이 클라이언트에 도착할 때, 늦게 혹은 빠르게 도착하는 미디어 단위들을 검사하고, 타임스탬프를 이용한 미디어 내 동기화 기법을 효과적으로 지원하고 있다. 또한 미디어 단위가 클라이언트에 도착할 때, 이를 저장하기 위한 버퍼의 빈공간이 있는지를 검사함으로써 네트워크 상태를 효과적으로 반영할 수 있다. 본 연구에서 제안하고 있는 동기화 모델은 분산환경에서의 프리젠테이션을 효과적으로 지원할 수 있을 뿐만 아니라, 사용자 맞춤(customized) 멀티미디어 프리젠테이션을 위한 적응적 QOS를 지원하는 시스템에도 유용하게 사용될 수 있을 것이다.

## 참 고 문 헌

- [1] L.LI, A. Karmouch and N.D. Georganas, "Multimedia Teleorchestra With Independent Sources: Part 1 and Part 2," ACM/ Springer -Verlag Journal of Multimedia Systems, Vol. 1, No.4, February, pp.143-165, 1994
- [2] T.D.C. Little and A. Ghafoor, "Synchro\_nization and storage Models for Multimedia Objects," IEEE J. on Selected Areas of Commun., Vol.8, No.3, 1990, pp.413-427, 1990
- [3] B. Prabhakaran and S.V. Raghavan, "Synch\_ronization Models For Multimedia Presentation With User Participation," ACM/Springer-Verlag, Journal of Multimedia Systems, Vol.2, No.2, pp. 53-62, 1994.
- [4] J.F. Allen, "Maintaining Knowledge about Temporal Intervalism," Communications of the ACM, Vol.26, No.11, pp.832-843, 1983
- [5] S.V. Raghavan, B Prabhakaran, and Satish K. Tripathi, "Synchronization Representation and Traffic Source Modeling in Orchestrated Presentation," IEEE J. on Selected Areas in Commun., Vol. 14, No. 1, 1995
- [6] Ralf Steinmetz, "Synchronization the Properites in Multimedia Systems," IEEE J. on Selected Areas in Commun., Vol.8, No.3, pp. 401-411, 1990
- [7] Petra Hoepner, "Synchronizing the Presen \_tation of Multimedia Objects", IEEE Computer Commun., Vol.15, No.9, pp. 557-564, 1992
- [8] K. Selcuk Candan, B. Prabhakaran, V.S. Subrahmanian, "CHIMP: A Framework for Supporting Distributed Multimedia Document Authoring and Presentation," In the Proc. of ACM Intl. Multimedia Conf., pp.329-340, 1996
- [9] D.A. Adjeroh and M. C. Lee, "Synch\_ronization Mechanisms for Distributed Multi \_media Presentation System," In the Proc. of ACM Intl. Multimedia Conf., pp. 30-37, 1995
- [10] J.F. Gibbon and T.D.C Little, "The Use of Network Delay Estimation for Multimedia Data Retrieval," IEEE Journal on Selected Areas in Commun., Vol.14, No.7, pp.1371-1387, 1996
- [11] J. Hui, E. karasan, J.Li, and J. Shang, "Client-Server Synchronization and Buffering for Variable Rate Multimedia Retrievals," IEEE J. on Selected Areas in Commun, Vol.14, No.1, pp.226-237, 1996
- [12] Y. Song, M. Miekle and A.Zhang, "Netmedia : Synchronization Streaming of Multimedia in Distributed Environments," In the Proc. of IEEE Intl. Conference on Multimedia Computing and System, pp. 585-590, 1999
- [13] G. Lu, Communication and Computing for Distributed Multimedia System, Artech House, pp.



275-304, 1996

[14] , "PST를 기반으로 하는 멀티미디어 프리젠테이션 모델, 정보처리학회 논문집, 1998

[15] , "A Multimedia Presentation Model Supporting Temporal and Spatial Synchronization Based on PST, In the Proc. of Intl. Symp. on Advances Intelligent Computing and Multimedia System, 1999

[16] J, Schnepf and et al., "doing FLIPS: Flexible Interactive Presentation Synchronization," IEEE J. on Selected Areas in Commun., Vol.14, No.1, pp.114-125, 1996



#### 최 숙 영

1988년 8월 전북대학교 이학사  
(전산학)

1991년 2월 전북대학교 대학원  
이학석사(전산학)

1996년 2월 충남대학교 대학원  
이학박사(전산학)

1996년 ~ 현재 우석대학교 컴퓨터교육과 조교수

관심 분야 : 멀티미디어응용, 원격교육, 병렬처리