

멀티미디어 데이터 스트리밍을 위한 미디어 동기방식 제어 알고리즘

신동진*, 김수창*, 이승민**, 송병권**, 정태의**

*한국전자통신연구원

**서경대학교 컴퓨터학과

e-mail : tejeong@skuniv.ac.kr

Media-Synchronization Control Algorithm for Multimedia Data Streaming

Dong-Jin Shin*, Su-Chang Kim*, Seung-Min Lee**,

Byung-Kwon Song**, Tae-Eui Jeong**

*Electronics and Telecommunications Research Institute

**Dept. of Computer Science, SeoKyung University

요 약

최근 고속 통신망과 멀티미디어 기술의 급속한 발전으로 다양한 멀티미디어 서비스에 대한 연구와 그에 대한 응용 개발이 활발하게 이루어지고 있다. 그러나, 멀티미디어 응용 서비스는 기존의 단일 미디어 서비스와는 달리 다양한 미디어의 통합된 전송을 필요로 하며, 개별 미디어의 정확한 연속성 및 관련된 미디어들 간의 적절한 시간적 동기화가 확보된 실시간 재생이 요구되는 고도화된 처리 기술을 필요로 하기 때문에 멀티미디어 동기화 알고리즘에 대한 연구는 필수적이고 매우 중요하다고 볼 수 있다. 본 논문은 Multimedia Data Streaming 을 위한 미디어 동기 방식 제어 알고리즘 고안 및 Lip Sync 방식 설계에 그 연구 목적이 있다.

1. 서론

최근 컴퓨터에서 멀티미디어처리 기술과 통신 기술의 비약적인 발전으로 멀티미디어 응용 부분이 급속히 발달되었다. 이것은 기존 데이터 및 음성중심의 서비스와 비교해 볼 때, 그래픽, 음성, 영상과 같이 사람들이 직접 인지하는 다양한 매체의 통합적인 활용에 따른 표현 능력, 정확한 의사 전달 능력 등 많은 장점을 가진다. 그러나 기술적인 발전과 이용자의 요구에 따른 자연스러운 서비스 변화와 함께 여러 가지 문제점들도 같이 나타나고 있는데, 특히 제공되는 서비스 품질(QoS: Quality of Service) 측면에서 해결해야 할 문제들에 직면하고 있다.

차세대 멀티미디어 시스템은 시간 의존적 및 시간 독립적 미디어를 컴퓨터 제어 하에 통합적으로 생성, 저장, 통신, 조작 및 프리젠테이션 하는 것으로 특징지어지는데, 원활한 통합을 제공하기 위한 다수의 기반기술에 대한 해결이 요구된다는 것이다. 이 같은 핵

심 이슈들 중 대표적인 것이 데이터의 디지털화 된 표현과 다양한 종류의 미디어간, 그리고 미디어 내에서 제공되는 동기화이다. 멀티미디어 시스템에서의 동기화는 시스템 내에 있는 미디어 객체간의 시간 관계를 말한다. 조금 더 넓은 의미로서 객체간의 내용(content), 공간(spatial), 시간(temporal) 관계로 구성된다 고 보기도 한다. 내용 관계란 내용적으로 관련 있는 데이터들간의 관계를 말한다. 공간 관계란 출력장치 상에 데이터가 프리젠테이션 될 위치와 관련된 관계를 말한다. 본 논문에서는 시간관계로 제한 시키고, 다루는 미디어 객체를 시간 의존적인 객체와 시간 독립적인 미디어 객체로 구별하여 사용한다.[1]

본 논문의 구성은 다음과 같다. 먼저 2 장에서 미디어 동기화 규격에 관한 일반적인 방법과 함께 페트리 넷에 기반한 기존의 연구 내용을 살펴보고, 제 3 장은 페트리 넷을 기반으로 더 확장된 모델을 정의하고, 4 장에서는 확장된 모델의 Protocol 을 정의하고, 마

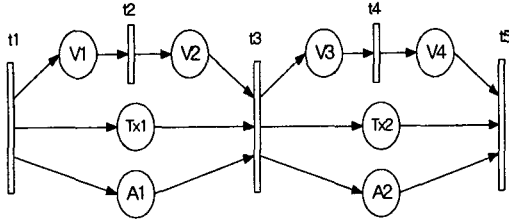
지막으로 5장에서 결론을 맺는다.

2. 기존 연구

기존 모델로는 Petri Net 을 기반으로 한 OCPN[2], XOCPN[3], RTSM[4], TFPN[5] 등이 있다

2.1 OCPN (Object Composition Petri-Net)

Petri-Net 은 1960 년대에 C.A. Petri 에 의해 소개된 이후 병렬 시스템 등의 시스템 동작 및 특성을 모델링 하고, 시뮬레이션 하는데 많이 사용되고 있다. Little 과 Ghafoor 에 의해 OCPN 이 소개된 이후 최근 멀티미디어 데이터의 모델링에 폭 넓게 이용되고 있다. OCPN 은 토큰소에 시간을 할당 함에 의해 동기화의 정도를 자유롭게 선택할 수 있다. OCPN 은 미디어 간의 시간 관계를 객체 수준에서 기술하며, 비디오, 오디오, 텍스트 등의 객체를 가진 멀티미디어 응용에서의 동기화 관계를 나타낼 수 있다.



<그림 1> OCPN Model

2.2 XOCPN (Extended Object Composition Petri Net)

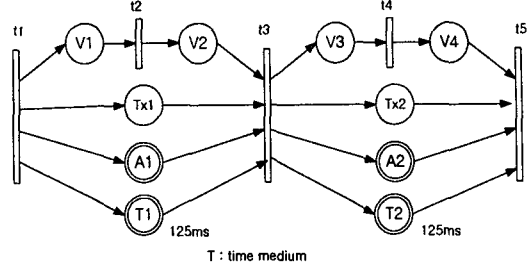
XOCPN 은 객체를 Synchronization Interval Units (SIU's)으로 나누어 설명하고 있다. SIU 는 하나의 객체, 예를 들면 Video1, Audio1 이런 식으로 되어 있는 객체를 다시 시간 단위로 나누어 놓은 것으로 개념을 더 세분화 시켰고, Inter-object Synchronization Points(ISP's) 와 Inter-stream Pacing Points (IPP's)의 두 종류 points 를 두어서 inter-stream 간의 동기화를 맞춘다. 기존의 OCPN 을 더욱 세분화 시켜, 동기화와 connection control 을 할 수 있게 확장 시켰다. Connection control 을 위해 초기 접속 SIU 와 접속 해제 SIU 를 사용한다.

2.3 RTSM (Real Time Synchronization Model)

RTSM 에서는 2 종류의 places 가 있는데, regular places 와 enforced places 이다. Regular places 에서 강화 되는 enforced places 는 구개의 원으로 표현되며, enforced places 가 fire 되면 다른 객체에 관계없이 다음 place 에 token 이 넘어간다. 즉, QoS 를 보장하기 위한 방안이 추가 되었다. ts 에서 transition 이 시작되어, enforced place 에서 token 이 unblock 되면, transition 이 끝나는 시점인 ti 에서 backtracking 이 시작된다. 다른 멀티미디어 객체에 token 이 할당되어 block 되는 것을 멈추게 해서, 다음 transition 을 시작하기 위한 작업을 하게 되는 것이다.

Enforced places 의 key medium 을 결정하는 것은 첫 번째 medium 의 중요도이고, 두 번째는 얼마나 jitter

에 민감한가 이다. 예를 들면 다른 것 보다 한 medium 의 quality 가 높다면 그것이 key medium 이 되는 것이고, 모두 다 똑같다면, 그 중에서 가장 jitter 에 민감한 것이 key medium 이 되는 것이다.

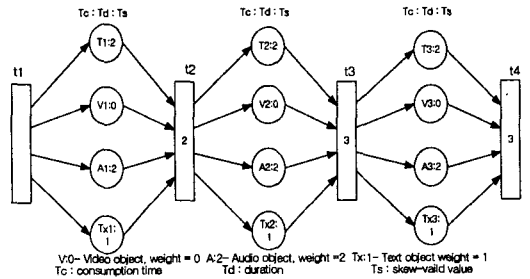


<그림 2> RTSM 의 Time Medium

그러나 key medium 자체에 대한 지연 문제가 발생할 수도 있기 때문에 절대적인 시간 간격을 나타내는 virtual time medium 을 정의하였다. Key medium 의 지연이 길어지더라도 가상의 time medium 은 일정한 시간 간격이 지나게 되면 점화되어 key medium 자체의 지연으로 인한 서비스 품질 저하를 보장하고 있다.

2.4 TFPN (Transition Function Petri Net)

각각의 객체가 가중치를 가지며 각 transition 은 자신의 점화식을 가진다. 오디오객체가 가중치 2, 텍스트 객체가 가중치 1 을 가지고 transition 의 점화조건식을 3 보다 크거나 같게 함으로써 2 개의 객체에 대한 QoS 를 가지게 할 수 있다. 또한 미디어간 동기화 하기 위한 QoS 파라미터를 두어 skew 를 반영하였다.



<그림 3> TFPN

3. 제안된 동기화 모델

새로 제안하는 모델은 EFTN(Extend Transition Function Petri-Net)으로 기존 모델들의 문제점을 파악하여 그 문제를 해결하며, 개선하는데 중점을 두었다.

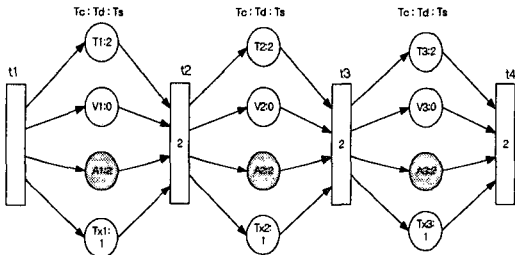
3.1 기존 모델들의 문제점

먼저 OCPN 과 XOCPN 에서의 문제점은 어떤 특정 객체에 대해서 QoS 를 제공하지 못하는 점이었고, RTSM 에서는 특정 객체 하나에 대해서만 QoS 를 제공하며, 두개 이상이 될 때에는 제공할 수 없었다. 또 skew 를 반영하지 못하는 단점이 있었다. TFPN 에서는

앞의 두 가지 문제는 해결하였으나 가중치가 높은 객체가 계속 먼저 fire 되면 나머지 객체들은 전송이 이루어지지 않은 상태에서도 계속 backtracking에 의해 stop 되어진다. 중간에 점화식의 가중치를 높일 수 있으나 계속 무시되는 객체에 대한 최소한 quality를 보장할 수는 없다.

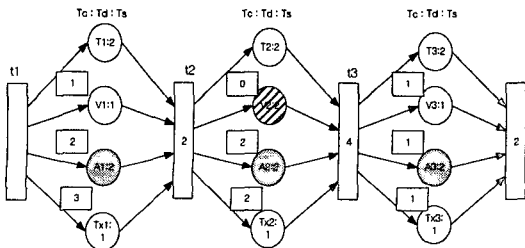
3.2 ETFPN (Extend Transition Function Petri-Net)

- ETFPN의 정의는 다음과 같다.
- $E = \{T, P, A, D, W, G, R, M, S\}$
- $T = \{t_1, t_2, t_3, \dots, t_n\}$ transitions.
- $P = \{p_1, p_2, p_3, \dots, p_m\}$ places (single circles).
- $A: \{T * P\} \cup \{P * T\}$ directed arcs.
- $D: P \rightarrow R, R+$ 는 실수 시간 주기(interval)
- $W: P \rightarrow I, I = \{0, 1, 2, 4, 8, \dots\}$ 가중치
- $G: T \rightarrow I, I = \{0, 1, 2, 3, 4, \dots\}$ 점화값
- $R: P \rightarrow \{r_1, r_2, r_3, \dots, r_k\}$ resource(type of place.)
- $M: P \rightarrow \{0, 1, 2\}$ marking(state of places.)
- $S: P \rightarrow I, I = \{0, 1, 2, 3, 4, \dots\}$ state maintenance



<그림 4> 높은 가중치의 객체로 인한 낮은 가중치 객체의 빈곤현상

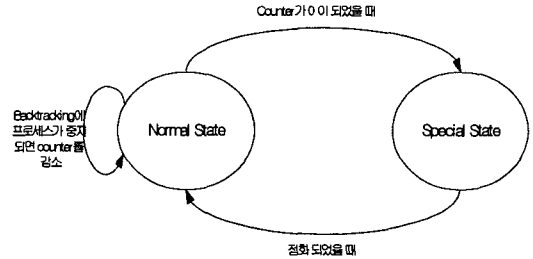
위의 <그림 4>와 같이 audio 객체의 중요도가 가장 높다. 여기서 audio 객체가 빨리 전송되면 transition에서는 점화값이 충족되므로 fire 된다. 그래서 해당 transition을 기준으로 아직 도착하지 않은 다른 객체들에 있는 token을 backtracking을 통해서 멈추게 한다. 이와 같은 작업은 다음 transition에 도착했을 때도 마찬가지로 이루어지게 된다. Audio 객체가 계속적으로 빨리 도착한다면, 다른 객체들에 대해서는 아무런 보장을 해줄 수 없게 된다. 이를 해결하는 방법은 가중치의 값을 변경하거나 점화값을 변경하는 방법이 있을 수 있겠지만, 점화값을 언제 어떻게 바꿔줘야 하는지에 대한 check가 늘 이루어져야만 한다. 그래서, 각 객체의 최소 quality보장에 관련된 state maintenance 값을 설정한다. 아래 그림 2와 같이 각 places의 사각형에 들어가는 값이 state maintenance 값이다.



<그림 5> ETFPN

Maintenance 값은 backtracking에 의해 stop을 하게 되면 감소하고, 해당 객체가 점화하게 되면 그 값이 원래의 값을 가지게 된다. 값이 0이 되게 되면 해당 객체의 가중치를 일시적으로 가장 높게 주어서 해당 객체에 대한 보장을 해주게 된다. 그림 2에서 보면 초기에는 audio 객체가 높은 가중치를 가지고 있고 첫 번째 transition에 먼저 도착하여 점화되었다. 비디오 객체의 state maintenance 값은 1을 가지고 있으므로 해서 0으로 감소되면서 해당 객체의 가중치가 증가된다. 다시 fire되게 되면 원래대로의 값으로 복귀하게 된다. 이 값으로 인하여 최소한의 quality를 보장 받게 된다.

State maintenance 값의 변화를 살펴보면. 먼저, default 값 즉 해당되는 객체가 적어도 얼마 만에 한번은 점화되어야 한다는 것을 염두에 두고 셋팅되어야 한다. 셋팅된 값은 각 transition에서 해당 객체가 점화되지 못하고 프로세스도중 backtracking에 의해 중단되는 경우에만 감소하게 된다. 그 값이 계속 감소하게 되어서 0이 되면 객체의 weight가 최고치가 되면서 해당 객체가 점화될 때 까지 기다리게 된다. 이 값은 두개의 상태를 가지게 되는데 backtracking에 의해 감소되는 normal state와 가장 높은 가중치를 임의로 갖게 되는 special state 두개의 상태를 가지게 된다. 다음 <그림 6>에서 그 상태의 변화를 보인다.



<그림 6> State Maintenance 상태 천이도

이 알고리즘에서 가장 중요시 하고 있는 점은 모든 객체들의 수용, 즉 풀고루 분포된 객체의 quality보장에 있다. 어느 한쪽에 치우치는 QoS가 아닌, 객체에 최소의 quality 제한을 줌으로 해서 모든 객체의 점화를 기본 개념으로 하고 있다. 하지만 실제적으로는 time에 가장 높은 가중치가 주어지게 되며, 점화 시키고자 하는 객체의 timeout이 발생한 경우에는 아무리 높은 가중치를 준 special state 상태라 하더라도 그냥 점화되게 된다. Real time multimedia의 성격을 그대로 가지게 된다. 그러므로, 이 알고리즘은 가장 중요한 시간에 중점을 두면서도 최대한으로 모든 객체를 수용하여 같이 점화하고자 하는 것이다.

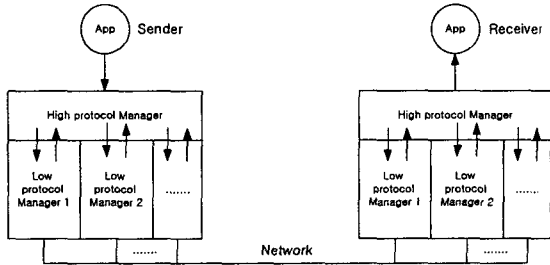
4. 프로토콜

이 장에서는 앞에서 설명된 모델에 따라 제안된 프로토콜에 대한 내용이다.

4.1 프로토콜 구조

<그림 7>과 같이 전체적인 구조는 각각의 하위 프

로토콜 매니저에서 받은 객체들을 위에 있는 상위 프로토콜매니저에서 모아서 application에 제공하게 된다. 그림에 나타난 것처럼 상위 프로토콜에서 일어나는 일들은 제안된 모델인 ETFPN에 의해서 처리하게 된다.



<그림 7> Architecture of Protocol

4.2 Packet Format

Packet 형식은 control packet 과 data packet 으로 나뉘며, control packet은 각 객체의 connection을 구별하고, 시간을 제한한다. <그림 4-8>은 control packet의 형식을 보여주고 있다.

Packet Type	Key SCID (Sub-Connection)	Time Media Duration
-------------	---------------------------	---------------------

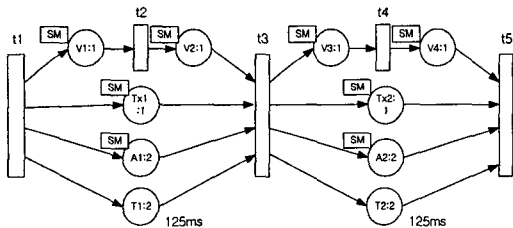
<그림 8> Control Packet

C	SCID for audio medium	125ms
---	-----------------------	-------

<그림 9> Control Packet Example

D Packet Type	SCID	Weight	PC	TID	SEQ	SM	Other Header Information	Data Portion
---------------	------	--------	----	-----	-----	----	--------------------------	--------------

<그림 10> Data Packet Format



<그림 11> EFSM Example

<그림 11>을 packet으로 표현한 것이 <그림 12>이다. 각 data packet의 각 필드에 그 값들이 setting된 상태이다. 예를 들어 A1을 보면 data packet 표시인 "D"가 있고, 그 다음 SCID(Sub-Connection Identifier) 각 media object 별로 Connection ID가 있다. 다음 필드는 weight로써 각 object connection 별로 QoS를 제공하게 되는 가중치의 값을 가진다. PC(Place Counter) 필드와 TID(Transition ID) 필드는 동시에 들어오는 각각

의 객체들이 해당 transition에 제대로 들어갈 수 있게 control해주는 것으로, 여기서 V1과 V3 사이에 하나의 transition이 있고, 그 Transition ID로 식별하게 된다. SEQ(Sequence number) 필드는 각 해당 object의 순서를 나타내는 것이다. SM(State Maintenance)은 해당 객체의 최소 QoS를 보장하는 것으로 앞에서 설명했던 state maintenance 값이다. Other header information에서는 port 번호나 QoS 등 여러 가지를 포함하고, Data는 실제 data가 들어가게 된다.

A1	D	1	2	X	X	1	SM	OTHER	Data
Tx1	D	2	1	X	X	1	SM	OTHER	Data
V1	D	3	1	1	T2	1	SM	OTHER	Data
V2	D	3	1	X	X	2	SM	OTHER	Data
A2	D	1	2	X	X	2	SM	OTHER	Data
Tx2	D	2	1	X	X	2	SM	OTHER	Data
V3	D	3	1	1	T4	3	SM	OTHER	Data
V4	D	3	1	X	X	4	SM	OTHER	Data

<그림 12> data packet example

5. 결론

본 논문은 분산 멀티미디어 시스템 및 스트리밍 서비스 제공에 있어 핵심적 기술인 동기화를 하기 위한 요구 사항들과 규격 및 기준에 대해 살펴 보았다. 그리고, 기존에 제안되었던 OCPN, XOCPN, RTSM, TFPN 등의 메커니즘을 비교 분석하였으며 이러한 분석을 바탕으로 하여 새로운 멀티미디어 동기화 규격 모델을 제안하였다. 앞으로의 연구 방향은 기존의 모델과 제안된 모델의 성능 비교 분석을 위하여 시뮬레이션을 할 예정이다.

참고문헌

- [1] K.Ravindran, "Realtime Synchronization of Multimedia Data Streams in High Speed Networks," proc. Of IEEE Multimedia '92, 1992.
- [2] T.D.C.Little and A.Ghafoor, "Multimedia Synchronization Protocols for Broadband Integrated Services," IEEE Journal on Selected Areas in Communication, VOL 9, No 9, Dec. 1991
- [3] Miae Woo, "A Synchronization Framework for Networked Multimedia Services", pp.1-36, December 1995
- [4] Chun-Chuan Yang, Jau-Hsiung Huang, "A Multimedia Synchronization Model and Its Implementation in Transport Protocols", IEEE Journal on Selected Areas in Communications, VOL. 14. NO. 1, January 1996
- [5] Yoo, Sang-Shin, "Unified Multimedia Synchronization Mechanism Based on High-Speed Networks, pp.6-36, December 1996
- [6] Chung-Ming Huang, "An EFSM-Based Multimedia Synchronization Model and the Authoring System", IEEE Journal on Selected Areas in Communications, VOL 14 NO 1, January 1996