

QoS 기반의 미디어내 동기화 설계

박 흥 진[†] · 이 준 연^{††} · 김 창 민^{†††}

요 약

멀티미디어 응용 서비스를 제공하기 위해 기존 데이터 통신과 비교하여 멀티미디어 통신은 여러 가지 요구사항을 지원해야 한다. 특히, 분산 환경에서 연속적으로 생성된 미디어를 목적지에서 시간적인 관계를 유지하면서 재생시키킬 수 있는 동기화 메커니즘은 무엇보다도 중요한 문제이다. 그러나, 생성지에서 일정하게 생성된 미디어는 전송 에러나 네트워크 전송 지연 시간 때문에 목적지에서는 일정한 시간 간격으로 재생을 보장받지 못한다.

본 논문은 네트워크를 통해 전송되는 오디오나 비디오 같은 멀티미디어 스트림의 연속적인 재생을 보장하기 위해 미디어내 동기화 메커니즘을 제안한다. 제안된 모델은 QoS기반의 버퍼크기와 재생 보정 알고리즘을 통해 목적지에서 연속적인 미디어 재생을 보장한다.

The Design of Intramedia Synchronization based QoS

Hong-Jin Park[†] · Jun-Yeon Lee^{††} · Chang-Min Kim^{†††}

ABSTRACT

In contrast to traditional data communication, the multimedia communication must support variable requirements to multimedia application service. Specially, it is a critical issue that the synchronization supports a continuous and temporal relationship media playout in a distributed environment. But the continuously generated media in source can't guarantee the constant interval on playout for the destination due to the inconsistent network transmission delay and transmission error.

In this paper, we introduce a intramedia synchronization mechanism for the continuous and synchronous playout of multimedia stream such as audio and video across a communication network. The proposed mechanism is based on the user defined QoS(Quality of Service) and the playout correction algorithm to guarantee the continuous media playout at the receiver.

1. 서 론

최근 광대역 네트워크와 멀티미디어 기술 발전으로 다양하고 새로운 멀티미디어 응용 서비스들이 나타나고 있다. 네트워크 기반의 멀티미디어 서비스에는 멀티미디어 정보 시스템, 화상회의 시스템, 원격 의료 진

료 시스템, 주문형 비디오 시스템등이 있으며 이들은 현재 연구·개발되고 있다[1][2].

분산 멀티미디어 응용 서비스를 제공하기 위해서 기존의 비 실시간 단일 미디어에 적합하게 설계된 데이터 통신에 비해 충족되어야할 다양한 요구사항이 있다. 이러한 멀티미디어 통신 요구사항은 다음과 같다. 첫째, 동적인 QoS(Quality of Service)를 사용자에게 제공할 수 있어야 한다. 멀티미디어 응용 서비스를 제공받는 사용자는 자신이 원하는 QoS를 요구하면 이

[†] 정 회 원 : 중앙대학교 대학원 컴퓨터공학과

^{††} 준 회 원 : 중앙대학교 대학원 컴퓨터공학과

^{†††} 정 회 원 : 성결대학교 컴퓨터학부 교수

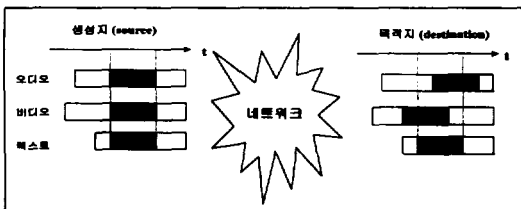
논문접수 : 1999년 5월 20일, 심사완료 : 1999년 7월 16일

에 맞는 통신이 동적으로 재설정되어 서비스를 제공해야 한다. QoS란 속도율, 지터(jitter), 비트 에러율, 패킷 에러율등을 포함하는 변수들의 집합이다. 둘째, 멀티미디어 통신은 연속적인 데이터 전송이 보장 되어야 한다. 연속미디어(오디오, 비디오) 데이터는 고속 전송이면서 지연에 민감하고 약간의 전송 에러를 허용할 수 있어야 하며, 지속적인 미디어 전송을 보장해야 한다. 셋째, 효율적인 멀티 캐스팅을 지원할 수 있어야 한다. 화상 회의 시스템, 주문형 멀티미디어 시스템 등 여러 멀티미디어 응용 서비스는 멀티 캐스팅이 요구된다. 넷째, 멀티미디어 통신은 동기화(synchronization)를 보장해야 한다. 멀티미디어 통신에 필요한 다양한 요구 사항 중 동기화는 특히 중요하다. 본 논문은 멀티미디어 통신 요구사항 중에서 동기화에 초점을 맞추었다.

멀티미디어 동기화란 미디어들 사이에 존재하는 시간적 관계를 의미한다[4]. 예를들어, 화상 회의 시스템에서 사람이 말하는 음성(voice)과 입의 움직임(video)은 일치되어 재생되어야 하는데 이를 립싱크(lip-sync)라 한다. 멀티미디어 동기화에는 크게 미디어간(inter-media) 동기화와 미디어내(intramedia) 동기화로 구분된다. 미디어간 동기화에는 립싱크를 유지하는 것과 같은 미디어들 간의 동기화를 맞추는 것을 의미하며, 미디어내 동기화는 단일 미디어 내에서 시간적 순서를 보장하는 것으로 정의된다.

Little에 의하면 저장되어 있는 미디어를 검색할 때 미디어들을 인위적으로 동기화를 맞추는 것을 synthetic 동기화라 하고, 미디어 데이터들이 생성과 함께 동기화가 결정되는 것을 live동기화라 한다. synthetic동기화에서는 미디어 데이터들이 멀티미디어 서버에 저장되어 있기 때문에 생성율과 데이터 검색율을 조정함으로써 소비율을 보다 크게 할 수 있다. 그러나, 데이터 생성과 함께 동기화가 결정되는 live동기화에서는 미디어 생성율과 소비율은 동일하며 일정하다[3][5][6].

(그림 1)에서 나타난 것처럼 생성지에서 연속적이고



(그림 1) 네트워크 전송 지연으로 인한 미디어 불일치

일정한 시간 간격으로 생성된 미디어들이 네트워크를 통해 전송될 때 일정하지 못한 네트워크 전송 지연과 전송 에러로 인해 목적지에서 연속적인 재생을 보장받지 못한다[7][8][9][10].

본 논문은 생성지에서 연속적으로 생성되는 미디어를 목적지에서 일정한 시간 간격으로 재생을 보장하는 미디어내 동기화를 제공하는데 있다. 이를 위해 사용자가 정의하는 QoS를 고려한 최적의 버퍼크기를 제안하고, 재생 보정 알고리즘을 통해 목적지에서 연속적이고 일정한 시간 간격으로 미디어 재생을 보장한다.

본 논문의 구성은 다음과 같다. 2장에서 본 논문이 제안하고 있는 미디어내 동기화 모델 구조를 기술하고, 3장에서는 미디어내 동기화를 제공하기 위한 버퍼에 대해 설명한다. 4장에서 도출된 미디어의 동기화를 맞추기 위해 재생 보정 알고리즘을 기술하며, 5장에서 본 논문의 성능을 평가한다. 마지막으로 6장에서 결론을 맺는다.

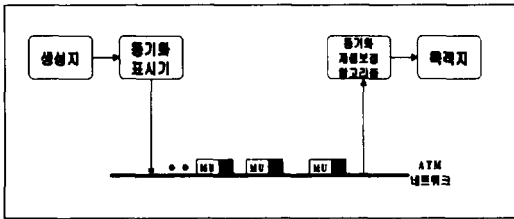
2. 동기화 모델

본 논문에서 동기화 모델 구조의 전체 조건은 다음과 같다. 미디어 데이터들의 전송 단위는 미디어 단위(media unit)로 한다. 예를들어 비디오는 프레임(frame), 오디오는 샘플(sample)의 집합이 하나의 미디어 단위로 취급되어 처리된다. 이러한 미디어 단위들이 네트워크를 통해 전송될 때, 동기화 정보를 제공하는 방법으로 동기화 표시기(synchronization marker)를 이용하며, 본 논문에서 동기화의 표시기로는 일련 번호(sequence number)를 사용한다. 즉, 생성지에서 생성된 각각의 미디어 단위들에게 동기화 표시기인 일련 번호를 추가하여 전송시키면, 목적지에서는 동기화 표시기로 표시된 일련 번호를 검사하여 동기화를 보장해준다. 또한, 생성율과 재생율이 동일한 live동기화를 기반한 미디어내 동기화에 초점을 맞추었다.

ATM 네트워크에서는 초기화 선택 동안 QoS가 만족되어 생성지에서 목적지까지 연결이 되면 ATM 네트워크의 전송 단위인 셀(cell)간에 순서가 보장된다(cell-sequence integrity). 또한 Rangan에 의하면 ATM 네트워크에서 자원 예약과 허가 제어를 통해 네트워크 지 시간의 최대 변화 범위를 보장할 수 있다[12].

본 논문에서는 이러한 ATM 네트워크의 특성을 기반으로 하여 미디어 단위들간의 전송 순서는 보장되며,

Rangan에 의해 미디어 단위들이 생성지에서 목적지로 네트워크를 통해 전송될 때 걸리는 네트워크 전송 지연시간을 최대 네트워크 전송 지연시간과 최소 네트워크 전송 지연시간 사이로 전송됨을 전제로 한다. 이러한 본 논문의 전제를 기반한 미디어내 동기화 모델 구조는 (그림 2)와 같다.



(그림 2) 미디어내 동기화 모델 구조

(그림 2)는 생성지에서 생성되는 미디어들에게 본 논문에서 사용하는 동기화 전송 정보로 동기화 헤더에 일련 번호를 미디어 단위들에게 추가시켜 ATM 네트워크를 통해 전송시킨다. ATM 네트워크에서는 각 미디어 단위들이 한정된 네트워크 전송 지연을 가지고, 미디어 단위들간 순서가 보장되어 목적지로 전송된다. 목적지에 도착된 미디어 단위가 허용될 수 있는 지터 범위로 도착되면 재생 보정 알고리즘을 통해 연속적이고 일정한 시간 간격 주기로 재생시켜 동기화를 보장해 준다. 만약, 허용될 수 없는 지터 범위로 미디어 단위들이 도착되는 경우나 전송 도중 손실된 경우 미디어 단위는 재생 보정 알고리즘을 통해 보정시켜 준다.

3. 동기화 제공을 위한 버퍼

미디어내 동기화를 제공하기 위해 목적지에서 버퍼가 필요하며, 이에 따른 버퍼 크기가 중요한 문제이다 [10][13].

3.1 목적지의 버퍼 크기

생성지에서 연속적이고 일정한 시간 간격으로 생성된 미디어 단위들이 네트워크를 통해 전송될 때, 목적지에 도착되는 시간을 식으로 나타내면 다음과 같다 (여기에서 $k=1..n$)

$$MU_k = D_k + (k-1) \times t \quad (1)$$

식 (1)에서 MU_k 는 생성지에서 생성된 k번째 미디어

단위를 나타내고, D_k 는 k번째 미디어 단위가 생성지에서 목적지까지 전송될 때 걸리는 시간(네트워크 전송 지연시간)을 나타낸다. t는 생성율의 기본 단위인 생성 주기를 나타내는데, 본 논문은 생성율과 재생율이 같은 live동기화를 전제로 하였으므로 t는 재생율의 기본 단위인 재생 주기도 의미한다. 예를 들어, 비디오 미디어는 생성지에서 1초에 30 프레임이 생성된다면 생성 주기는 1/30이고, 이러한 생성 주기로 네트워크를 통해 전송시킨다. 네트워크를 통해 전송된 미디어 단위는 목적지에서도 1/30의 재생 주기로 비디오 미디어 프레임을 재생시켜야한다.

식 (1)에서 첫 번째 미디어 단위 MU_1 는 생성지에서 생성되어 목적지까지 전송될 때 걸리는 네트워크 전송 지연시간이 D_1 이고, 두 번째 미디어 단위 MU_2 가 가지는 네트워크 전송 지연시간 D_2 와 생성지에서는 첫 번째 미디어 단위 MU_1 을 전송한 후에 생성주기 t 시간 후에 두 번째 미디어가 전송되므로, 실질적으로 두 번째 미디어 단위가 목적지에 도착되는 시간은 $D_2 + t$ 시간이다. 계속해서 생성지에서는 미디어 단위들을 생성주기 t 시간만큼의 차이로 미디어 단위들이 전송되므로 k번째 미디어 단위 MU_k 가 목적지에 도착되는 실제적인 시간은 $MU_k = D_k + (k-1) \times t$ 시간이고, k+n 번째 미디어 단위 MU_{k+n} 가 목적지에 도착되는 실제적인 도착 시간은 $D_{k+n} + (k+n-1) \times t$ 시간임을 알 수 있다.

또한, 목적지에서는 임의의 미디어 단위가 도착해서 재생 주기 t시간 동안 재생되고 있을 때, 그 재생 주기 t 시간 내에 도착되는 미디어 단위들은 저장할 수 있어야 한다. 즉, 위의 식 (1)에서 k번째 미디어 단위를 기준 미디어 단위라고 하면, 그 기준 미디어 단위가 목적지에 도착 후 재생하고 있는 동안 즉, 재생 주기 t 시간보다 작거나 같은 시간으로 도착되는 미디어 단위들은 버퍼에 저장되어야 한다. 이를 위의 식에서 k번째 미디어 단위와 k 번째에서 임의의 n번째로 도착되는 k+n번째 미디어 단위 MU_{k+n} 에 적용시키면 다음과 같은 식을 나타낼 수 있다.

$$(D_{k+n} + (k+n-1) \times t) - (D_k + (k-1) \times t) \leq t$$

이를 n에 대해 정리하면 다음과 같다.

$$n \leq \frac{D_k - D_{k+n}}{t} + 1 \quad (2)$$

위의 식 (2)가 의미하는 바는 임의의 한 미디어가 재생 주기 동안 재생되고 있을 때, 그 다음 n개의 미디어 단위가 재생 주기안에 도착할 수 있음을 나타낸다. 따라서, 목적지에서는 현재 재생되고 있는 미디어 단위와 재생 주기 t 시간 내에 도착 가능한 n개의 미디어 단위들을 버퍼에 저장할 수 있도록 n+1개의 버퍼가 필요하다.

또한, ATM 네트워크에서는 미디어 단위간 전송 순서가 보장되는 특성으로 인해 k+n번째 미디어 단위가 목적지에 도착되는 시간은 k번째 미디어 단위가 목적지에 도착되는 시간보다 크다. 이를 위의 식 (2)에 대입하면 다음과 같다.

$$D_k + (k-1) \times t < D_{k+n} + (k+n-1) \times t$$

이를 n에 대해 정리하면 다음과 같다.

$$\frac{D_k - D_{k+n}}{t} \leq n \tag{3}$$

식 (2)와 식 (3)를 정리하면, n에 대해 다음과 같은 일반적인 식 (4)를 얻어낼 수 있다.

$$\frac{D_k - D_{k+n}}{t} < n \leq \frac{D_k - D_{k+n}}{t} + 1 \tag{4}$$

위의 식 (4)에서 n은 임의의 k번째 미디어 단위가 목적지에 도착해서, 재생 주기 t 시간 동안 재생되고 있을 동안, 그 재생 주기 시간 동안 도착될 수 있는 미디어 단위들을 나타낸다. 따라서, 목적지에서 필요한 버퍼 크기는 현재 재생되고 있는 미디어 단위와 재생 주기 t 시간 내에 도착 가능한 n개의 미디어 단위를 저장할 수 있도록 n+1이 되어야 한다. 이를 식 (4)에 대입하고, n+1를 n'라고 하면 식 (5)와 같이 네트워크 상에 존재하는 전송지연을 해결하기 위해 목적지에서 사용되는 버퍼의 크기를 알 수 있다.

$$\frac{D_k - D_{k+n}}{t} + 1 < n' \leq \frac{D_k - D_{k+n}}{t} + 2 \tag{5}$$

3.2 허용 지터에 따른 버퍼 크기

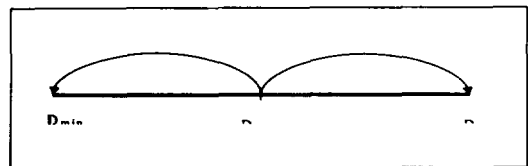
네트워크 전송 지연을 해결하기 위해 사용되는 버퍼의 크기를 나타내는 식 (5)를 이용하여 두 개의 식을 유도한다. 첫 번째로 임의의 미디어 단위가 재생 되고 있는 동안 목적지에서의 도착된 미디어 단위들이 버퍼에 모두 저장될 수 있는 경우를 만족시킬 수 있도록 n

이 가질 수 있는 최대의 경우를 알아본다. 두 번째로 임의의 미디어 단위가 재생되고 있는 동안 목적지에서의 도착된 미디어 단위들이 평균 네트워크 전송 지연 시간에서 사용자가 정의하는 허용될 수 있는 지터 범위 내에 도착되는 미디어 단위들만 목적지에 저장시킬 때의 버퍼 크기를 알아본다.

먼저 식 (5)를 이용하여 임의의 미디어 단위가 목적지에서 재생되고 있는 동안 도착된 미디어 단위들이 버퍼에 모두 저장될 수 있는 경우를 만족시킬 수 있도록 n이 가질 수 있는 최대의 경우를 고려해 보자. n이 가질 수 있는 최대의 경우는 식 (5)에서 $D_k - D_{k+n}$ 값이 최대일 경우가 된다. $D_k - D_{k+n}$ 이 최대일 경우는 k번째 미디어 단위 MU_k 가 네트워크를 통해 걸리는 네트워크 전송 지연시간이 최대 네트워크 전송 지연시간(D_{max})이 걸려 목적지에 도착되고, k+n번째 미디어 단위 MU_{k+n} 는 네트워크를 통해 걸리는 네트워크 전송 지연 시간이 최소 네트워크 전송 지연시간(D_{min})으로 걸려 목적지에 도착될 때, $D_k - D_{k+n}$ 값이 최대이다. 또한 이때 n 값도 최대가 됨을 알 수 있다. 이 경우를 식 (5)에 대입하면, 다음과 같은 식을 얻을 수 있다.

$$\frac{D_{max} - D_{min}}{t} + 1 < n' \leq \frac{D_{max} - D_{min}}{t} + 2 \tag{6}$$

식 (6)이 의미하는 바는 생성지에서 k 번째로 생성된 미디어 단위 MU_k 가 네트워크를 통해 목적지로 최대 네트워크 전송 지연시간이 걸려 목적지에 도착해서, 재생되고 있는 동안, 다음 미디어 단위들이 네트워크를 통해 목적지로 최소 네트워크 전송 지연시간으로 도착될 때, 목적지에서의 최대 버퍼 크기는 n+1개가 필요함을 나타낸다.



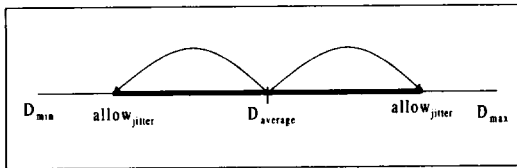
(그림 3) 한정된 네트워크 전송 지연시간에서 최대 허용지터

(그림 3)은 식 (6)에 대한 설명으로, 이 경우에는 임의의 미디어 단위가 재생되고 있는 동안에 도착된 모든 미디어 단위들을 버퍼에 저장할 수 있도록 버퍼의

크기를 고려한 경우이다. 이경우 생성된 미디어 단위들이 최대 네트워크 전송 지연시간과 최소 네트워크 전송 지연시간으로 전송지연 변화가 최대 변경되므로 지연 변화(delay variance) 즉, 지터의 범위가 최대가 됨을 알 수 있다. 허용할 수 있는 지터의 범위가 최대가 되므로 재생 주기 내에 목적지에 도착된 미디어 데이터들은 목적지에서 제거(dropout)되지 않는다.

두 번째로 식 (5)을 이용하여 임의의 미디어 단위가 재생될 동안에 목적지에서의 도착된 미디어 단위들이 평균 네트워크 전송 지연시간에서 허용될 수 있는 지터 범위 내에 도착되는 미디어 단위들만 목적지에 저장시킬 때의 버퍼 크기를 알아보자. 이 경우에서 사용자가 요구한 지터의 범위 내에 도착된 미디어 단위들만 목적지 버퍼에 저장시킬 수 있도록 버퍼의 크기를 정한다. 목적지에 도착된 미디어 단위가 허용 지터의 범위를 넘어 너무 빠르게 도착될 경우, 이러한 미디어 단위를 버퍼에 저장시키면 목적지에서의 버퍼는 오버플로우가 발생되므로 제거시킨다. 또한 목적지에 도착된 미디어 단위가 허용 지터 범위를 넘어 너무 늦게 도착된 경우, 목적지에는 이미 쓸 못 없는 미디어 단위이므로 전송 도중 잃어버린 것으로 간주되어 처리한다.

따라서, 목적지 버퍼에 저장될 수 미디어 단위는 임의의 미디어 단위가 재생되고 있는 동안, 평균 네트워크 전송 지연시간에서 허용될 수 있는 지터 범위 안에 도착된 미디어 단위들이다



(그림 4) 평균 네트워크 전송 지연과 허용지터

(그림 4)에서와 같이 생성지에서 생성된 미디어 단위가 네트워크를 통해 전송될 때, 허용될 수 있는 최대 네트워크 전송 지연시간은 $D_{average} + allow_{jitter}$ 이고, 허용될 수 있는 최소 네트워크 전송 지연시간은 $D_{average} - allow_{jitter}$ 이다. 이 범위 안에 만족되지 못한 미디어 단위들은 허용 할 수 있는 지터 범위를 넘어선 경우가 되며, 목적지 버퍼에 저장시키지 않는다. 이 경우를 식 (5)에 대입하면 다음과 같은 식을 얻을 수 있다.

$$\frac{(D_{average} + allow_{jitter}) - (D_{average} - allow_{jitter})}{t} + 1 < n'$$

$$\leq \frac{(D_{average} + allow_{jitter}) - (D_{average} - allow_{jitter})}{t} + 2$$

이를 정리하면 다음과 같은 식 (7)을 얻을 수 있다.

$$\frac{2 \times allow_{jitter}}{t} + 1 < n' \leq \frac{2 \times allow_{jitter}}{t} + 2 \quad (7)$$

식 (7)이 의미하는 바는 생성지에서 미디어 단위가 네트워크를 통해 전송된 후 재생 되고 있는 동안, 평균 네트워크 전송 지연시간과 허용될 수 있는 지터 사이에 도착되는 미디어 단위들을 나타내므로 n' 개는 목적지에서 저장해야 할 미디어 단위의 개수가 되며 목적지에서의 버퍼 크기임을 알 수 있다.

또한, 식 (7)에서 나타난 것 같이 사용자가 어느 정도까지 허용 지터를 적용시키는 것에 따라 버퍼 크기는 가변적임을 알 수 있다. 허용 지터 범위가 크면 사용자의 QoS를 높일 수 있는 반면, 허용 지터 범위가 크기 때문에 목적지에서의 버퍼 크기도 증가할 것이다. 만약, 허용 할 수 있는 지터의 범위가 작으면 QoS를 낮추는 결과를 초래하는 대신 목적지에서 버퍼 크기는 상대적으로 줄어들 것이다. 따라서 식 (7)의 경우에 중요한 사항은 사용자가 요구하는 QoS중에 허용할 수 있는 지터의 범위 따라 최적의 버퍼의 크기를 정하므로써 사용자의 요구에 맞도록 QoS를 보장 시키는 것이다.

4. 재생 보정 알고리즘

생성지에서 연속적이고 일정한 시간 간격으로 생성된 미디어 데이터들이 네트워크를 통해 전송될 때 네트워크 상에 존재하는 전송지연과 전송 에러로 인해 목적지에서 연속적인 재생을 보장받지 못한다. 이러한 문제점을 해결하기 위해 목적지에서는 제한된 버퍼와 재생 보정 알고리즘이 필요하다.

(그림 5)는 생성지에서 생성된 미디어 단위들이 목적지로 도착될 때, 도착 시간을 검사하는 알고리즘이다. 즉, 목적지에 도착한 미디어 단위의 도착 시간이 허용될 수 있는 범위로 도착했는가를 검사하여 허용 범위 내로 도착한 미디어 단위들만 버퍼에 저장시킨다. 허용될 수 있는 범위를 넘어 너무 늦게 도착한 미디어 단위나 너무 빠르게 도착한 미디어 단위는 (그림 5)의 알고리즘을 통해 제거된다.

```

while(Has the media unit arrived ?){
    seq_nu=sync_header of media unit;
    if((arrival_time - (playout_period_time*(seq_nu-1))) >
        (Daverage+allow_jitter)
        dropout: /* arrived too late */
    if((arrival_time - (playout_period_time*(seq_nu-1))) <
        (Daverage-allow_jitter)
        dropout: /* arrived too early */
    put media_unit into buffer;
}

```

(그림 5) 허용지터 범위를 검사하는 알고리즘

허용된 범위 내에 도착된 미디어 단위들을 일정한 재생 주기로 재생을 보장해야한다. 이렇게 재생 주기로 재생 시켜 주고, 전송도중 손실(loss)된 미디어 단위를 보정 시켜 주는 알고리즘은 (그림 6)과 같다.

```

while(buffer is not empty){
    usleep(playout_period_time);
    call playout_correction();
}

playout_correction( ){
    get media_unit from buffer;
    seq_nu=sync_header of media unit;
    if((the number of t by present +1)!=seq_nu) /* normal*/
        playout media_unit;
    if((the number of t by present +1)!=seq_nu)
        /* when the media unit is loss */
        pausing((seq_nu-(the number of t by present))*t);
    playout media_unit;
}

```

(그림 6) 재생 보정 알고리즘

(그림 6)의 알고리즘은 먼저, 허용될 수 있는 범위로 도착된 미디어 단위들을 버퍼에 저장시키는 (그림 5)의 알고리즘을 통해 일정한 재생 주기 시간으로 재생을 보장해 주는 알고리즘이다. 이 알고리즘에서는 일정한 재생 주기 시간을 보장하기 위해 `usleep()`이라는 함수를 사용하였고, 전송 도중 미디어 단위 손실 검사를 위해 동기화 표시기로 사용되었던 일련 번호와 현재 재생 되어야할 재생 주기와 비교하여 미디어 단위 손실 여부를 검사한다. 즉, 동기화 헤더로 사용된 일련 번호와 현재 재생되어야 할 재생 주기와 비교하여 같을 때 정상적인 경우가 되므로 재생 시켜 주면 된다. 만약, 동기화 헤더로 사용된 일련 번호와 현재 재생되어야 할 재생 주기와 다를 경우 전송 도중 미디어 단위가 손실되었음을 의미한다.

예를들면, 생성지에서 생성되어 ATM 네트워크를 통해 목적지로 전송된 미디어 단위가 동기화 헤더로 사용된 일련 번호가 4이고, 현재 재생되어야 할 재생 주기가 3일 경우 동기화 헤더 정보가 현재 재생되어야 할 재생 주기와 다를 경우이다. 따라서 이 경우에 3번째 미디어 단위는 전송 도중 손실을 의미하며 재생 보정 알고리즘은 손실된 미디어 단위를 잠시대기 (pausing)시킨다. 여기에서 잠시대기시킨다는 의미는 바로 앞에 재생되었던 미디어 단위를 손실된 미디어 단위 주기에 복사(duplication)시켜 재생시키거나, 널 (null) 미디어 단위를 재생 시킨다는 의미이다.

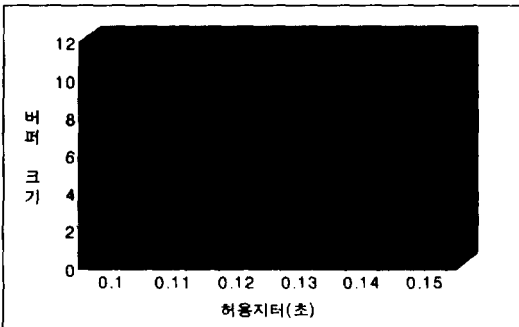
5. 성능평가

본 장에서는 논문에서 제안하고 있는 버퍼크기와 이를 기반한 재생 보정 알고리즘을 성능을 평가한다. 성능 평가 도구로는 SIMAN(SIMulation ANalysys)을 이용하였다. SIMAN은 이산적 모델과 연속적인 모델을 지원하는 범용 시뮬레이션 언어이다[14].

논문의 성능 평가 환경은 다음과 같다. 생성지에서 생성되는 미디어 단위들의 개수들은 각각 1000개, 5000개, 10000개로 하였으며, 생성주기는 비디오 미디어의 생성주기(1/30초)를 기본으로 하였다. 또한, 미디어 단위들이 네트워크를 통해 전송될 때 걸리는 네트워크 전송 지연시간은 Furht 논문에 근간하여 평균 전송 지연시간을 0.25초로 하며, 최대 전송 지연시간 0.4초, 최소 전송 지연시간은 0.1초로 하였다. 그리고 본 논문에서는 ATM 네트워크의 특성을 기반으로 했기 때문에 전송도중 미디어 손실율은 아주 작은 값이며, 본 논문에서는 손실율을 0.1%로 하였다[3].

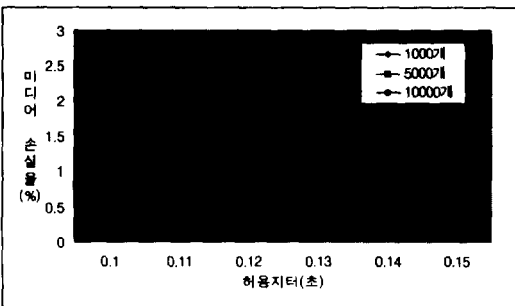
5.1 평가 결과

(그림 7)은 평균 전송 지연시간에서 허용지터 범위를 변화시킴에 따른 제안한 목적지에서의 버퍼 크기 변화를 나타낸다. 본 논문이 제안하고 평균 네트워크 전송 지연시간 0.25초에서 허용지터가 0.1초일 경우 목적지에서 가질 수 있는 최대의 버퍼크기는 8개임을 나타내고 있다. 한정된 네트워크 전송 지연시간에서 최대 허용지터를 가질 경우(0.15초) 최대의 버퍼크기는 12개임을 알수 있다. 또한 (그림 7)은 허용지터 범위가 증가함에 따라 본 논문이 제안하고 있는 버퍼의 크기도 증가됨을 나타내고 있다.



(그림 7) 허용 지터 변화에 따른 버퍼 크기 변화량

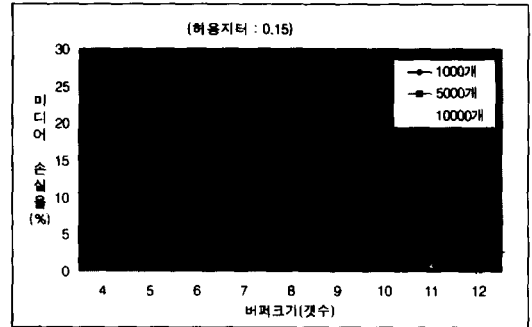
(그림 8)은 허용지터의 변화에 따른 미디어 단위들의 손실율을 나타낸다. 허용지터 범위가 0.1초일 경우 (그림 8)에 나타나고 있듯이 목적지에서 버퍼크기 8개 이고 이때 미디어 손실율은 2.7%~2.8% 사이를 나타내고 있다. 즉 평균전송 지연시간 0.25초에서 허용지터 범위가 0.1초일 경우 2.7%~2.8%에 해당하는 미디어 단위는 허용지터를 넘어서 도착된 경우나, 전송도중 손실된 미디어 단위 개수임을 나타내고 있다. 허용 지터의 범위가 최대일 경우(0.15초) 0.1%의 최소의 미디어 단위 손실율을 나타내고 있다. 손실된 0.1%의 미디어 단위는 본논문이 임의적으로 전송 도중 손실시킨 미디어 단위이다.



(그림 8) 허용지터 범위를 변화시킬 때 미디어 손실율

(그림 9)는 허용지터 범위를 일정하게 고정시킨 후 (0.15초) 버퍼크기 변화에 따른 미디어 손실율을 나타낸다. 본 논문에서는 허용지터가 0.15일 경우 12개의 버퍼 크기를 제안했는데, 12개 미만으로 목적지에서 버퍼크기를 정할 경우 미디어 손실율은 증가하고 있음을 나타내고 있다. 또한 목적지에서 제안한 버퍼크기

이상으로 할당할 경우에는 목적지의 자원낭비를 초래한다.



(그림 9) 허용지터가 일정할 때 버퍼크기에 따른 미디어 손실율

6. 결 론

멀티미디어 통신은 다양한 멀티미디어 응용 서비스를 제공하기 위해 기존의 데이터 통신과 비교하여 다양한 요구 사항을 제공할 수 있어야 한다. 특히, 단일 미디어 내에서 시간적 관계를 유지시키는 미디어내 동기화는 중요한 사항이다.

본 논문은 생성지에서 생성된 미디어가 네트워크 전송 지연으로 인해 발생하는 미디어의 불일치를 해결하기 위해 미디어내 동기화 메커니즘을 제안하였다. 본 논문에서는 미디어내 동기화 메커니즘을 클라이언트에서 수행함으로써 순방향 에러처리를 보장할 수 있으며, 특히 분산 환경에서 사용자의 QoS를 고려한 최적의 버퍼크기를 제안하였다. 또한, 제안된 버퍼크기에 기반하여 전송 에러를 보정하는 재생 보정 알고리즘을 설계하였다. 성능 평가한 결과 사용자가 제공하는 QoS는 목적지에서의 버퍼크기를 정하는데 중요한 요소가 됨을 알 수 있다.

향후 연구로는 미디어내 동기화를 기반으로 미디어간 시간적 관계를 유지하는 미디어간 동기화 메커니즘에 대한 연구가 진행되어야 할 것이다.

참 고 문 헌

- [1] J. F. K. Buford, Multimedia Systems, Addison-Wesley, 1994
- [2] D. Shepherd, M. Salmony, "Extending OSI to

support synchronization required by multimedia applications," Computer Communication, Vol.13, No.7, Sept. 1990.

- [3] B. Furht, "Multimedia Systems : An Overview," IEEE Multimedia, 1994.
- [4] W. Liao, V. O.K. Li, "Synchronization of distributed multimedia systems with user interaction," Multimedia Systems, 1998.
- [5] T. D. C. Little and A. Ghafoor, "Network Considerations for Distributed Multimedia Object Composition and Communication," IEEE Network magazine, Nov. 1990.
- [6] D. Kohler, H. Muller, "Multimedia Playout Synchronization Using Buffer Level Control," 2nd International Workshop on Advanced Teleservice & Highspeed communication Architecture. 1994.
- [7] K. Rothermel, T. Helbig, "An adaptive protocol for synchronizing media streams," Multimedia System, 1997.
- [8] E. Biersack, W. Geyer, C. Bernhardt, "Intra- and Inter-Stream Synchronization for Stored Multimedia System," Proceedings of MULTIMEDIA, 1996.
- [9] Shin T. LIANG, Po L. TIEN and M. C. YUANG, "Threshold-Based Intra-Video Synchronizaton for Multimedia Communications," IEICF Transmission. Communications, Apr. 1998.
- [10] Yong X. Changdong L. Myung J. L and Tarek N. S. "Adaptive Multimedia Synchronization in a Teleconference System," Proc. IEEE ICC, 1996
- [12] R. O. Onvural, Asynchronous Transfer Mode Networks : Performance Issues, Artech House Inc., 1994
- [13] S. Ramanathan and P. Venkat Rangan "Feedback Techniques for Intramedia Continuity and Intermedia Synchronization in Distributed Multimedia Systems," The Computer Journal, Vol.36,

No.1, 1993

- [14] Pegden, C. D. et al, Introduction to Simulation Using SIMAN, McGraw-Hill, 1990



박 홍 진

e-mail : hjpark@sslslab.cse.cau.ac.kr
 1993년 원광대학교 컴퓨터공학과 졸업(공학사)
 1995년 중앙대학교 대학원 컴퓨터공학과(공학석사)
 1997년~현재 중앙대학교 대학원 컴퓨터공학과 박사과정
 관심분야 : 멀티미디어 동기화, 분산 시스템, Web OS



이 준 연

e-mail : swljy@sslslab.cse.cau.ac.kr
 1990년 중앙대학교 전산과 졸업(학사)
 1992년 중앙대학교 대학원 전산과(공학학사)
 1994년~1995년 (주)마이크로소프트
 1995년~현재 중앙대학교 대학원 컴퓨터공학과 박사과정
 관심분야 : 동기화, 부하 균등화, 분산 시스템, 운영체제



김 창 민

e-mail : kimcm@hana.sungkyul.ac.kr
 1977년 서강대학교 수학과 졸업(학사)
 1985년 중앙대학교 대학원 전산과(공학석사)
 1991년 중앙대학교 대학원 전산과(공학박사)
 1989년~1994년 관동대학교 전산과 조교수
 1994년~현재 성결대학교 컴퓨터학부 조교수
 관심분야 : 동기화, 부하분산, 분산 시스템, 운영체제