
QoS를 이용한 동적 멀티미디어 전송 및 프리젠테이션 동기화 기법

나인호*, 양해권*, 고남영**

A Dynamic Synchronization Method for Multimedia Delivery and Presentation based on QoS

In-Ho Ra*, Hae-Kweon Yang*, Nam-Young Ko**

이 논문은 한국학술진흥재단의 '96신진교수연구과제에 대한 연구비 지원에 의해 이루어졌음

요 약

멀티미디어 동기화는 분산 멀티미디어 환경에서 주어진 시간내에 멀티미디어 데이터를 네트워크를 통해 동기식으로 전송하여 연속적인 멀티미디어 프리젠테이션을 보장하는데 필요한 기법이다. 본 논문은 멀티미디어 응용의 서비스 품질 및 특성을 이용하여 지연에 민감한 멀티미디어 데이터를 서비스 품질의 저하없이 동기식으로 전송 및 출력할 수 있도록 하는 미디어 간 및 미디어 내 동기화 기법을 제안한다. 제안된 기법에서는 네트워크 지연 변동으로 인하여 미디어 데이터의 전송 시간을 예측할 수 없고 지역 및 원격 시스템 사이에 동기화를 위한 전역 클럭을 이용할 수 없는 환경에서 스큐와 지터로 인한 비동기 현상을 방지하기 위해 논리 시간 시스템, 동적 버퍼 관리 기법, 동기화 구간의 조정 기법을 적용하였다. 특히, 최대 전송 지연 허용 시간, 최대 패킷 손실률 등과 같은 프리젠테이션 품질을 이용하여 네트워크 지연 변동을 흡수할 수 있고 동기화에 필요한 버퍼 요구량을 줄일 수 있는 방안을 제시하였다.

Abstract

Method for synchronizing multimedia data is needed to support continuous transmission of multimedia data through a network in a bounded time and it also required for supporting continuous presentation of multimedia data with the required nominal playout rate in distributed network environments. This paper describes a new

* 군산대학교 정보통신공학과

** 군산대학교 전파공학과

접수일자 : 1997년 9월 9일

synchronization method for supporting delay-sensitive multimedia presentation without degradation of quality of services of multimedia application. It mainly aims to support both intramedia and intermedia synchronization by absorbing network variations which may cause skew or jitter. In order to remove asynchronization problems, we make use of logical time system, dynamic buffer control method, and adjusting synchronization intervals based on the quality of services of a multimedia. It might be more suitable for working on distributed multimedia systems where the network delay variation is changed from time to time and no global clock is supported. And it also can effectively reduce the amount of buffer requirements needed for transferring multimedia data between source and destination system by adjusting synchronization intervals with acceptable packet delay limits and packet loss rates.

1. 서 론

최근들어 하드웨어 및 소프트웨어 기술이 현격히 발전함에 따라 네트워킹과 다운사이징 전략을 기반으로 한 저가의 고성능 워크스테이션이 널리 보급되고 있으며 사용자가 취급하는 정보의 형태도 기존의 텍스트 문서 위주에서 시청각 형태의 멀티미디어 정보로 다변화되고 있는 추세에 있다. 또한, 고해상도 디스플레이 장치, 사운드 카드, CD-ROM과 같은 대용량 저장 장치 등이 보편화되고 이미지, 비디오 또는 오디오 등과 같은 미디어들이 서로 유기적으로 결합된 멀티미디어 응용 프로그램을 이용하면서 다양한 미디어 데이터들을 통합적이고 유기적으로 처리하여 사용자에게 여러 가지 형태로 전달할 수 있는 멀티미디어 시스템이 요구되고 있다.

멀티미디어 시스템은 미디어 데이터들이 그것들 간의 시공간적인 관계에 따라 동기화되어 프리젠테이션되도록 시스템 내부에 이를 지원하는 동기화 메카니즘을 필수적으로 제공하여야 한다. 그러나, 기존의 동기화 기법들은 특정한 시스템 환경에서 주문형 비디오나 화상회의와 같이 특정한 동기화 조건을 요구로 하는 응용 프로그램의 프리젠테이션 동기화에 적합하도록 설계되어 있기 때문에 일반적인 시스템 환경에서 다양한 멀티미디어 응용 프로그램의 전송 및 수행에는 효과적으로 이용할 수 없는 문제점을 가지고 있다[1,2,3].

본 논문은 이러한 문제를 해결하기 위해 멀티미디어 전송 및 프리젠테이션을 위한 새로운 동기화 기법을 제시하고자 한다. 제안된 기법은 원격 멀티

미디어 시스템으로부터 네트워크를 통해 전송된 멀티미디어 데이터 스트림을 저장하는 버퍼링 부분과 동기화를 위해 필요한 시간 정보를 생성하고 동기화 구간 조정 과 멀티미디어 프리젠테이션 특성을 이용하여 미디어 내(intramedia) 및 미디어 간(intermedia) 동기화를 수행하는 부분으로 구성된다. 이러한 부분들을 통해 멀티미디어 데이터의 검색, 전송, 버퍼링, 프리젠테이션에 관련된 스케줄링을 수행하고 이것들을 이용하여 동기화를 수행함으로써 프리젠테이션에 요구된 서비스 품질을 만족시킬 수 있도록 한다. 즉, 본 논문에서 제안한 기법은 멀티미디어 응용의 서비스 품질의 특성을 이용한 동기화 기법으로서 네트워크의 전송 특성과 미디어 전송에 관련된 서비스 특성을 고려하여 멀티미디어 데이터의 서비스 품질이 저하되지 않도록 하면서 전송이 지연된 미디어 데이터를 동적으로 처리하는 기법이다.

본 논문에서는 네트워크의 트래픽 상태에 따라 동적으로 변하는 미디어 데이터의 실제 지연 시간을 예측한 지연 시간 가상치를 이용하여 미디어 데이터를 전송한 뒤, 이를 버퍼에 저장하고 요구된 동기화 조건에 따라 미디어 데이터 간의 시간 관계를 유지하면서 처리하는 방법을 사용한다. 또한, 네트워크 트래픽 상태와 미디어 데이터의 전송 시간에 대한 정보를 계속적으로 생성 및 분석함으로써 프리젠테이션 시점에 요구된 미디어 데이터의 전송 지연으로 인하여 발생하는 비동기화 현상을 감시하도록 한다. 감시 결과, 응용에서 허용하는 네트워크 최대 전송 지연, QoS 저하 허용률 및 에

러울을 초과하면, 이를 해당 미디어의 서비스 특성에 따라 처리하는 적응형 동기화 기법을 통해 급격한 서비스 품질의 저하를 방지할 수 있는 멀티미디어 프리젠테이션을 수행할 수 있도록 하였다.

또한, 제안된 동기화 기법은 전송 및 수신 멀티미디어 시스템 간의 전역 클럭 부재로 발생하는 클럭 속도 차이에 따른 비동기화 문제를 처리할 수 있도록 설계되었다. 이를 위해 프리젠테이션 장치의 입출력 속도 조절, 버퍼링 조건의 동적 조정, 순간적인 불연속 허용률 그리고 데이터 손실 허용률을 기반으로 한 미디어 데이터의 동기화 구간 조정 기법을 통해 미디어 데이터의 전송 지연에 따른 비동기화 문제를 해결할 수 있도록 하였다.

II. 제안된 멀티미디어 동기화 기법

1. 네트워크 지연 변동 처리 방안

본 논문에서는 네트워크 지연시간의 변동으로 인하여 발생하는 비동기 문제를 해결하기 위한 방안으로 다음과 같은 방법들을 적용하였다.

첫째, 멀티미디어 데이터를 주고 받는 송신 및 수신 멀티미디어 시스템에서 사용하는 클럭은 서로 다를 수 있다. 멀티미디어 데이터를 주고 받는 타이밍을 정확히 맞추기 위해서는 데이터를 전송하기 이전에 이들 간의 지역 시스템 클럭을 동기화한다.

둘째, 전송 미디어 데이터는 출력되기 이전에 버퍼에 먼저 저장된다. 이 때, 버퍼의 크기는 고정되어 있으므로 늦게 도착하는 패킷이나 일찍 도착하는 패킷으로 인하여 버퍼 언더플로우나 오버플로우 현상이 발생할 수 있다. 이러한 문제점을 해결하기 위해 동적 버퍼링 조건에 따라 전송된 패킷의 저장 허용 및 기각을 결정한다. 또한, 버퍼에 도착하는 패킷들의 상태를 감시하여 동기화에 필요한 정보를 생성한다.

셋째, 정적 동기화 단계에서 계산한 각 미디어의 전송 시간이 네트워크의 실질 지연 시간을 반영하지 못하면 버퍼에 도착하는 패킷들 간의 간격 차이가 임계치를 벗어나 버퍼링과 프리젠테이션에 영향을 미친다. 이러한 경우, 피드백 정보를 전송하여 새로운 전송 지연 값을 설정하고 수신측의

검색 및 전송 시간 스케줄을 재구성하도록 한다.

넷째, 어떤 미디어 데이터의 패킷이 네트워크 트래픽 상태의 변화에 의해 지연되어 도착하거나 재전송되어야 하는 경우가 발생한다. 이 때, 이러한 데이터 스트림과 동시에 프리젠테이션되어야 할 다른 미디어 데이터와의 시간 간격이 발생하여 동기화 조건이 위배되는 비동기화 현상이 발생하므로 이러한 문제점을 처리하는 동기화 작업이 수행되어야 한다.

또한, 다양한 네트워크 및 멀티미디어 시스템 환경에서 여러가지 종류의 멀티미디어 응용에 적합한 동기화 서비스를 제공하기 위해 다음과 같은 설계 목표를 설정한다.

첫째, 수신자에 의한 동적 적응형 동기화 기법: 지연 시간이 유동적이고, 송수신자 간의 클럭 차이가 있을 수 있는 네트워크와 시스템 환경에서 지연 시간에 민감한 멀티미디어 응용에 대한 동기화된 프리젠테이션을 수행하기 위해 수신자가 현재의 동기화 상태에 가장 적합한 동기화 조건을 동적으로 조정할 수 있도록 한다.

둘째, 논리 시간 시스템에 의한 동기화 수행: 송수신자 간에는 클럭 속도 차이가 발생할 수 있으며, 수신측 프리젠테이션 장치들 간에도 이것들의 고유한 물리적인 특성으로 인하여 클럭 차이가 존재할 수 있다. 전역 클럭을 사용하여 이러한 문제를 해결하는데에는 한계가 있으므로 각각의 프리젠테이션 장치에 서로 다른 논리 시간 장치를 두고 프리젠테이션 속도 및 동기화 수행 정도에 따라 논리 시간 장치의 진행 속도를 변경함으로써 클럭 차이로 인하여 발생하는 문제를 해결한다.

셋째, 최대 지터 허용 시간을 이용한 동기화 구간 조정: 오디오나 비디오 같은 연속 미디어 데이터 스트림은 어느 정도의 불연속적인 프리젠테이션을 허용한다. 예를 들어, 음성인 경우 발음 구간 내에서 하나의 음소와 그 다음 음소의 출력 간격이 10ms 이하이면 사람은 두 개의 음소가 연속적으로 발음되는 것으로 인식한다. 이와 같이 음성 미디어의 최대 지터 허용 시간을 이용하면 미디어 내의 동기화 구간을 어느 정도 확장할 수 있으므로 확장된 구간 내에서 불규칙적으로 도착하는 패킷들을 서로 동기화하여 서비스 품질을 저하시키

지 않고서도 최적의 동기화를 수행할 수 있도록 한다.

넷째, 패킷 도착을 감시에 의한 지연 시간 조정 및 전송 시간 재스케줄링: 정적 스케줄링 단계에서 멀티미디어 데이터 스트림의 검색 및 전송을 위해 설정된 네트워크 지연 시간은 실제적인 지연 시간과 서로 다를 수 있다. 즉, 네트워크 지연 시간 변동폭에 따라 실제의 지연 시간과 설정된 지연 시간 사이에 시간 간격이 존재하게 된다. 따라서, 실제적인 네트워크 전송 시간을 계속적으로 반영하기 위해서는 버퍼에 도착하는 패킷들의 도착률(interarrival packet rate)을 감시하여 패킷들의 도착 간격을 분석함으로써 지연 시간 변동을 전송 시간 스케줄에 반영하여 항상 최적의 전송이 이루어지도록 한다. 또한, 동기화에 영향을 미칠 정도로 심각한 지연 시간 변동이 발생하면 전송 시간 스케줄을 재작성하도록 수신자가 송신자에게 요청할 수 있도록 한다.

다섯째, 처리율 조절에 의한 지연 시간 흡수: 멀티미디어 응용이 허용하는 한계 내에서 프리젠테이션 장치로 출력되는 미디어 데이터 스트림의 출력 속도를 일시적으로 증가 또는 감소시킬 수 있도록 한다. 이러한 방법을 통해 프리젠테이션 품질에는 별다른 영향을 주지 않으면서 일시적인 미세한 지연 시간 초과를 어느 정도 흡수할 수 있다. 즉, 동기화 구간 내에서의 패킷 처리율을 조정하면 일시적인 전송 지연을 흡수할 수 있고, 이로 인하여 네트워크 지연 시간 조정 및 전송 시간 재작성에 따른 오버헤드를 최소화할 수 있도록 한다.

여섯째, 미디어 간 동기화는 미디어 내 동기화 시점과 밀접한 관계를 가지고 있다. 따라서, 미디어 내 동기화 조건을 충족시키면서 미디어 간 동기화가 수행되도록 동기화 구간을 설정 및 조절함으로써 전체적으로 자연스러운 동기화가 이루어지도록 한다. 이를 위해 미디어 간 동기화의 기준이 되는 주(master) 미디어 데이터를 설정하고 다른 것들은 종(slave) 미디어 데이터로 지정한 다음, 주 미디어 데이터 내의 동기화 구간이 변경되더라도 동적으로 주종 미디어 데이터 간의 상대적 동기화 구간 조절비에 따라 미디어 데이터들이 서로 동기화될 수 있도록 한다.

2. 논리 시간 시스템

멀티미디어 프리젠테이션에 참여하는 각각의 멀티미디어 시스템들은 모두 자신의 고유한 지역 클럭을 가지고 있다. 따라서, 서로 다른 클럭 주기를 가지고 있는 멀티미디어 시스템에서 생성 및 전송되는 미디어 데이터 스트림 간의 시간 관계를 일관적으로 유지하기 위해서는 하나의 공통적인 시간 시스템이 요구된다[4]. 예를 들어, 서로 다른 시간 시스템을 가진 송수신 시스템에서 비디오 데이터 스트림과 오디오 데이터 스트림을 서로 다른 채널을 통해 전송하여 프리젠테이션하여야 하는 상황에서 발생하는 문제점들은 다음과 같은 것들이 있다.

첫째, 클럭 속도 차이에 의해 미디어 데이터의 전송 개시 시점 및 프리젠테이션 시간이 서로 다를 수 있으므로 일정한 시간이 경과되면 두 데이터 스트림 간에 프리젠테이션 스큐가 발생할 수 있다.

둘째, 오디오 및 비디오 프리젠테이션 장치는 각각의 오실레이터에 의한 고유의 클럭을 사용하므로, 두 장치의 발진 주기가 정확히 일치하지 않으면 일정 시간이 경과된 후 오디오 데이터와 비디오 데이터가 동기화되지 않는다.

셋째, 멀티미디어 시스템에서는 시간에 의해 미디어 데이터의 검색, 전송, 그리고 프리젠테이션 시점을 결정하기 때문에 미디어 데이터 내부에는 이러한 시간 정보가 포함되어 있어야 한다. 이 때 사용되는 시간 정보의 형태는 멀티미디어 시스템에 따라 서로 다르며, 대개 타임 스탬프나 미디어의 시작 시점과 이것을 기준으로 한 오프셋(offset) 또는 순차 번호(sequence number)를 사용한다. 전자는 주로 멀티미디어 시스템의 클럭 속도가 서로 동일한 경우에 사용되며, 절대적인 방식으로 시간을 표현하므로 시간 정보의 표현 방법은 간단하지만 서로 다른 클럭 속도를 가진 멀티미디어 시스템 환경에서는 사용될 수 없다. 후자는 서로 다른 클럭 속도를 가진 시스템에서 시간 정보의 표현을 위해 사용되며, 상대적인 시간 표현 방식을 사용한다. 즉, 송신 멀티미디어 시스템에서는 연속적인 미디어 데이터 스트림들의 시간 정보를 순차 번호 및 오프셋의 형태로 전송하고 수신 멀티미디어 시스

템에서는 이것을 절대적인 시간으로 변환하여 사용한다. 이 때에도 전송이나 프리젠테이션 개시와 같이 서로 약속된 시간에 준비된 동작을 수행하기 위해 하나의 일치된 클럭이 때때로 필요하며, 이러한 클럭 정보는 접속 설정 시간이나 전송 도중에 피드백 정보를 통해 전달되어 클럭을 동기화하는데 사용된다. 이 방법의 단점은 동기화된 클럭을 유지하기 위해 주기적으로 클럭 정보를 주고 받아야 한다는 것이며, 주고 받은 클럭 정보가 실제의 네트워크 지연 변동을 정확히 반영하지 못하는 경우가 발생하는 문제점을 가지고 있다[5,6].

이와 같이 하나의 공통된 물리적 클럭을 이용하여 동기화된 프리젠테이션을 수행하는데에는 여러 가지 문제가 발생하므로 본 논문에서는 논리적인 시간에 따라 동기화된 프리젠테이션을 수행하는 방법으로서 동기화 구간을 기반으로 한 논리 시간 시스템(LTS: Logical Time System) 기법을 제안한다. 제안된 LTS는 다음과 같은 특징에 따라 운영된다.

- 서로 다른 송수신자 간의 클럭 동기화를 위한 정보는 네트워크 연결이 완료되어 최초의 데이터 전송이 개시되기 이전에 한 번만 교환된다.
- 수신 멀티미디어 시스템의 클럭은 물리적인 클럭이 아닌 논리적인 클럭을 사용하며, 논리적인 클럭의 클럭 속도는 패킷의 프리젠테이션 단위로 진행된다. 즉, 패킷이 프리젠테이션 될 때마다 이에 해당하는 논리 클럭의 시간이 진행된다. 따라서, 해당 미디어 프리젠테이션 장치로 출력되는 패킷 처리 속도에 따라 논리 클럭의 주기가 결정된다.
- 각 프리젠테이션 장치마다 이에 해당하는 논리 시간 장치를 하나씩 할당한다.
- 서로 다른 프리젠테이션 속도를 가진 미디어 데이터들은 그것들의 상대적인 논리 클럭 속도율에 따라 주어진 동기화 시점에서 프리젠테이션되도록 하여 미디어 데이터 간 동기화를 수행한다.
- 논리 클럭은 미디어 간 동기화를 수행하기 위해 일시 정지되거나 클럭 주기가 변경될 수 있도록 한다.
- 미디어 데이터의 시간 표현 방법으로 전송 개

시 및 프리젠테이션 시점을 기준으로 한 읍셋 방법을 사용한다.

예를 들어, 서로 다른 프리젠테이션 속도를 가진 두 개의 연속 미디어 데이터 스트림 A와 B에 대해 논리 시간 시스템을 이용한 동기화 방법은 다음과 같다. 먼저, 미디어 데이터 스트림 A와 B의 각 패킷들이 프리젠테이션되는 시간 간격을 각각 T_A 와 T_B 라 하자. 또한, A와 B에 할당된 논리 클럭 장치를 각각 LTU_A 와 LTU_B 라 하고, T_A 와 T_B 의 상대적인 시간 간격비를 R_{AB} 라 하면 R_{AB} 는 식(1)에 의해 구해진다.

$$T_A \leq T_B \text{ 일 경우, } R_{AB} = T_A / T_B \dots\dots\dots(1)$$

$$\text{그 밖의 경우, } R_{AB} = T_B / T_A$$

미디어 스트림 A와 B간의 상대적인 프리젠테이션 속도가 식 (1)에 의해 결정되면, LTU_A 와 LTU_B 는 A와 B의 패킷들이 정상적으로 동기화되어 프리젠테이션될 때마다 각각 식 (2)와 식 (3)에 의해 계산된 값만큼 증분된다.

$$LTU_A = \frac{T_B}{R_{AB}} \dots\dots\dots(2)$$

$$LTU_B = R_{AB} * T_A \dots\dots\dots(3)$$

만약, 동기화를 위해 미디어 스트림 A와 B의 프리젠테이션 시간 간격 T_A 와 T_B 가 어떤 동기화 구간 내에서 일시적으로 변경되면 A와 B에 대한 상대적인 시간 간격비 R_{AB} 도 자동적으로 변경된다. 또한, R_{AB} 가 변경되면 LTU_A 와 LTU_B 의 진행 속도 역시 이에 따라 변경되기 때문에 항상 정확한 프리젠테이션 동기화 시점을 결정할 수 있다.

3. 버퍼링 조건 및 버퍼 관리

송신 멀티미디어 시스템에서 패킷 단위로 전송된 미디어 데이터는 지정된 버퍼에 순차적으로 저장된다. 이 때, 버퍼의 크기는 고정되어 있으므로 정적 동기화 단계에서 측정된 네트워크 지연 시간과 실제 지연 시간에 차이가 발생하면 버퍼 오버

플로우 또는 언더플로우가 발생한다. 버퍼 오버플로우를 방지하기 위해 네트워크의 최대 지연 시간을 수용할 수 있는 만큼의 버퍼 공간을 할당할 수도 있으나, 이 방법은 최대 지연 시간을 측정할 수 없는 네트워크 환경에서는 적용될 수 없으며, 버퍼링에 요구되는 버퍼 용량 또한 심각한 정도로 증대되어 동기화 수행으로 인한 시스템 자원의 활용도가 저하된다.

본 논문에서는 버퍼 언더플로우나 오버플로우를 방지하면서 최소한의 버퍼 공간만으로 최적의 버퍼 레벨을 유지하기 위해 다음과 같은 방안을 제시한다.

첫째, 전송된 패킷의 저장 여부를 결정하는 버퍼링 조건을 결정하고 이를 기반으로 전송된 패킷의 저장 허용, 스킵, 그리고 재전송 등의 기법을 사용하여 항상 일정 수준의 버퍼 레벨을 유지하도록 한다.

둘째, 버퍼로 전송되는 패킷의 도착률, 프리젠테이션 장치로 출력되기 위해 버퍼에서 제거되는 패킷의 처리율, 그리고 패킷이 버퍼에 도착하여 처리될 때까지 버퍼 잔류시간(existing time)을 측정하여 버퍼의 레벨 변화를 감시함으로써 버퍼링 조건을 동적으로 변화시킨다.

셋째, 멀티미디어 응용에서 허용하는 패킷 손실률에 따라 프리젠테이션 서비스 품질에는 영향을 주지 않는 한도 내에서 전송된 패킷을 제거하는 기법을 사용하여 버퍼의 레벨을 일정 수준으로 유지하도록 한다.

이러한 버퍼 관리 정책을 기반으로 하여 버퍼링 조건을 결정하고 버퍼의 레벨을 유지하기 위해서 본 논문에서는 다음과 같은 사항들을 고려하였다 [1,2,4].

- α : 패킷 도착율(packet arrival rates)
- β : 패킷 처리율(packet playout rates)
- μ : 버퍼 크기(buffer size)
- ω : 패킷 크기(packet size)
- λ : 현재의 버퍼 레벨(buffer level)
- ρ : 버퍼 오버플로우를 방지하기 위한 최대 임계치(upper threshold)
- γ : 버퍼 언더플로우를 방지하기 위한 최소 임계치(lower threshold)

- ϕ : 단위 시간당 패킷 손실 허용률(allowed packet loss rate)
- ϕ_{current} : 현재까지의 누적 패킷 손실률(cumulated packet loss rate)

본 논문에서 제안한 버퍼링 조건 및 관리 기법은 다음과 같다. 버퍼로 전송된 패킷의 수용 여부는 현재의 버퍼 레벨에 따라 결정된다. 가장 이상적인 버퍼 상태는 패킷의 도착율과 처리율이 일치할 때 이루어지는데, 이것은 미디어 데이터의 전송 스케줄링 시간에 예측한 네트워크 지연 시간이 실제의 지연 시간과 거의 일치하여 패킷이 도착하자마자 처리되는 것을 의미한다. 그러나 이러한 현상은 네트워크 지연 시간의 변동으로 인하여 거의 존재하지 않는다. 이미 언급한 바와 같이 어떤 패킷이 자신의 프리젠테이션 시점보다 일찍 도착하면 프리젠테이션될 때까지 버퍼에 머물러 있어야 하므로 버퍼에 존재하는 시간 동안 버퍼를 차지하게 된다. 따라서, 이러한 패킷이 많을수록 버퍼 레벨이 증가하여 버퍼 오버플로우가 발생하게 된다. 또한, 자신의 프리젠테이션 시점보다 늦게 도착한 패킷들은 프리젠테이션에 사용할 수 없기 때문에 버퍼에서 제거되므로 버퍼 언더플로우 현상이 발생할 수 있다.

본 논문에서는 이러한 문제점을 해결하기 위한 방안으로 버퍼 레벨의 최대 및 최소 임계치를 설정하고, 새로 전송된 패킷의 저장으로 인하여 버퍼의 레벨이 최대 임계치를 초과하지 않으면서 프리젠테이션에 사용할 수 없는 패킷이 아니라면 버퍼에 저장하도록 하는 버퍼링 조건을 제안하였다. 또한, 버퍼에 저장할 수 없는 패킷인 경우에는 응용에서 허용하는 패킷 손실률을 초과하지 않는 한도에서 스킵하도록 한다. 이 때 스킵으로 인하여 버퍼 레벨이 최소 임계치 보다 떨어지지 않아야 한다. 만약 위의 두 가지 조건을 위배하는 경우에는 실제의 네트워크 지연 시간 변화를 반영하는 전송 시간 스케줄을 재작성하도록 한다. 이 때 중요한 것은 전송 시간 스케줄 재작성 작업의 수행과 새로운 전송 스케줄을 기반으로 한 패킷 전송으로 인해 서비스 품질의 수준이 멀티미디어 응용에서 요구하는 서비스 품질 저하 허용률보다 떨어져서

는 안된다는 것이다. 이러한 현상의 발생 여부는 전송 시간 스케줄 재작성의 시기와 방법에 따라 결정된다.

따라서 버퍼를 관찰한 결과로 생성되는 네트워크의 실제 전송 지연 시간과 전송 스케줄 작성시에 예측한 전송 지연 시간과의 차이를 기존의 전송 스케줄에 가감하여 새로운 전송 스케줄을 효과적으로 빠른 시간 내에 결정하는 방법을 사용하도록 한다. 또한, 전송 스케줄의 재작성 시기는 버퍼 언더플로우나 오버플로우 임계치에 도달할 때 수행되도록 하여 실제로 버퍼 오버플로우 및 언더플로우 현상이 발생하기 이전에 이러한 문제가 해결될 수 있도록 하였다. 마지막으로, 비디오 데이터와 같이 동기화를 위해 기존의 데이터를 다시 사용하여야 하는 경우, 버퍼에 존재하는 패킷들은 프리젠테이션 수행이 완료된 이후 동기화를 위해 더 이상 필요 없을 때에만 버퍼에서 제거되도록 하였다.

다음의 Buffer_management() 알고리즘은 위에서 기술한 정책을 기반으로 동적 버퍼링 조건 조정 및 버퍼 관리 기법을 나타낸 것이다.

Algorithm Buffer Management(PTT,PPTT)

Input: PTT(Presentation Time Table)

Output: PPTT(Packet Presentation Time Table)

Let $P(i,j)$ be the newly arrived j^{th} packet of media i through a given network;
 Let $PPTT[i][j]$ be the presentation time of $P(i,j)$ for PTT;
 Let $LTU(i)$ be the logical time unit assigned to the presentation device of media i ;
 Let T_c be the current time of $LTU(i)$;
 Let α be the packet arrival rate in a given unit time;
 Let β be the packet playout rate in a given unit time;
 Let ω be a packet size;
 Let n be the number of packets to be presented in a unit time;
 Let ρ_i and γ_i be the upper and lower threshold of the buffer for media i , respectively;
 Let φ_i be the allowed packet loss rates of media i in a given unit time;

Let $\varphi_{current(i)}$ be the cumulated packet loss rate for media i in a given unit time

```

 $\lambda = (\alpha - \beta) * \omega$ ; /* determine the current buffer level */
if( $\omega + \lambda < \rho$  and  $PPTT[i][j] \leq T_c$ ) then {
store  $P(i,j)$  to the corresponding buffer;
 $\lambda = \lambda + \omega$ ;
}
else if( $\varphi_{current(i)} \leq \varphi_i$  and  $\lambda < \gamma$ ) then {
skip  $P(i,j)$ ;
 $\varphi_{current(i)} = \varphi_{current(i)} + 1 / n$ ;
 $\lambda = \lambda + \omega$ ;
}
    
```

4. 미디어 내 동기화

주어진 하나의 미디어 데이터 스트림 내의 모든 패킷들을 일정한 간격으로 연속해서 출력하는 것을 미디어 내 동기화라고 한다. 이것은 서로 다른 지연 시간을 가지고 전송된 패킷들을 일정한 속도 또는 간격으로 프리젠테이션 장치로 출력하는 것과 동일한 의미를 가진다. 일반적으로 미디어 데이터 스트림을 구성하는 패킷들의 버퍼내 도착율과 그것들의 처리율이 동일한 경우에 가장 최상의 미디어 내 동기화를 이룰 수 있다. 이것은 네트워크의 지연 시간이 일정하고 송수신 멀티미디어 시스템 간의 클럭 속도가 동일한 경우에 한하여 동기화가 자연스럽게 이루어 질 수 있으며, 이러한 경우를 정적 동기화라 한다. 그러나 대부분의 경우 네트워크는 가변 지연 시간이라는 특성을 가지고 있고 송수신 멀티미디어 시스템들은 서로 자신의 고유한 지역 클럭을 가지며, 이것들의 속도는 대개 서로 다르기 때문에 미디어 내 동기화가 자연스럽게 수행되는 경우는 거의 없다. 따라서, 대부분의 시스템 환경에서는 미디어 내 동기화 작업이 수행되어야 한다.

미디어 데이터 스트림을 구성하는 패킷들 간에는 동기화된 프리젠테이션을 위한 동기화 구간이 존재한다. 단위 시간마다 정상적으로 프리젠테이션되어야 할 패킷의 개수를 N 이라 할 때, 매 ρ ($1/N$) 시간마다 하나의 패킷이 출력되어야 한

다. 따라서, 각각의 패킷들은 ρ 마다 동기화되어 출력되어야 하므로, 이러한 패킷들 간의 출력 간격을 동기화 구간(synchronization interval)이라 한다. 각 패킷의 동기화 구간의 크기를 Δs 라 할 때, i 번째 패킷이 T_i 라는 시간에 출력되었으면 $i+1$ 번째 패킷은 $T_i + \Delta s$ 시간에 출력되어야 미디어 간 동기화가 이루어질 것이다. 그러나 연속 미디어 데이터는 그 종류에 따라 어느 정도의 불연속을 허용한다. 예를 들어, 음성 미디어인 경우 종단간 최대 지터 지연이 10 ms 이하이면 서비스 품질에 영향을 주지 않고 일시적인 불연속을 허용할 수 있다. 또한, 비디오 데이터인 경우에는 지정된 동기화 구간에 어느 정도 분량의 패킷을 출력하지 못해도 비디오의 서비스 품질에는 큰 영향을 주지 않는다. 이러한 미디어 데이터의 불연속 허용 특성을 동기화 구간에 적용하면 명목상의(nominal) 동기화 구간보다 더 크게 확장할 수 있어 네트워크 지연 시간 변동으로 인하여 늦게 도착한 패킷들도 프리젠테이션할 수 있으므로 지연 시간 변동을 어느 정도 흡수할 수 있다. 이와 같이 멀티미디어 응용에서 허용하는 프리젠테이션 특성에 따라 허용된 QoS 범위 내에서 미디어 간 동기화를 수행함으로써 보다 유연한 동기화 정책을 수행할 수 있다.

지금까지 언급한 동기화 구간 조정 기법을 기반으로 한 미디어 내 동기화 기법을 설명하면 다음과 같다. 프리젠테이션 장치로 출력되어야 할 단위 시간당 패킷의 갯수를 N 라 하면, 단위 시간당 각 패킷들이 $1/N$ 시간 간격으로 출력되어야 하므로, 해당 미디어 데이터 스트림을 구성하는 패킷들 간의 명목상의 동기화 구간 ρ 는 $1/N$ 이 된다. 또한, 해당 미디어 데이터가 허용하는 순간 불연속 허용률을 λ 라 하면 확장된(extended) 동기화 구간 $\rho' = \rho + \lambda$ 가 된다. 따라서, 미디어 데이터 스트림 i 에 대한 j 번째 패킷이 P_{ij} 라는 시간에 프리젠테이션되었다면 $j+1$ 번째 패킷의 프리젠테이션 시점은 $P_{i(j+1)} = P_{ij} + \rho'$ 이 되어야 미디어 내 동기화를 달성할 수 있다. 따라서, $j+1$ 번째 패킷의 버퍼 내 도착 시점을 $A_{i(j+1)}$ 라고 하면 $A_{i(j+1)}$ 이 $P_{i(j+1)}$ 보다 작거나 같아야 동기화 조건을 만족할 수 있다. 즉, 미디어 간 동기화를 위해서 j 번째 패킷과 $j+1$ 번째 패킷은 동기화 구간 내에서 다음의 식 (4)를 만족해야 한다.

$$\begin{aligned}
 A_{ij} &\leq P_{ij+1} \\
 &\leq P_{ij} + \rho' \\
 &\leq P_{ij} + \rho + \lambda \dots\dots\dots(4) \\
 &\leq P_{ij} + 1/N + \lambda
 \end{aligned}$$

위의 동기화 조건을 기반으로 한 미디어 내 동기화 기법을 제안하기 위해 본 논문에서는 다음과 같은 환경을 설정하였다. 먼저, 각 미디어 데이터 스트림에는 동기화를 위한 하나의 논리 시간 장치(LTU)를 할당하여 패킷의 출력 속도에 따른 프리젠테이션 시간 변화를 나타내도록 한다. $LTU(i)$ 의 초기 값은 정적 동기화 단계에서 생성한 미디어 데이터 i 의 프리젠테이션 시작 시간이며, $LTU(i)$ 는 정상적으로 패킷이 프리젠테이션되면 동기화 구간만큼 규칙적으로 증가된다. 따라서, 미디어 데이터 i 의 프리젠테이션 시작 시간을 $T_{start}(i)$ 라 하면 명목상의 동기화 구간에 따라 j 번째 패킷이 프리젠테이션되면 $LTU(i) = T_{start}(i) + \rho * j$ 가 된다.

그러나 네트워크 지연 시간 변동으로 인하여 설정된 초기 지연 시간 보다 더 늦은 지연 시간을 가지고 도착한 패킷으로 인하여 동기화 조건을 항상 만족시킬 수 없게 된다. 따라서, 이러한 경우에는 패킷의 처리 속도 즉, 프리젠테이션 속도를 해당 미디어 데이터에서 허용하는 한도에서 일시적으로 증가시키거나 동기화 조건을 만족할 수 없는 패킷을 스킵함으로써 서비스 품질에는 영향을 미치지 않으면서 미디어 내 동기화를 수행하여야 한다.

만약, j 번째 패킷이 위의 동기화 조건을 만족하지 못할 때에는 일시적인 비동기화 현상이 발생하게 된다. 이러한 경우에는 응용이 허용하는 한도에서 적절한 동기화 절차가 수행되어야 한다. 예를 들어, $j+1$ 번째 패킷에 설정된 지연 시간을 D_{j+1} 라 하고, 이것의 실제 지연을 A_{j+1} 라고 하였을 때, j 번째 패킷이 자신의 선행 패킷인 j 번째 패킷과의 정상 및 확장된 동기화 구간에서 동기화될 수 없을 경우, 스킵 조건을 만족하면 스킵한다. 이 때, 패킷이 스킵되면 이에 따른 논리 시간도 해당 동기화 구간만큼 증가되고 스킵된 패킷은 손실된 패킷으로 간주된다. 단위 시간당 스킵된 패킷의 갯수는 해당 미디어 데이터에서 허용하는 단위 시간당 패킷 손실 한계를 초과해서는 안되며, 연속된 패킷의

스킵을 방지하기 위해 설정된 지연 시간과 실제 지연 시간의 차이를 계산하여 이후의 미디어 데이터 스트림이 제시간에 도착하여 동기화 조건을 만족할 수 있도록 송신자의 전송 스케줄을 재작성하도록 하여야 한다.

미디어 데이터 i 의 단위 시간당 패킷 손실 허용률을 δ 라 하고 현재까지의 프리젠테이션 시간 $LTU_{current}(i)$ 와 단위 시간 T_U 와 상대적인 시간 관계를 τ 라 하면 $\tau = LTU_{current}(i) / T_U$ 가 된다. 따라서, 현재까지 멀티미디어 응용에서 허용 가능한 누적 패킷 손실 한계는 $L_{cumulated}(\tau) = \tau * \delta$ 가 된다. 따라서, 동기화 조건에 위배된 패킷의 스킵 여부는 $L_{cumulated}(\tau)$ 를 허용하지 않는 한도 내에서 이루어져야 한다. 또한, 실질 지연 시간과 설정된 지연 시간과 차이를 전송 스케줄에 반영하기 위해서는 수신자가 새로운 지연 시간 옵션을 설정하여 송신자에게 전송하고, 송신자가 이를 이용하여 새로운 전송 스케줄을 작성하도록 하여야 한다. 이때, 이러한 시간 동안 누적된 패킷 손실률이 허용 한계에 도달하면 패킷을 스킵할 수 없으므로 지연 시간 조정을 위한 임계치를 두고 해당 미디어에서 허용하는 최대 패킷 손실률에 도달하기 전에 먼저 임계치에 도달하면 송수신자 간의 전송 시간 재스케줄링 작업을 수행하도록 한다. 현재의 프리젠테이션 시점에서 미디어 데이터 i 의 패킷이 손실되면 $LTU_{current}(i)$ 의 값은 동기화 구간만큼 증가되어 $LTU_{current}(i) = LTU_{current}(i) + \rho$ 가 된다.

일정한 기간 동안 전송 시간 재스케줄링 작업이 반복되면 네트워크의 지연 시간이 계속 상승된다는 것을 의미한다. 이것은 지연 시간을 재설정하는 것만으로는 지연 시간을 충분히 흡수할 수 없다는 것을 의미하므로 프리젠테이션 장치의 출력 속도를 일시적으로 감소시켜 네트워크 지연 시간을 흡수하여 미디어 내 동기화를 이루어야 하는데, 이런 경우 동기화 구간은 정상의 경우보다 더욱 확장된다. 그러나 이 때에도 동기화 구간의 확장이 응용에서 허용하는 범위를 벗어나지 않도록 한다.

프리젠테이션 장치의 출력 속도가 변화되면 이에 따라 정상적인 동기화 구간의 크기도 변화되어야 한다. 또한, 확장된 동기화 구간에 의해 동기화된 패킷의 수가 많으면 원래의 정상적인 동기화

구간과의 스큐가 발생한다. 즉, 정상적인 프리젠테이션과 동기화된 프리젠테이션 사이에 스큐가 발생한 것이므로 다시 정상적인 프리젠테이션 시간과 동기화를 하여야 하므로 미디어 내 동기화 기법에서 적용된 것과 같이 패킷의 스킵을 통해 재동기화를 수행한다.

확장된 동기화 구간에서의 동기화로 인하여 경우 프리젠테이션 시간에 스큐가 발생하면 다음과 같은 방식으로 처리한다.

첫째, 스큐의 허용 한도는 명목상의 동기화 구간의 크기(즉, 한 패킷의 프리젠테이션 시간 간격)인 ρ 보다 작아야 한다. 이것은 정상적인 동기화 구간에서의 프리젠테이션과 확장된 구간에서의 스큐 처리를 ρ 단위로 수행하는 것을 의미한다.

둘째, 일정한 구간에서 누적된 스큐 정도가 ρ 에 근접하면 다음 패킷을 스킵시킨다. 이 때 해당 패킷의 스킵으로 인하여 단위 시간 동안에 허용된 최대 패킷 손실률을 초과하지 않도록 한다.

셋째, 스큐를 제거하기 위한 시간을 결정하기 위해서는 누적된 스큐의 처리 시점을 나타내기 위한 임계치를 설정하여야 한다. 임계치가 ρ 에 가까울수록 스큐의 정도는 커지지만 스킵되는 패킷의 갯수가 달라지므로 서비스 품질에 영향을 주지 않는 한도에서 임계치를 동적으로 변화시킬 수 있도록 한다.

넷째, 정상적인 동기화 구간과 확장된 동기화 구간의 재동기화를 위해 패킷을 스킵하면 스킵된 패킷의 출력 구간과 누적된 스큐와의 차이가 존재한다. 이러한 차이를 T_{diff} 라 하면, 현재의 논리 시간은 T_{diff} 만큼 증가되어야 한다. 다음 식 (5)는 j 번째 패킷을 스킵하여 스큐를 처리한 결과로 논리 시간을 재조정하여야 하는 경우에 사용된다.

$$\begin{aligned}
 LTU(i)_{current} &= LTU(i)_{current} + D_{diff} \\
 &= LTU(i)_{current} + T_{srart}(i) + \rho * (j) \dots\dots\dots(5) \\
 &- LTU(i)_{current}
 \end{aligned}$$

다음의 `Intramedia_synchronization()` 알고리즘은 본 논문에서 제안한 확장된 동기화 구간 및 패킷 손실률 그리고 프리젠테이션 불연속 허용률을 기반으로 한 미디어 내 동기화 기법을 나타낸 것이다.

Intramedia_Synchronization(PTT,PPTT,P(i,j))

Input: PTT(Presentation Time Table),
 PPTT(Packet Presentation Time Table),
 P(i,j): the j^{th} packet of media i to be presented;

Let PPTT[i][j] be the presentation time of P(i,j) in PTT;
 Let LTU(i) be the logical time unit assigned to the presentation device of media i ;
 Let T_c be the current time of LTU(i);
 Let n be the number of packets of media i presented in a unit time;
 Let θ_i be the playout duration of a packet in media i ;
 Let λ be the allowed maximum jitter rates of media i ;
 Let ρ_i be the nominal synchronization interval of media i ;
 Let γ_i be the extended synchronization interval of media i ;
 Let ψ_i be the allowed maximum playout rate of media i ;
 Let $\psi_{\text{curr}(i)}$ be the current playout rates of media i ;
 Let φ_i be the allowed packet loss rate of media i presented in a unit time;
 Let $\varphi_{\text{cum}(i)}$ be the cumulated packet loss rate of media i for in a given unit time;
 Let $S_{\text{cum}(i)}$ be the cumulated presentation skew of media i for presented in a given unit time;
 Let D_{ij} and A_{ij} be the nominal and actual network transfer delay of the j^{th} packet of media i , respectively;

$\rho_i = 1 / \theta_i$;
 $\gamma_i = \rho_i + \lambda$;
 status = Intermedia_Synchronization(P(i,j));
 if(PPTT[i][j] \leq PPTT[i][j-1] + ρ_i and status == OK) then {
 present P(i,j) to the presentation device of media i ;
 LTU(i) = LTU(i) + θ_i ;
 }
 else if(P[i][j-1] + $\rho_i <$ PPTT[i][j] \leq P[i][j-1] + γ_i and
 $S_{\text{cum}(i)} \leq \rho_i$ and status == OK) then {

present P(i,j) to the presentation device of media i ;
 LTU(i) = LTU(i) + θ_i + A_{ij} - D_{ij} ;
 }
 else {
 if($\varphi_{\text{cum}(i)} < \varphi_i$ and $\psi_{\text{curr}(i)} < \psi_i$) then {
 skip P(i,j);
 $\varphi_{\text{cum}(i)} = \varphi_{\text{cum}(i)} + 1 / n$;
 LTU(i) = LTU(i) + θ_i ;
 $\psi_{\text{curr}(i)} = \psi_{\text{curr}(i)} + (D_{ij} - A_{ij}) / \theta_i$;
 $\theta_i = \theta_i + 1 / n$;
 }
 else reschedule the transfer times of remaining packets in media i ;
 }

5. 미디어 간 동기화

서로 다른 실시간 미디어 스트림 간의 동기화는 이들 간의 상대 처리 속도(relative playout rate) 즉, 단위 시간당 처리되는 패킷 처리율 간의 상대 속도가 프리젠테이션의 모든 동기화 구간에서 일정하게 진행될 때 이루어진다. 어떤 미디어 데이터 A와 B의 패킷들이 각각 단위 시간당 θ_A , θ_B 의 처리 속도로 프리젠테이션되어야 한다고 가정하자. 이 때, 각 미디어 데이터에 대한 동기화 구간의 크기를 ρ_A , ρ_B 라 하면 $\rho_A = 1 / \theta_A$, $\rho_B = 1 / \theta_B$ 가 된다.

미디어 간 동기화는 어떤 미디어 데이터를 기준으로 하여 동기화를 수행하느냐에 따라 결정된다. 동기화의 기준이 되는 미디어를 주(master) 미디어라 하고 주 미디어에 동기화되는 미디어를 종(slave) 미디어라 한다. 주종 미디어의 선택은 멀티미디어 응용의 특성에 따라 결정되는데, 화상 전화인 경우 음성 미디어가 주 미디어로 작용하고 비디오 미디어는 종 미디어 역할을 한다. 왜냐하면 비디오와 음성이 합성된 멀티미디어 응용의 대부분이 입술 동기(lip-synch)에 중점을 두고 있기 때문이다. 예를들어 A를 비디오 미디어, B를 음성 미디어라 가정하면, 음성 미디어의 한 패킷이 출력되는 시간 동안 비디오 미디어는 두 개의 패킷이 동일

한 시간에 출력되어야 미디어 간 동기화를 달성할 수 있다고 가정하자. 이 때, 네트워크 지연 변동을 흡수하기 위한 미디어 내 동기화 작업이 주 미디어 데이터에 수행되면 이로 인하여 주 미디어의 동기화 구간이 정상적인 경우의 동기화 구간과는 다르게 구성될 수 있다. 따라서, 주 미디어의 동기화 구간이 변경되면 이에 따라 종 미디어의 동기화 구간도 변경되어야 한다. 예를 들어, 주 미디어가 정상적인 속도로 일정하게 출력되었을 때에는 동기화 시점에 맞춰 종 미디어의 프리젠테이션 속도를 조절하면 주 미디어와 종 미디어 간의 동기화가 자연스럽게 수행되나, 이러한 경우는 거의 발생하지 않고 대부분 주 미디어의 동기화 구간은 네트워크 지연 변동에 따라 변경된다. 따라서, 주 미디어와 종 미디어 간에는 새로운 동기화 시점이 생성되며, 동기화 시점은 동적으로 변화되기 때문에 미디어 간 동기화를 수행시 이러한 것들을 모두 고려해야만 최적의 동기화를 성취할 수 있다. 본 논문에서 제안하는 미디어 간의 동기화 전략은 다음과 같다.

첫째, 미디어 간 동기화는 하나의 주 미디어의 프리젠테이션 속도를 기준으로 한 종 미디어들의 상대적인 프리젠테이션 속도 조절비를 유지함으로써 수행된다.

둘째, 종 미디어와 주 미디어 간의 동기화 시점은 주 미디어를 구성하는 패킷들의 처리 시작 시점을 나타내는 논리 시간 장치에 표시된 시간을 기준으로 한다.

셋째, 미디어 데이터 간에는 어느 정도의 불연속을 허용하므로 이를 이용하여 유연한 미디어 간 동기화를 수행한다.

넷째, 주 미디어와 종 미디어 간의 동기화를 위해 스킵 또는 중복시키는 방법을 이용한다.

다섯째, 주 미디어와 종 미디어 간의 프리젠테이션 스큐의 정도가 최대 허용치에 도달하면 주 미디어 데이터 스트림에서 허용하는 최대 불연속 허용 한계만큼 주 미디어의 프리젠테이션을 일시 정지한다. 따라서, 이 시간만큼 종 미디어와 주 미디어 간의 프리젠테이션 스큐를 처리하도록 한다. 이 때 프리젠테이션 일시 중지로 인한 공백 시간에는 이에 해당하는 대체 행위를 수행하도록 한다.

여섯째, 주종 미디어 간의 프리젠테이션 스큐가 줄어들지 않으면 스큐의 정도에 따라 종 미디어 프리젠테이션 장치의 프리젠테이션 속도를 조절한다.

일곱째, 미디어 간 동기화를 감시하는 동기화 모니터를 통해 주종 프리젠테이션 장치의 프리젠테이션 상태를 감시하여 미디어 간 동기화를 수행하는데 필요한 동기화 정보를 생성한다.

이러한 정책을 기반으로 미디어 간 동기화를 수행하기 위해서는 먼저, 각 미디어 데이터의 프리젠테이션 속도에 따라 상대 속도 조절비를 계산하여야 한다. 미디어 데이터 스트림 간의 상대 속도 조절비를 이용한 미디어 간 동기화 기법은 Anderson에 의해 제안된 바 있다[4]. 이 기법에서는 하나의 주 미디어 스트림과 여러 개의 종 미디어 스트림 간의 고정적인 상대 속도 조절비를 이용한다. 즉, 주 미디어의 프리젠테이션 속도는 항상 일정하다고 가정함으로써 주 미디어를 기준으로 한 주종 미디어 간의 상대 속도를 이용한 미디어 간 동기화를 수행하도록 제한하고 있다. 그러나 주 미디어 데이터 스트림의 프리젠테이션 속도를 일정하게 고정시키는 것은 현실적으로 불가능하며, 이것은 네트워크 지연 변동 시간을 허용하는 실시간 멀티미디어 전송 시스템 환경에서는 사용할 수 없는 단점을 가진다. 주 미디어 데이터에서 단위 시간당 프리젠테이션되어야 하는 패킷의 수를 N_{master} 라 하고 어떤 종 미디어 데이터에서의 그것을 N_{slave} 라 하자. P_{master} 와 P_{slave} 를 각각 주 및 종 미디어 데이터의 명목상의 동기화 구간 내에서 패킷들의 출력 속도라 하면 $P_{master} = 1 / N_{master}$, $P_{slave} = 1 / N_{slave}$ 이다. 또한, 실시간에 네트워크 지연 시간 변동으로 인해 출력되는 주 및 종 미디어 데이터 패킷들의 실질 출력 속도를 각각 A_{master} , A_{slave} 라 하자. 그러면 미디어 내의 동기화를 위한 최적의 조건은 $P_{master} = A_{master}$, $P_{slave} = A_{slave}$ 가 된다. 여기서 미디어 간 동기화를 위해서 필요한 주종 미디어 간의 명목상의 상대 패킷 처리율 $R_{nominal} = P_{slave} / P_{master}$ 가 되며, 실시간에 발생하는 실제적인 상대 패킷 처리율 $R_{actual} = A_{slave} / A_{master}$ 가 된다. 따라서, 명목 및 실제 동기화 구간 내에서 주종 미디어 간의 동기화는 아래의 식(6)과 (7)을 만족할 때 달성할 수 있다.

$$P_{master} = R_{nominal} * P_{slave} \dots\dots\dots (6)$$

$$A_{master} = R_{actual} * A_{slave} \dots\dots\dots (7)$$

따라서, 실시간에 발생하는 최적의 미디어 간 동기화 조건은 식 (8)과 같다.

$$\frac{R_{actual}}{R_{nominal}} = \frac{\frac{A_{slave}}{A_{master}}}{\frac{P_{slave}}{P_{master}}} = \frac{A_{slave} * P_{master}}{A_{master} * P_{slave}} = 1 \dots\dots (8)$$

그러나 위의 최적 조건은 설정된 네트워크 지연 시간과 실제 네트워크 지연 시간이 일치하지 않는 경우에는 만족될 수 없다. 따라서, 멀티미디어 응용이 허용하는 미디어 간의 순간적인 불연속 허용 범위 내에서 미디어 간의 동기화를 수행하는 유연한 동기화 조건을 설정하여 준최적의 미디어 간 동기화를 수행하도록 한다.

미디어 데이터 A와 B 간의 순간적인 불연속 허용률을 ρ라 할 때, 수정된 준최적의 미디어 간 동기화 조건은 식 (9)와 같다.

$$\frac{R_{actual}}{R_{nominal}} = \frac{\frac{A_{slave}}{A_{master}}}{\frac{P_{slave}}{P_{master}}} = \frac{A_{slave} * P_{master}}{A_{master} * P_{slave}} = 1 \pm \rho \dots\dots (9)$$

네트워크 트래픽 상태는 실시간에 따라 가변적으로 변하기 때문에 네트워크를 통해 전송된 데이터의 도착 시간이 일정 시간 동안 계속 지연되면 지연된 미디어 데이터 스트림과 정상적으로 출력된 미디어 데이터 간에 프리젠테이션 스큐가 발생된다. 이와 같이 스큐가 발생하면 동기화 작업을 수행하는 동안 취해야 할 대체 행위를 결정해야 한다. 지연된 미디어 데이터의 특성에 따라 스큐 처리시 수행해야 할 대체 행위는 다음과 같다.

첫째, 동기화 사건을 기다리는 미디어를 블럭킹하지 않고 출력한다.

둘째, 동기화 사건을 기다리는 미디어 데이터는 블럭된다.

셋째, 동기화 사건을 기다리는 미디어를 기본적으로는 블럭킹하되 블럭킹되는 과정에서 대기 상태에 있는 것이 아니고 기다리는 동안 어떤 행위를 한다.

다음의 Intermedia_synchronization() 알고리즘은 주종 미디어 간의 상대 프리젠테이션 속도, 미디어 간 불연속 허용률, 그리고 스킵 및 중복을 기반으로 한 미디어 간 동기화 기법을 나타낸 것이다.

Intermedia Synchronization(P(i,j))

Input: P(i,j): the jth packet of media i to be presented;
Output: OK, NOK;

```

Let m and i be master and slave media, respectively;
Let LTU(i) be the logical time unit assigned to the presentation device of media i;
Let θi be the nominal synchronization interval of media i;
Let n be the number of packets of media i in a unit time; Let ρ(m,i) be the allowed maximum discontinuity rate between m and i;
Let Pk and Ak be the nominal and actual packet playout rates of media k, respectively;
Let ψi be the allowed maximum playout rate of media i;
Let ψcurr(i) be the current playout rate of media i;
Let φi be the allowed packet loss rate of media i for a given unit time;
Let φcum(i) be the cumulated packet loss rate for media i for a given unit time;
Let Scum(i,j) be the cumulated presentation skew for media i for a given unit time;
if(1-ρ(m,i) ≤ (Ai * Pm) / (Am * Pi) ≤ 1 + ρ(m,i))
then {
return(OK);
}
else {
if(LTU(m) < LTU(i) and φcum(i) < φ(i) and Scum(m,i) < θi) then {
skip P(i,j);
φcum(i) = φcum(i) + 1 / n;
LTU(i) = LTU(i) + 1 / Pi;
}
else {

```

```

if( $S_{cum(m,i)} \geq \theta_i$ ) {
duplicate P(m,j) and playout the substituted operation
    for media i;
}
else {
duplicate P(i,j-1) and playout the
substituted operation for media i;
 $\psi_{curr(i)} = \psi_{curr(i)} + P_i / \theta_i$ ;
 $\theta_i = \theta_i + 1 / n$ ;
}
}
return(NOK);
}
    
```

III. 결 론

본 논문은 동기화 정보와 미디어 데이터의 동기화 구간 조정 및 프리젠테이션 특성을 이용한 동적 적응형 동기화 기법의 설계 방안으로서 전송된 미디어 데이터 스트림이 예정된 도착 시점보다 늦거나 빠르게 도착하더라도 이것들이 동기화에 영향을 주지 않도록 하는 유연한 동기화 기법을 제시하는데 목적을 두고 있다. 이를 위해 미디어 데이터 내의 불연속 허용률을 이용한 확장된 동기화 구간 내에서의 미디어 내 및 미디어 간 동기화 알고리즘을 제시하였다. 또한, 프리젠테이션 장치의 처리 속도 조절과 미디어 데이터 간의 상대적인 프리젠테이션 속도 조절비 및 데이터 손실 허용률을 이용한 미디어 데이터의 스킵 및 중복 기법을 통해 지터나 스큐를 처리하는 동기화 방안을 제시하여 네트워크에서 발생하는 지연 시간 변동을 줄일 수 있는 방안을 제안하였다.

본 논문에서 제안된 동기화 알고리즘은 멀티미디어 운영체제와 실시간 고속 멀티미디어 전송 프로토콜이 지원되면 더욱 효율적인 성능을 제공할 수 있으며, 실용화에 쉽게 접근할 수 있을 것이다. 현재 우리는 SLAM II를 이용하여 본 논문에서 제

안된 알고리즘들의 성능을 평가 및 분석하기 위한 시뮬레이션을 수행하고 있으며 시뮬레이션 결과를 바탕으로 물리적 멀티미디어 프리젠테이션 장치와 논리 시간 시스템과 결합하기 위한 효율적인 장치 인터페이스가 설계할 예정이다. 마지막으로 실시간 환경에서 제안된 동기화 알고리즘의 수행으로 인한 실행 시간을 줄일 수 있도록 제안된 동기화 알고리즘을 펌웨어(firmware)로 구현하기 위한 방안을 연구할 예정이다.

참고문헌

- [1] L. L. A. Karmouch and N. D. Georganas, "Synchronization in Real Time Multimedia Data Delivery," Proc. of ICC '92, 1992.
- [2] S. Ramanathan and P. V. Rangan, "Continuous Media Synchronization in Distributed Multimedia Systems," Proc. of 3rd International Workshop on Network and Operating System Support for Digital Audio and Video, Sandiego, California, pp. 289-296, Nov. 1992.
- [3] A. Jones and A. Hopper, "Handling Audio and Video Streams in a Distributed Environment," Proc. of the 14th ACM Symposium on Operating Systems Principles, pp. 231-243, Dec. 1993.
- [4] D. P. Anderson and G. Homay, "A Continuous Media I/O Server and Its Synchronization Mechanism," IEEE Computer, vol. 24, No. 10, pp. 51-57, Oct. 1991.
- [5] W. A. Montgomery, "Techniques for Packet Voice Synchronization," IEEE Journal on Selected Areas in Communications, Vol. SAC-1, No. 6, pp. 1022-1028, Dec. 1993.
- [6] S. Ramanathan and P. V. Rangan, "Feedback Techniques for Intra-Media Continuity and Inter-Media Synchronization in Distributed Multi-Media Systems," The Computer Journal, Mar. 1993.



나 인 호(In-Ho Ra)

1988년 2월 울산대학교 전자계산학과(공학사)
1991년 2월 중앙대학교 대학원 전자계산학과(공학석사)
1995년 8월 중앙대학교 대학원 전자계산학과(공학박사)

1995년 9월~현재 : 군산대학교 정보통신공학과 조교수
1997년 7월~현재 : 한국해양정보통신학회 편집이사
1997년 9월~현재 : 전주첨단영상산업추진협의회 위원
1997년 12월~현재 : 전자통신연구원 초빙연구원
관심분야 : 멀티미디어통신시스템, 분산시스템, 병렬처리



양 해 권(Hae-Kweon Yang)

1976년 2월 서울대학교 전기공학과(공학사)
1991년 2월 울산대학교 대학원 전기공학과(공학석사)
1995년 8월 전북대학교 대학원 전기공학과(공학박사)

1985년 9월~1987년 2월 : 전북산업대학교 전자계산학과 전임강사
1987년 3월~현재 : 군산대학교 정보통신공학과 부교수
1993년 6월~현재 : 전북지역정보화 추진협의회 운영위원
관심분야 : 통신망, ATM, 이동통신, 멀티미디어



고 남 영(Nam-Young Ko)

1972년 2월 광운대학교 무선통신공학과(공학사)
1980년 2월 건국대학교 대학원 통신행정전공(행정학석사)
1994년 2월 국민대학교 대학원 통신행정전공(행정학박사)

1996년 8월 Pacific Western Univ., Communication PhD
1974년 3월~현재 : 군산대학교 전파공학과 교수
관심분야 : 통신행정 및 정책, 전파통신, 멀티미디어