

타일드-디스플레이 시스템에서 실시간 동영상 상영기의 설계 및 구현

1. 기존의 동기화 방식
2. 동영상 분할 방법
3. 동기화 방법
4. 흐름 제어 방법
5. DirectShow의 사용
6. 결 론

요 약

본 논문에서는 타일드 디스플레이 시스템에서 실시간 동영상 상영기의 설계 및 구현을 하면서 발생하는 흐름제어 문제에 대한 해결책을 제시하였으며, 동기화와 DirectShow를 이용한 실시간 상영의 구현 방법을 설명하였다.

1. 기존의 동기화 방식
2. 동영상 분할 방법
3. 동기화 방법
4. 흐름 제어 방법
5. DirectShow의 사용

1. 기존의 동기화 방법

표 1 기존 시스템의 실시간 방송 및 동기화 방법들

	실시간 동영상 방송 및 수신 방법	PC들 사이의 동기화 방법
SAGE (Scalable Adaptive Graphics Environment) by U. Of Illinois at Chicago, 2006 [2]	-UDP 상에서 각 비디오 프레임들의 번호를 표시하는 프로토콜 사용 -Video Stream : Gbps interface	-TCP channel작은 지연시간과 높은 우선순위의 TCP 사용, 별도의 동기화를 위한 쓰레드 사용
TeraVision at U. Of Illinois by Chicago, 2004 [3]	-Sync Message : FastEthernet interface	-TCP 상에서 Two way handshake 방식 사용 -TCP 소켓의 TCP_NO_DELAY 옵션을 사용하여 지연시간을 최소화
AG (Access Grid) Media Architecture by GIS [4]	-UMTP (UDP Multicast Tunneling protocol)을 사용	-TCP를 이용하여 자체 구현
S/W Environments for Cluster-based Display System by Princeton Univ. [4]	-Myrinet에서의 graphic primitive 방송	-System-level sync. (SSE)와 Application-level sync (ASE) 제안 100 Mbps Ethernet상에서의 Winsock을 이용한 구현(no detailed description)

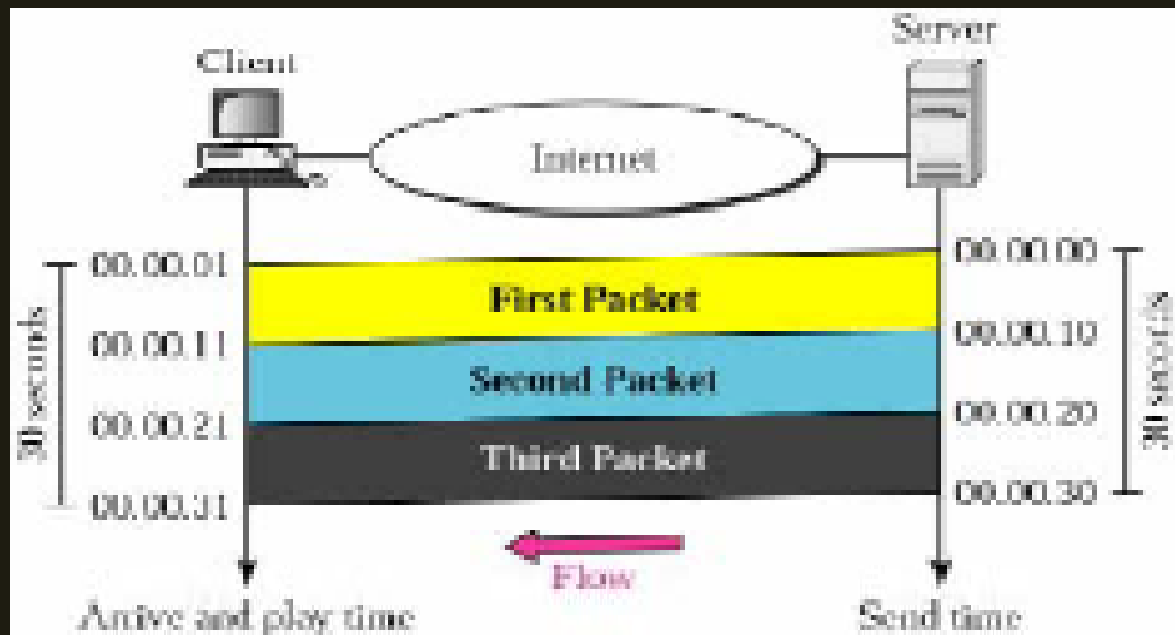
기존의 방식에서는 동기화 정보 전송을 위한 패킷의 주기적 전송으로 인해 지연시간이 발생하여 오버헤드가 발생할 수 있는 단점이 있다.

2. 화면 분할 방법

1. 송신측에서 비디오의 압축을 푼 후 타일드 디스플레이의 영역에 맞추어 나누고 난 후 전송
2. 송신측에서 비디오의 압축을 푼 후 영역을 나누어 다시 압축하여 전송
3. 송신측에서 압축 되어있는 비디오를 전송하고 좌표정보를 전송하여 각 상영기가 압축을 푼 후 해당 좌표에 맞도록 클리핑 후 디스플레이 하는 방법.
 - 네트워크 및 시스템 자원 절약의 이점

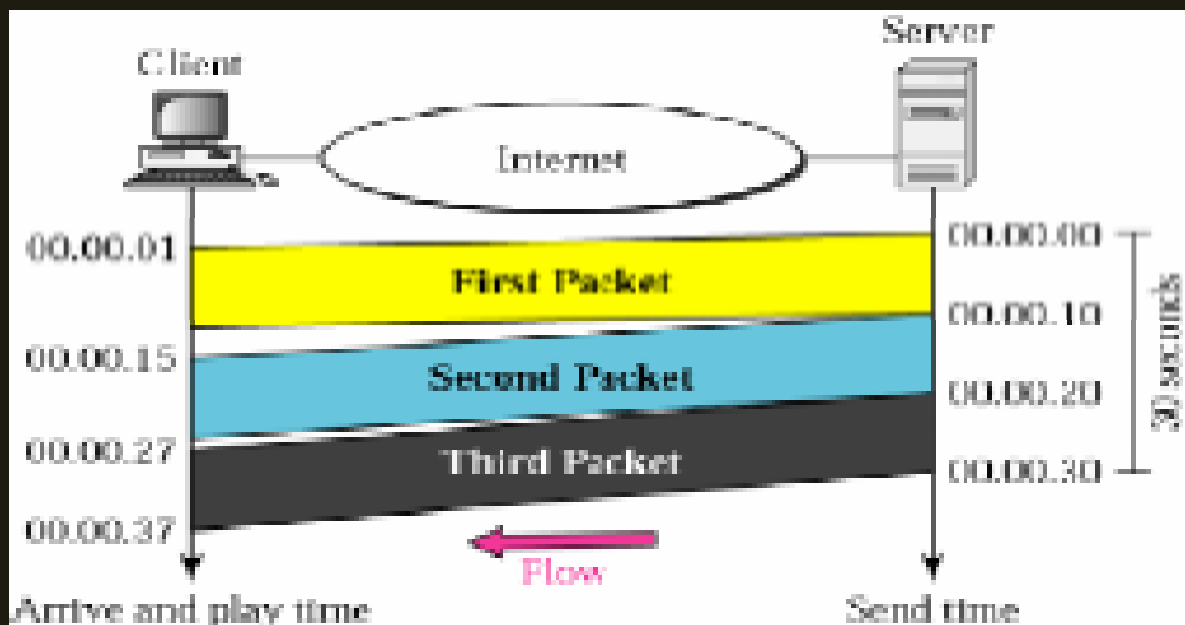
알아두어야 할 사항

이상적인 실시간 전송



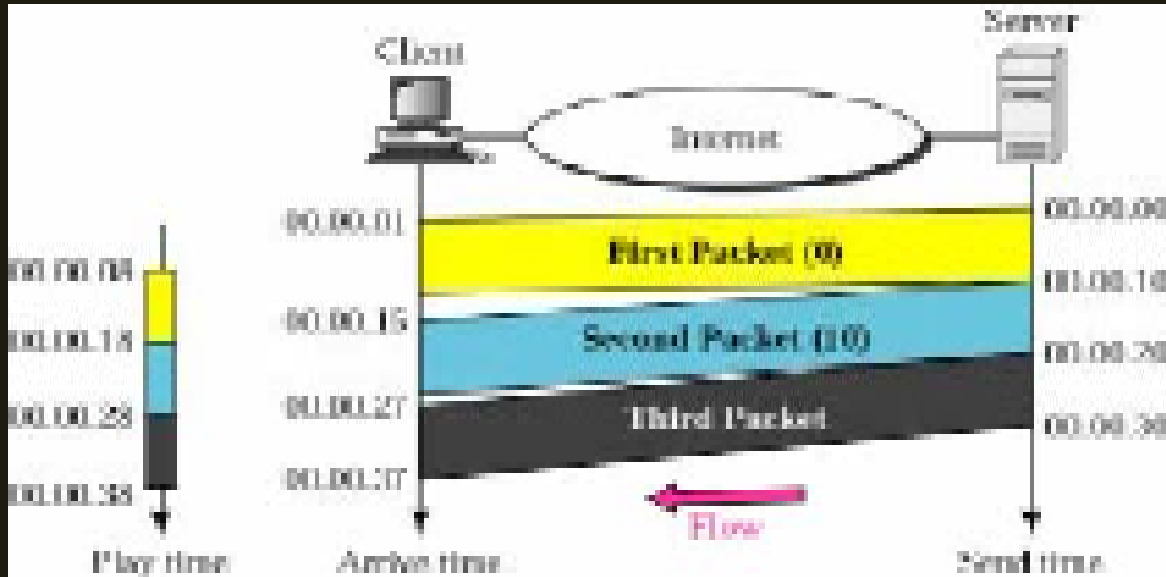
알아두어야 할 사항

지터(Jitter)란??



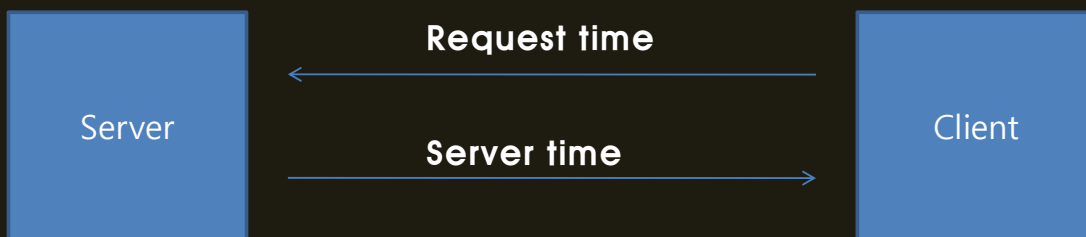
알아두어야 할 사항

지터에 대한 해결책의 하나인 타임스탬프



알아두어야 할 사항

Cristians Algorithm



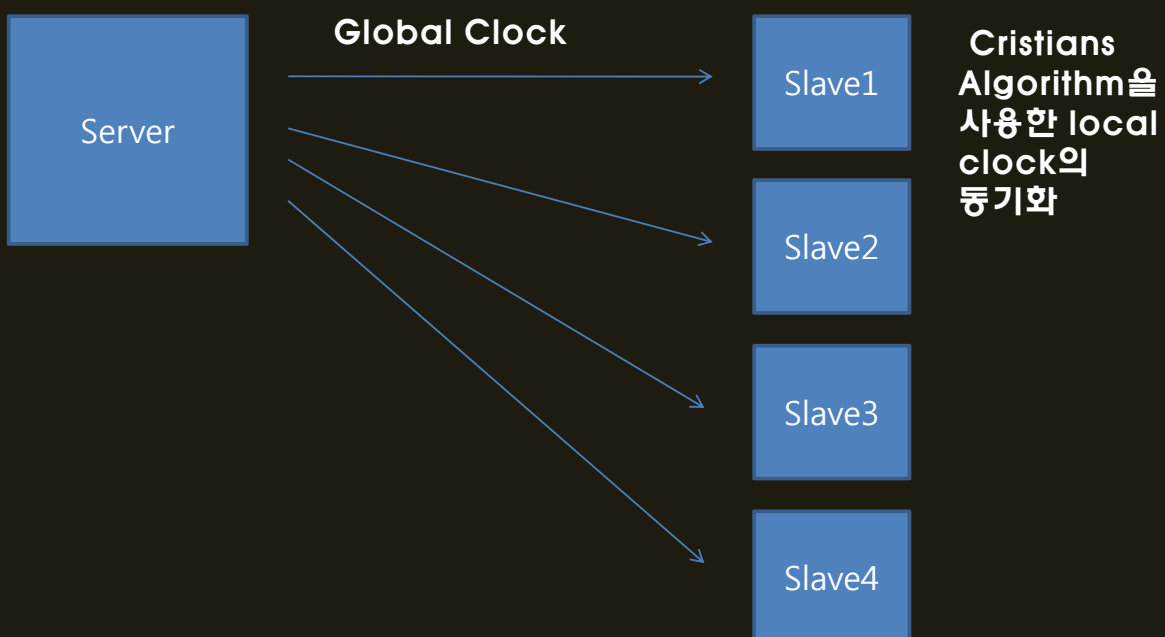
1. client가 server의 시간을 요청한다.
2. Server는 보내기 직전의 가장 최신의 시간을 client로 보낸다.
3. client는 server로부터 받은 시간에 $RTT/2$ 를 더하여 셋팅한다.
 - RTT (Round Trip Time) : 왕복소요시간
4. server의 시간에 편도의 소요시간을 더하게 됨으로서 server와 동기화 된다.

3. 동기화 방법

동기화를 위해 송신측에서 동기화 정보 전송을 하게 되면 오버헤드 및 전송지연으로 인한 동기화의 어려움이 있어 스트림타임, 타임스탬프, 레퍼런스 클럭을 사용하여 동기화 하였다.

스트림타임이란 동시재생을 위한 논리적인 재생시간을 의미하며 타임스탬프는 지터에 의한 재생시간 지연을 막기 위해 사용하며, 레퍼런스 클럭은 각 상영기의 동시 상영시 로컬하드웨어의 동기화를 위해 각각의 하드웨어클럭을 동기화 하는데 사용한다.

3. 동기화 방법



3. 흐름제어 방법

실시간 동영상 전송시 발생 할 수 있는 오버, 언더플로우 방지를 위해 상영기의 버퍼량을 감지하여 전송률을 조절해야 할 필요가 있다.

전송방법은 두가지정도로 분류 할 수 있는데,

1. 버스트(burst) 모드 : 특정한 시점에 집중적으로 보내도록 하는 모드
 - 지터 및 오버플로우 발생의 단점.
2. 일정속도 모드 : 같은 전송률도 주기적으로 보내는 모드
 - CPU의 지속적인 독점으로 인한 단점.

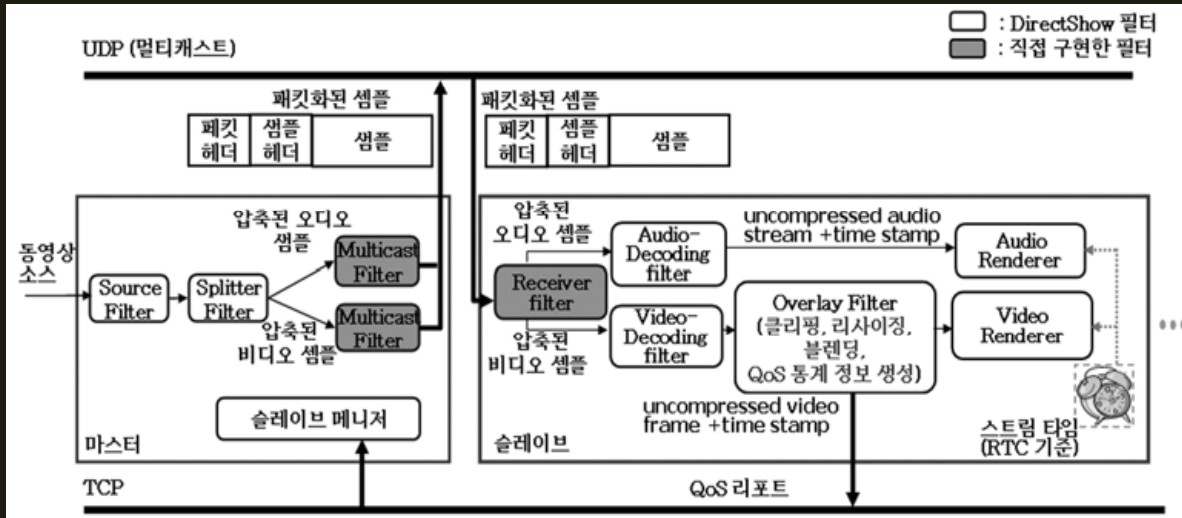
3. 흐름제어 방법

버스트 모드 전송시 버스트시간에 전송 할 버스트의 크기는 미디어 샘플(압축된 동영상의 단위)에 따르며 초당 최대 전송률 / 평균 프레임레이트의 근사값인 미디어샘플크기를 사용한다.

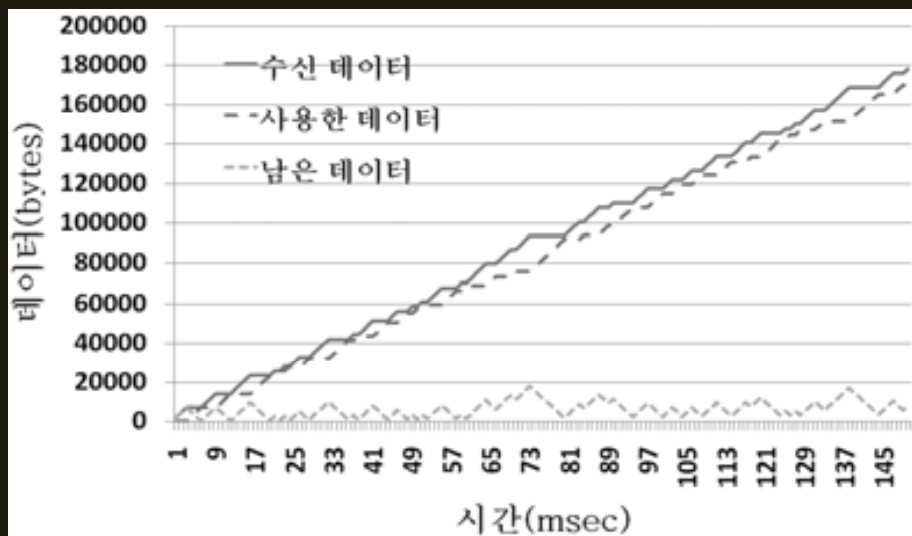
또한, 버스트 모드의 단점인 오버, 언더플로우 문제를 줄이기 위해 사전에 테스트한 버퍼의 크기(초기 버퍼링시간 * 평균 비트레이트)로 초기버퍼를 사용하여 언더플로우시 디코딩 및 렌더링이 블로킹 되는 현상을 줄여주게 되며 재생 중 전송률에 따라 버퍼의 크기를 조절한다.

4. DirectShow의 사용

본 상영기에서는 다양한 영상과 음성포맷에 대한 제한을 줄이기 위해 DirectShow에서 제공하는 필터를 사용한다.

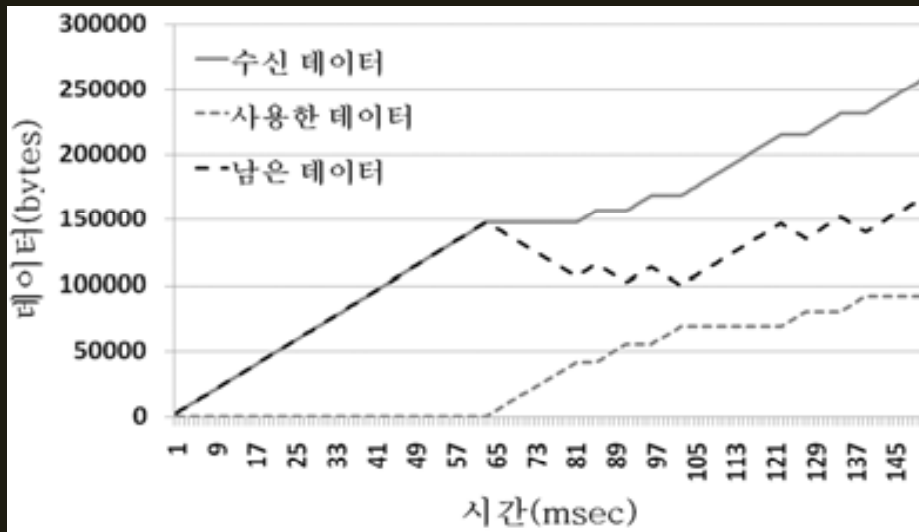


흐름제어 실험 및 분석



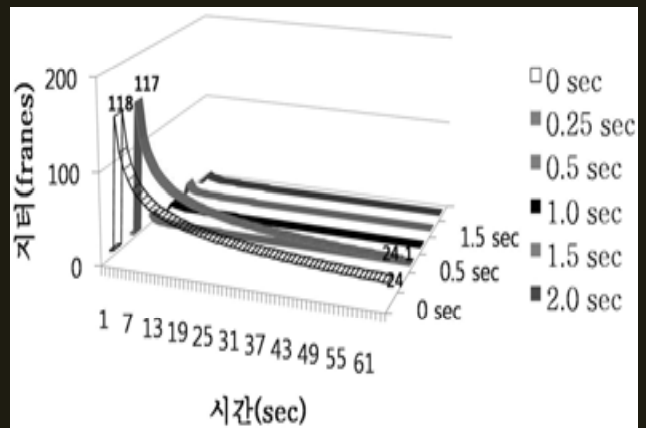
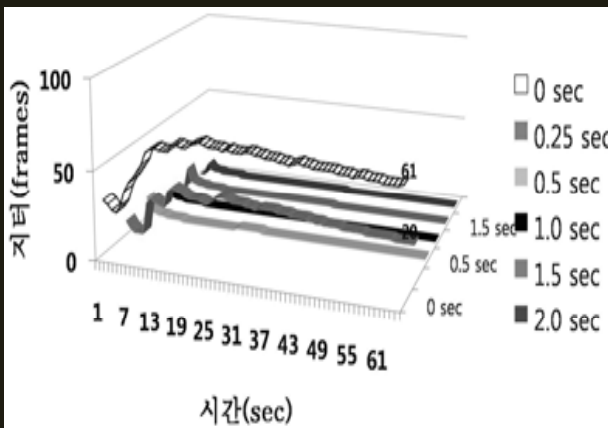
버퍼가 없을 때 사용률 및 소비율

흐름제어 실험 및 분석



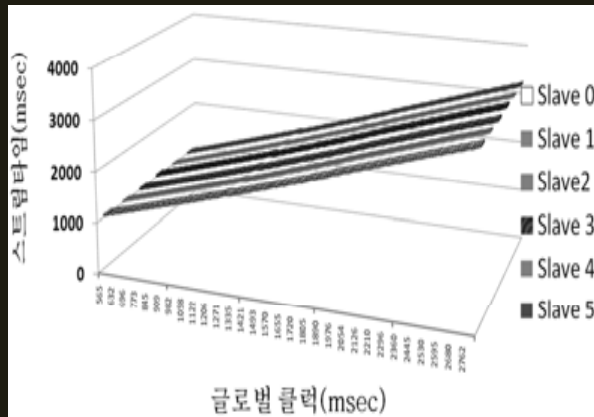
버퍼가 있을 때 사용률 및 소비율

흐름제어 실험 및 분석

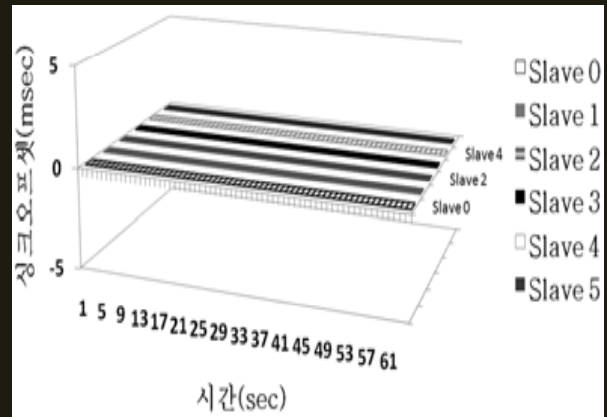


버퍼 크기에 따른 동영상 품질 변화 테스트

동기화 실험 및 분석



각 슬레이브의 스트림타임



각 슬레이브의 싱크 오프셋

5. 결론

타일드 디스플레이 시스템에서 실시간 동영상 재생에 있어서 해결해야 할 흐름제어와 동기화 방법에 대하여 해결책을 구현하고 설명하였다.

가장 중요한 부분은 실시간 처리에 필요한 흐름제어에서 적절한 크기의 버퍼사용과 동기화 방법에서의 오버플로우 및 딜레이를 최소화 하기 위한 타임스탬프, 스트림타임, 레퍼런스 클럭을 사용하여 실시간 동영상 재생의 문제점을 최소화 하고자 하였다.

👉 RTP & RTCP??