

# **Week 13.**

**Multimedia Services**

# Multimedia Applications

---

- 비디오의 특성은 압축을 잘 할 수 있다는 것이다.
- 비디오의 압축은 공간적 특성을 이용하거나 시간적 특성을 이용해서 줄이는 방식 두 가지가 있다.
  - 공간적 특성
    - 하나의 픽셀과 그 옆 픽셀은 거의 똑같은 색깔을 낼 것이다.
    - 각 픽셀의 색깔을 전송하는 것이 아닌, 해당 픽셀의 색깔과 그 색깔이 몇 번 반복되는지를 전송하여 압축한다.
  - 시간적 특성
    - 초당 30프레임일 때, 4프레임과 5프레임의 이미지는 거의 차이가 없을 것이다.
    - $i$ 번째 프레임과  $i+1$ 번째 프레임 중 공통적인 부분은 전송하지 않고 달라진 부분만 전송해서 압축.
- Bit rate를 조절하면서도 압축을 할 수 있음.
- 1024\*768의 데이터가 있을 때 데이터 픽셀을 짝수 번째 만 전달하게 된다면 512\*384 크기의 데이터가 될 것이다. 그렇다면 전체 데이터의 양은 1/4가 된다.

# Multimedia Applications

---

- 오디오의 특징

- 아날로그 데이터이다. 이것을 디지털화 해야 컴퓨터 상에서 표현하고 네트워크로 전달할 수 있다.
- 대표적인 방법으로 Pulse Code Modulation (PCM)이 있다.
- 인터넷 전화, 공중 전화망 같은 경우 초당 8000개의 샘플링을 해서 각각의 샘플링 된 데이터를 8bits로 표현하기 때문에 64kbps의 전송용량을 가진다.
- 초당 샘플링하는 데이터가 많을수록 quantization error가 줄어들어서 더 깨끗한 음악을 들을 수 있다.

# Multimedia Applications

---

- 네트워크 응용 멀티미디어의 종류

- Streaming stored audio and video – 서버에 저장된 것을 스트리밍하는 서비스

- 일시정지, 재생이 가능해야하고 (interactivity)
    - 녹화되는 속도와 제공하는 속도가 동일해야 함. 배속 재생 예외 (continuous playout)
    - 높은 throughput이 요구된다. 유튜브, 넷플릭스, 아마존 등이 있다.

- Conversational voice/video-over-IP – 화상회의 서비스 등

- 지연이 심하면 서비스에 불편함을 느낌. (Delay-sensitive)
    - 데이터 손실이 일어나도 맥락을 통해 이해가 가능한 부분이 있음. (Loss-tolerant)
    - 낮은 Delay가 요구됨. 스카이프 등이 있다.

- Streaming live audio and video – 라이브 스포츠 중계 등

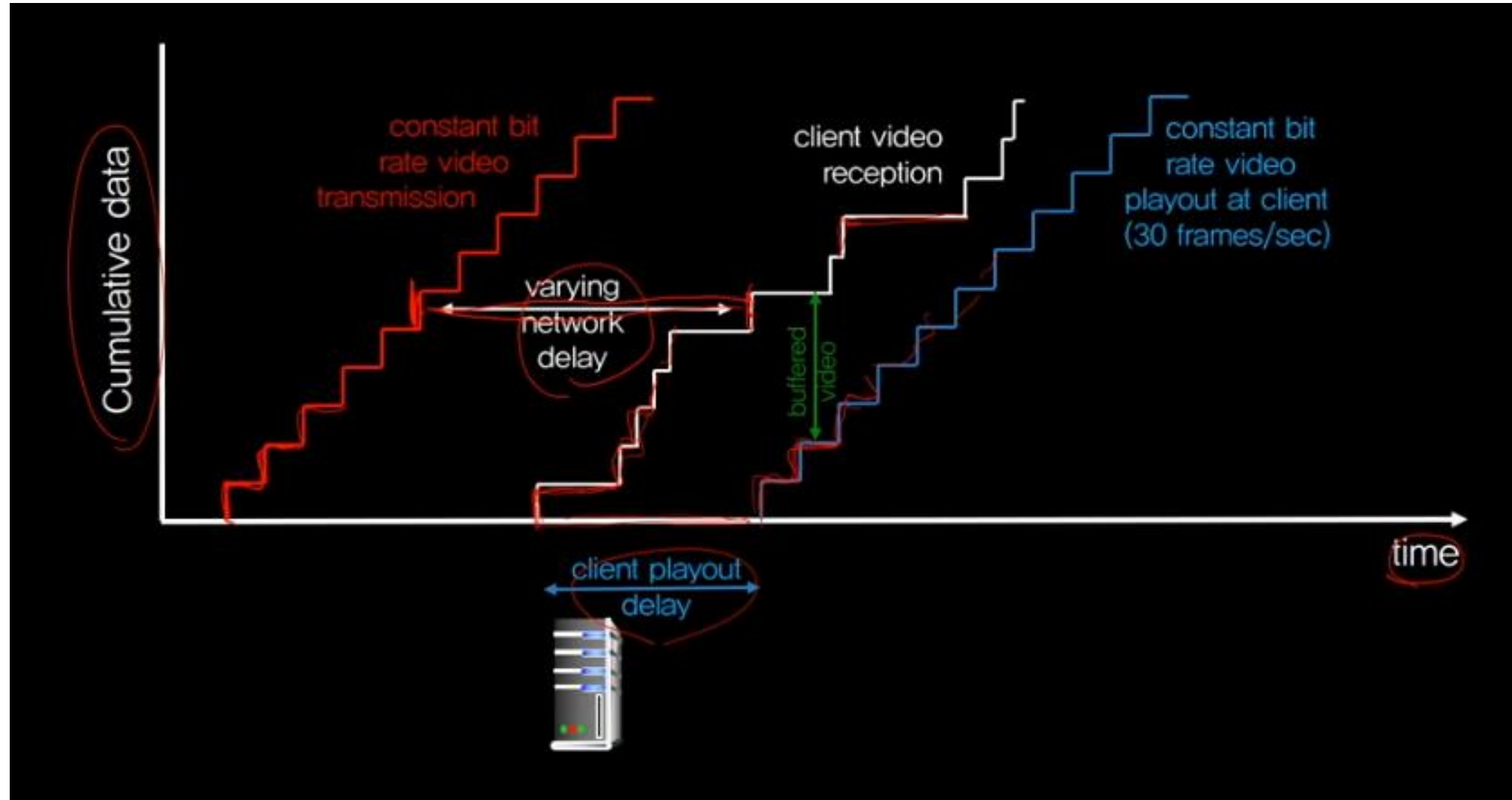
- 맨 첫번째 streaming stored audio and video와 거의 같음.

# Streaming stored video

---

- 크게 두가지로 나뉘고, HTTP에서 하나가 더 나와 총 세가지로 나뉜다.
  - UDP Streaming
  - HTTP Streaming
  - Adaptive HTTP Streaming
- 이 서비스 들의 공통적인 특징은 클라이언트 응용이 많은 데이터를 버퍼링하는 기능을 가짐. (Client application buffering)
- 전송된 비디오 데이터를 받자마자 재생하게 된다면 일정하지 않은 네트워크 딜레이 때문에 사용자는 불편함을 느낄 수밖에 없음. → 클라이언트가 데이터를 버퍼링 함. 어느정도 모아놴다가 재생함.
- CBR(Constant Bit Rate): 버퍼를 채워놴다가 재생할 때 데이터가 쭉 빠져나가게 되는데, 이 때 빠져나가는 시간.
- 초기 playout delay tradeoff: 사용자들이 재생버튼을 누르고 1~2분 기다리면 당연히 안정적인 서비스가 제공되겠지만 사용자들은 기다릴 리가 만무하다. 적당한 길이의 딜레이를 찾아야 한다.

# Streaming stored video



# Streaming stored video

---

- UDP 스트리밍 특징

- Playout delay가 짧은 편. Congestion control 메커니즘이 필요 없어서
- Bandwidth가 들쭉날쭉하기 때문에 사용자는 불편을 겪게 된다. 그러나 UDP는 playout delay가 짧기 때문에 이런 문제가 도드라져 보인다.
- 양방향 커넥션이 없다. 컨트롤 기능을 추가하려면 컨트롤 서버를 따로 뒀서 매치를 시켜야 함.
- UDP 트래픽이 방화벽 같은 침입 감지 시스템을 통과하지 못한다.

- HTTP 기반 스트리밍 특징 (HTTP over TCP)

- TCP연결을 통한 것이다 보니 한 프레임이 에러 발생하면 재요구/재전송 해야함.
- 이럴 경우에 congestion control 메커니즘이 발동해서 data rate/throughput도 낮아질 수 있기 때문에 미리 데이터를 가져오는 prefetching이 필요함.
- 그래서 UDP와 비교했을 때 훨씬 큰 playout delay
- Firewall 문제없이 통과 가능.

- HTTP byte-range header in GET message

- A지점까지 영화를 보다가 중간에 끊고 며칠 뒤 다시 보려고 할 때 A지점 다음부터 재생이 되도록 할 수 있음.
- 10초 앞으로, 뒤로 등등 이런 기능도 해당 byte-range header 정보를 GET 메시지에 담아 보냄으로써 가능.

# Streaming stored video

---

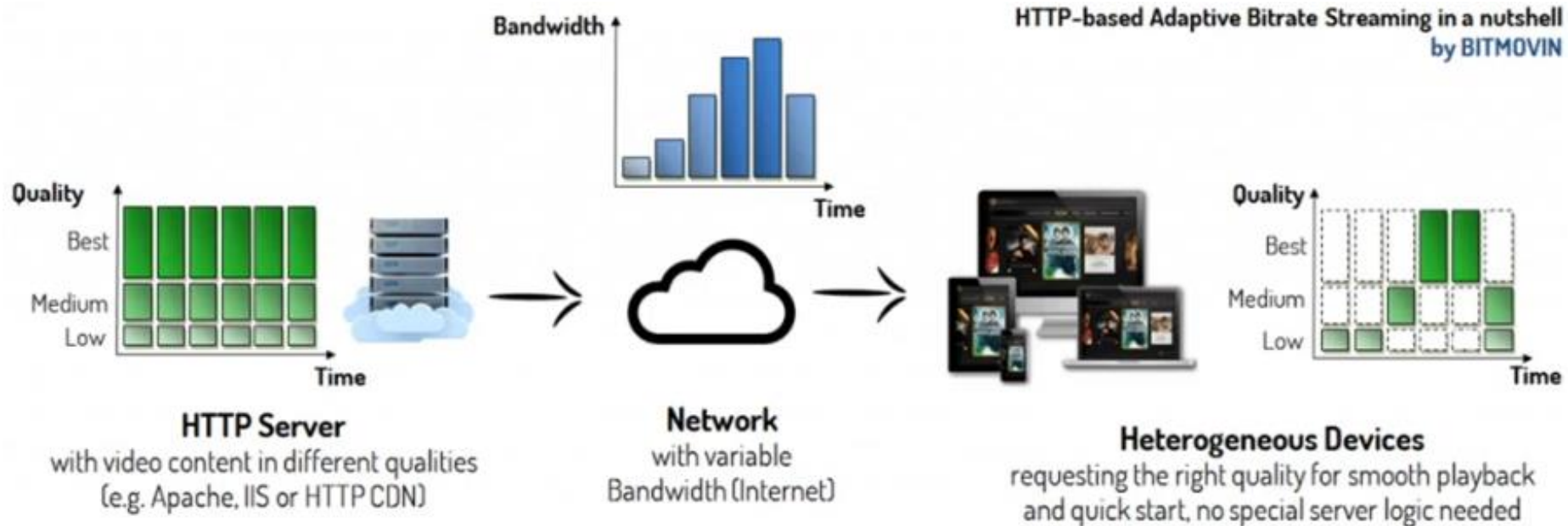
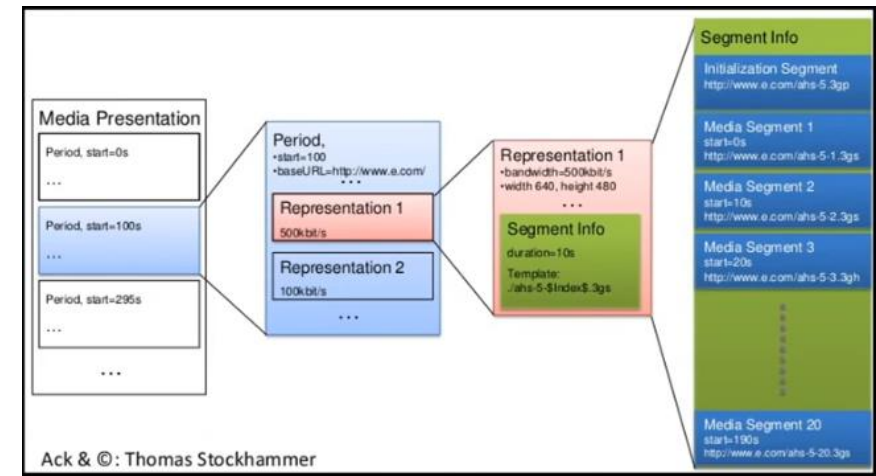
- 끝까지 재생하지 않고 종료해버리면, 미리 받아놓은 데이터가 쓸모 없어짐.  
→ 인터넷 리소스의 낭비.
- 그래서 아무리 네트워크의 상황이 좋아도 너무 많은 데이터를 한꺼번에 다 전송해 놓지는 않는다. 적당한 크기의 버퍼를 어플리케이션이 정해놓으면, 그 사이즈 만큼만 유지 함.
- HTTP 스트리밍의 단점.
  - 클라이언트들 마다 데이터를 소비할 수 있는 능력이 다른데, 비디오 크기나 품질이 한 가지 밖에 없음. → 누구는 잘 보이고, 누구는 낡김.
  - 그래서 adaptive HTTP 스트리밍이 제시됨.
- Adaptive HTTP 스트리밍 (DASH 서비스)
  - Dynamic, Adaptive Streaming over HTTP
  - 원하는 비디오 품질을 클라이언트가 결정할 수 있게 함.
  - Encoding rate도 결정할 수 있게 함.
  - 어디에 chunk를 요구할 것인지도 결정할 수 있게 함. (거리상으로 가깝지 않아도 A쪽에 congestion이 많이 발생했다면 B나 C 쪽에서 요구할 수 있도록)



# Streaming stored video

- DASH Operation : Server

- 비디오 파일을 여러 개의 chunks로 나눈다.
- 여러 encoding rate으로 각 chunk를 저장한다.
- 해당 chunk를 어디서 구할 수 있는지 URL이 담겨있는 Manifest file(xml파일)도 만들어둔다.



# Voice-over-IP

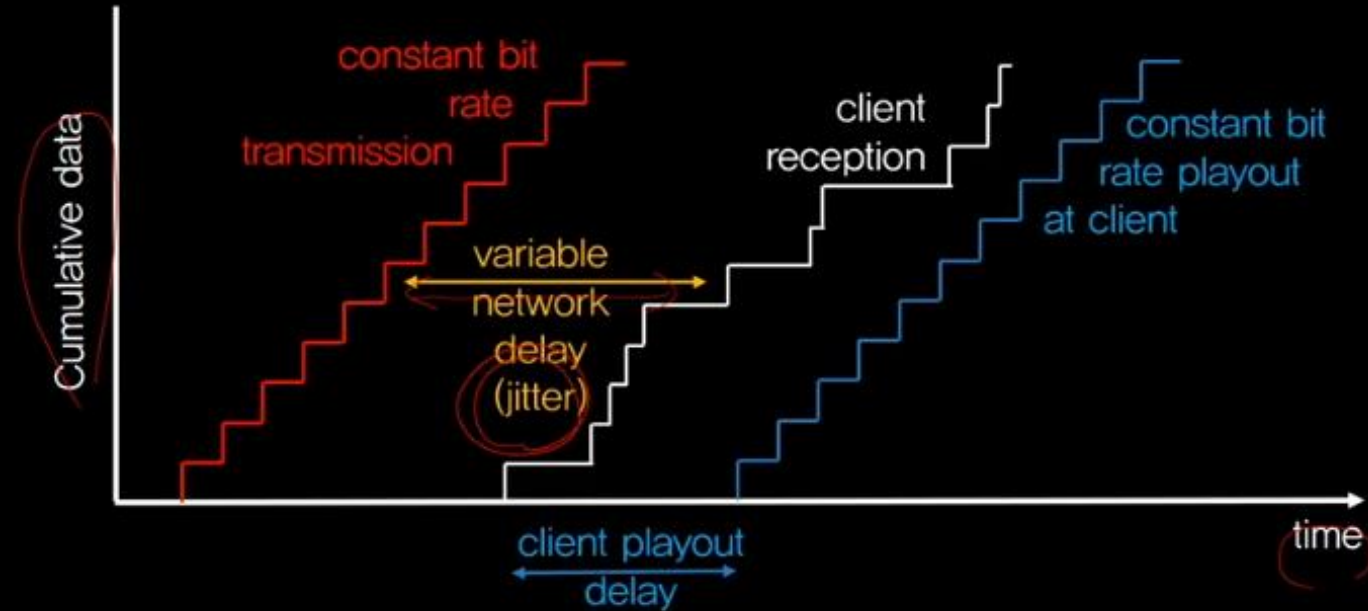
---

- 스트리밍은 몰라도 Voice-over-IP는 대화서비스다. 그러면 적어도 딜레이가 400ms 미만이어야 한다.
- 또한 인터넷을 사용한 것이기 때문에 패킷로스를 막기 위해선 TCP를 사용하면 되겠지만,
- TCP 재전송 등 end-to-end delay를 증가시키는 요인 때문에 TCP는 사용할 수 없다.
- 그래서 UDP를 채택했다.
- 연구결과 패킷의 20%정도를 잃어버려도 통화서비스가 어느정도는 가능했고, 소스 인코딩 방식을 통해 패킷 손실을 좀 숨기면 만회할 수 있는 부분이 있었기 때문.

# Voice-over-IP

- Delay jitter

- **Jitter**: fluctuation of the time from when a packet is generated at the source until it is received at the receiver



패킷이 src에서 발생해서 전송이 되어서 수신자에게 도착할 때 까지의 시간이 delay, 그 delay가 들쭉날쭉한 현상을 Jitter라 부른다.

# Voice-over-IP

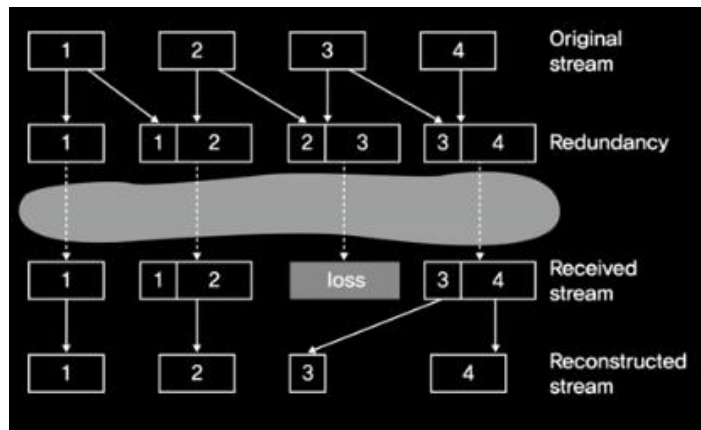
- 수신자가 Jitter를 해결하는 방법

- Timestamp : 송신자가 데이터를 샘플링 할 때 샘플링된 데이터가 언제 샘플링 된 것인지 표시
- Delaying playout
  - Fixed playout delay : 재생 지연 시간이 일정. 예를 들자면 모든 데이터가 Qms씩 미뤄진다. Q가 크면 패킷 손실율이 낮아지지만 상호대화를 한다는 느낌이 없을 것이다. (trade off).
  - Adaptive playout delay

- Voice-over-IP 서비스의 패킷 손실을 줄이는 방법

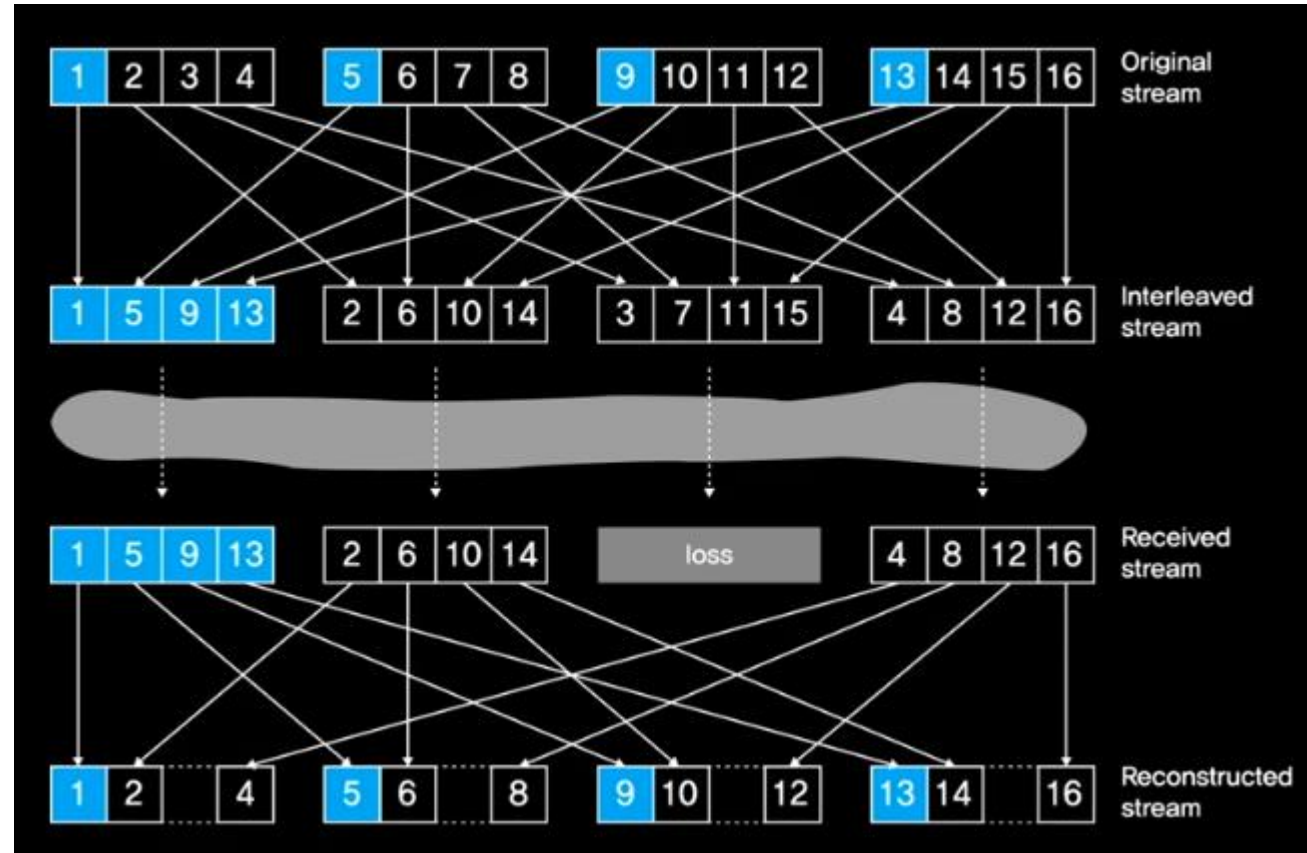
- Forward error correction

- Exclusive OR-ing of every n chunk : n개의 모든 청크를 exclusive-OR 해서 n+1번째 불필요한 청크를 만들어서 전달. 만약 n+1개의 청크 중에 하나가 손실됐다면 n+1번째 청크와 하나가 손실된 n개의 청크 모두 exclusive-OR하면 복원이 된다.
- Lower-resolution audio stream



# Voice-over-IP

- Interleaving



이건 에러를 감추는 방식의 일종이다. 중간에서 섞기 때문에 패킷 하나가 손실되어도 통화 내용이 통째로 사라지는 것이 아니라 중간중간 음절만 빠진다.

# Realtime Transport Protocol (RTP)

---

- 오디오나 비디오데이터의 공통적인 부분들을 표준화 해서 제공을 하면 멀티미디어 응용 개발자들이 플랫폼이 다르더라도 서로 호환성을 가질 수 있지 않을까? 해서 나왔음.
- 주로 최종 송신자, 수신자에서 플레이가 된다. UDP 기반이다.
- 이름에 Real-time이 붙어있지만, 데이터 전달의 실시간성을 보장하지는 않음.  
레코딩 되는 시점의 느낌하고 똑같은 방식으로 들을 수 있게 해주는 프로토콜임.

# Realtime Transport Protocol (RTP)

## ● RTP 헤더

payload type	sequence number	time stamp	Synchronization Source ID	Miscellaneous fields
--------------	-----------------	------------	---------------------------	----------------------

- **Payload type** (7 bits): indicates type of encoding currently being used
  - payload type 0: PCM mu-law, 64 kbps
  - payload type 3: GSM, 13 kbps
  - ...
  - payload type 31: H.261
  - payload type 33: MPEG2 video
- **Sequence number** (16 bits): increment by one for each RTP packet sent
  - detect packet loss, restore packet sequence
- **Timestamp field** (32 bits): sampling instant of first byte in this RTP packet
  - for audio, timestamp clock increments by one for each sampling period
  - if a chunk has 160 encoded samples, timestamp increases by 160 for each RTP packet
- **SSRC field** (32 bits): identifies source of RTP stream; each stream in an RTP session has distinct SSRC

- Payload : 이 데이터가 어떠한 방식으로 인코딩 되어 있는지를 나타냄. 7비트라  $2^7$ 까지 표현 가능.
- Sequence number : 16비트, RTP 패킷을 전송할 때마다 값이 1씩 증가.
- Timestamp field : 어떤 프레임이 '첫 번째 시작부터해서 얼마큼의 시간이 흘렀다'는 것을 나타냄.
- SSRC (Synchronization Source ID) : 하나의 세션 안에 데이터 스트림이 여러 개가 있을 수 있음. 이 데이터 스트림들을 구별하기 위해서 부여하는 랜덤한 번호.

# Realtime Transport Protocol (RTP)

---

- RTCP

- RTCP는 컨트롤 프로토콜. RTP가 제대로 동작하고 있는지를 지속적으로 관찰.
- IP와 ICMP의 관계가 RTP와 RTCP의 관계랑 비슷.
- RTCP는 RTP를 통해서 보내는 데이터의 5%미만이다.

- RTSP

- RTSP는 HTTP 스트리밍에서 필요가 없다고 언급했던 컨트롤을 위한 TCP 커넥션이다.
- UDP 기반 서비스에서는 스트리밍 시 pause나 10초 넘기기 등 기능이 안돼서 추가로 마련한 커넥션이다.



# Session Initiation Protocol

---

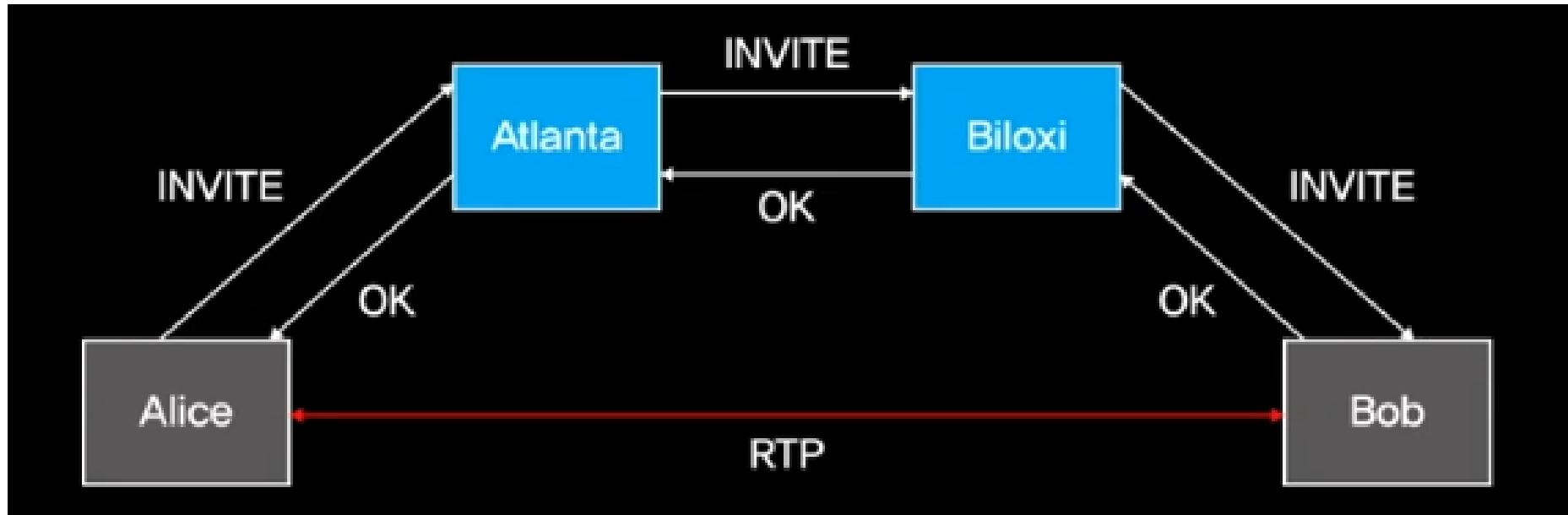
- 세션을 초기화하는 프로토콜. 사람들이 데이터 통신을 원하는 상대의 URL을 기억하고 있는게 아니기 때문에 이름이나 이메일 주소 같은 걸로 연결을 설정할 수 있는 방식이 있으면 좋겠다.
- 그걸 매핑하는 서버를 만들면 되지 않나?  
→ DHCP를 주로 사용하기 때문에 어떤 IP주소를 쓸지 알 수 없음. 이를 해결하기 위해 나온 것이 SIP

## ● SIP Services

- 전화 하는 caller와 받는 callee사이의 IP 네트워크 위아래가 전화 설정을 해줘야 함.
  - 연결을 설정해서 누가 전화를 하고 싶어하는지 callee에게 알려줘야 함.
  - 인코딩 방식에 대한 합의를 이루게 할 수 있어야 함.
  - 전화를 끊을 수 있게 해야 함.
- Caller로 하여금 callee가 사용하고 있는 IP주소를 알아낼 수 있는 방식을 제공해야함.
- Call에 대한 매니지먼트를 가능하게 해야 함.

# Session Initiation Protocol

- SIP 프록시
  - 프록시들이 유저들의 IP주소를 알아야함. → SIP register를 통해 알 수 있음.



# Network Support for Multimedia

---

- 멀티미디어 데이터는 일반적인 데이터에 비해서 딜레이에도 민감하고 많은 bandwidth를 필요로 함. 그래서 서포트가 필요한데, 서포트하는 방식을 크게 세가지로 나뉘볼 수 있음.
1. Making the best of best-effort service
    - IP를 베스트로 만든다는 이야기. 무슨 문제가 생기면 bandwidth를 추가로 설치한다든지 등.
    - Network dimensioning : 어떻게 네트워크를 구성하는 것이 합리적인가
  2. Differentiated service
    - 데이터 트래픽을 구별을 통해 Queuing delay를 짧게 해서 지연시간을 보장해준다.
  3. Per-connection Quality-of-Service guarantees
    - 각각 커넥션 별로 구별하는 것. 서비스 요금을 다르게 받고 있다면 특정 사용자의 데이터는 빨리 전달해 준다든지 하는 것이다.

1번으로 갈수록 쉽고, 3번으로 갈수록 복잡한 방법이다. 그러나 1번은 돈이 많이 든다.

# Network Support for Multimedia

---

- Dimensioning Best-Effort Networks

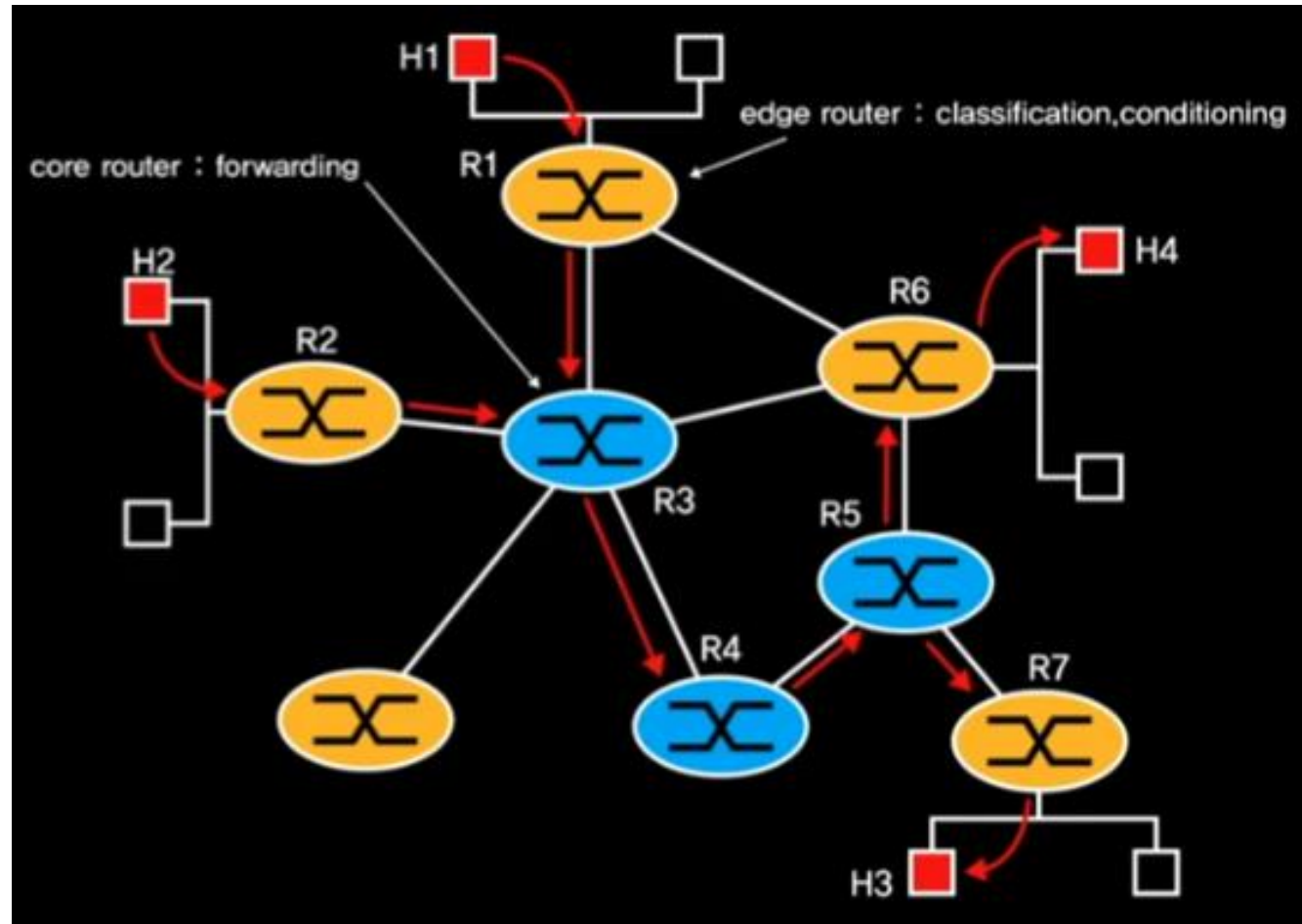
- 돈을 투자해서 link capacity를 늘려서 congestion이 일어나지 않도록 함.
- 네트워크 메커니즘들을 개발할 필요 없는 것이 장점.
- 비용이 많이드는 것이 단점.
  - ◆ 그래서 나온 것이 Network dimensioning(최적의 네트워크 topology 연구)
  - ◆ Bandwidth Provisioning(각각의 링크에 어느 정도의 bandwidth를 할당해야 하는가)

- Differentiated service

- 패킷을 구별함으로써 처리해야하는 데이터의 우선순위를 정할 수 있다.
- 그런데 특정 데이터의 트래픽이 항상 있어서 다른 데이터를 처리하지 못하는 상황이 발생했다.
  - Traffic Isolation을 도입하여 전송되지 못하는 데이터가 있는지 감시한다. Or 정해진 링크 capacity를 초과하는 데이터는 억누른다.
- 이것이 링크 스케줄링 이라는 형태로 나타나 있다. (chap 4 참조)

# Network Support for Multimedia

- Differentiated service
  - Diffserv Architecture



# Network Support for Multimedia

---

- Per-connection Quality-of-Service guarantees

- Resource reservation이 필요함. 만약 모든 자원이 사용되고 있어서 예약이 안 된다고 하면 reservation을 받으면 안된다.

- 그래서 예약을 받을지 말지를 정하는 메커니즘인 call admission control mechanism이 있다.

# summary

---

- RTP (Real-time Transport Protocol): IP와 UDP 상에서 실시간 멀티미디어 서비스를 위해 사용하는 프로토콜
- RTSP (Real-Time Streaming Protocol): 스트리밍 멀티미디어 서비스의 재생/멈춤, 빨리감기, 되감기 등의 서비스를 제공하는 프로토콜
- RTCP (Real-Time Control Protocol): RTP 서비스에 대한 통계 및 제어 정보를 전달하기 위한 프로토콜
- SIP (Session Initiation Protocol): 실시간 멀티미디어 서비스를 위한 연결을 설립, 유지, 종료하기 위한 프로토콜
- DiffServ: 서로 다른 성격의 트래픽을 구분하여 각각에 맞는 전송 품질을 제공하기 위한 네트워크 구조

감사합니다