컴퓨터망 13주차

Multimedia Networking

Streaming stored

Netflix

Voice over ip

카카오톡

Streaming live

1. streaming stored

constant bit rate 초당 30프레임 정도.

전송하면

순서대로, 일정한 bit rate 로 보냄.

Network delay 가 variable 하면.

(Delay variation 이 심하다.)

Jitter 가 존재한다. =Delay(max) – Delay (min ) ==0

충분히 시간을 벌어놓고, 재생하여야 variable 한 delay 대응 할 수 있음.

Client playout delay 라고 한다.

수신 데이터를 버퍼에 쌓아놓는다. Buffering

Buffer 에서 차는 비율은 가변.

Buffer에서 쓰는 양은 일정

Buffer 에서 차있는 레벨 Q(t) 한번 더보기

Trans port 계층으로 사용할 수 있는 것 UDP, TCP(reliable)

UDP 에서 제공할 때는?

Congestion level 과 transmission level 과 무관함

그렇기 때문에 LOSS 가 많이 생길 수 도 있음.

그럼 장점은 뭘까?

음성 같은 내용이 날라올 때 TCP 는 모든 순서를 맞춰놓고 버퍼에서 올리는데 (Play)

UDP 는 순서를 크게 고려하지 않고 덩어리 째로 올려버림, Loss 가 있어도 읽음.

( application level 에서 error recovery 함, 단 시간이 충분하다면….)

But UDP 는 fire wall 을 잘 통과 못하더라… 연결 설정을 잘 안하니까.

But UDP 위에 RTP를 올린다면..? 달라질 수 있음.

Tcp 위에서 제공한다면 (HTTP 에서 제공한다) 고 보면됨.

하지만 HTTP over QUIC 도있음 생각해보면 (UDP 기반)

Method Get 을 사용함.

Congestion control 함. Loss x

대신 fill rate flucate 생길 수 있음. (but inorder delivery)

Firewall 통과 잘 할 수 있음.

DASH Protocol 사용 할 수 있음.

Quic 사용하는 경우도 있고.

UDP 위에서 전송할 때 RTP 사용한다고 함.

HTTP 는 실제 데이터 전송은 RTSP, control of Streaming 이 필요함.

Rewind, fast, forward, pause 같은 user control 을 제공해야함.

HTTP 같은 경우는 해당 service 를 지원안함. 원래 그목적으로 만든거 아님.

Present description?

Meta file 이 Data 의 종류를 나타냄.

Set up 하고

Play 요청, data 수신(전송)

RTP 로 전송 (over UDP)

RTSP? RTP 둘다 같은건가 UDP TCP 상관 없이 쓰는 건가?

FTP 에서 요청과 파일 송수신은 별도의 연결을 통해 송수신 함.

Out of band control channel

RTSP control message 는 별도의 채널(얘가 control message 인가보다)

VoIP end to end delay requirement

좋은 network 망 위에 돌려야됨. 특징여러개 나열한거 간단하게 공부하기.

Session initialization: 이동시에 IP 가 자꾸 바뀐다면, 나의 IP 주소, port number, encoding algorithm 등을 상대방에게 알려줘야 함. Session 을 만들기 위한 하나의 사전 작업

Value added services : call 을 다른곳으로 전달하는 forwarding, 부재시 전화번호 연결.

Screening(차단), recording (녹음) 등의 기능 등을 제공한다.

참고… 911 과 같은 서비스도 지원해야함(긴급전화번호..)

VOIP 특성

스피커(화자)는 계속해서 말하는게 아님. Talking spurts, silent periods 반복

8000번 sampling 8bit quntaization= 64kbps during talk spurt 기간에 전송.

20msec 치 Pkt 만들어서 전송. 64kbps 는 어디서 쓰는거야?

Pkt Loss

Network congestion 때문에 Loss 생김.

Delay 로스도 생김. Ip datagram 이 너무 늦게 도착할 때( Play out 시점에 맞지 않는 data)

Loss tolerance 기법이 다양함. 화질이나 음성이 좀 깨져도 재생하는데 크게 문제 없음.

Fixed play out delay

일정한 간격을 기다려준 뒤 constant bit rate로 재생함.

한참 기다릴수록 안정적.

실시간 interactive 가 떨어짐. Loss가 대신 적음.

전송할 때 time stamp t를(pkt 만들때) 를 가짐. 그래서 play out 을 t+q 때 재생함.

이 q 안에 pkt 가 도착 못하면 loss 됨 pkt.

Large q : less pkt loss

Small q : large pkt loss

Adaptive playout delay

목표: low late loss rate, play out delay

Play out 입장에서

Talk spurt 는 일정하게 필요함. 대신 silence period 를 가변적으로 늘렸다가 줄이자.

빨리말한다. 늦게말한다. 차이. 그 한번에 쫘르륵 쏟아지듯 말하는게 이것때문인듯.

Network 혼잡도를 고려, play out delay( talk spurt 의 시작점) 을조정하겠다.

==silence period 가변적으로 조정.

Chunk 는 여전히 20msec으로 만들어짐.

Delay == 공식.

Variation : 앞에 평균 + 뒤에 오차만큼 학습해주기 .

Di 는 구했고…총 v 까지 다 더해서. Play out delay 계산해줌.

VOIP 는 recovery 필요

A가 B에게 전송 (Loss 발생 인지 ) retransmission .

Time permitting: A가 다시 보냈을 때 delay loss 가 생기지 않는 범위에서 .

요청 -> 재전송된 데이터가 날아옴 이 과정이 과연 delay loss 내에 들어올 수 있을까?

Delay loss에 대한 우려가 상당히 큼. ( 추가적 rtt 필요)

Forward Error correction, 충분한 bit 를 보내서, 거기서 correction 해버리자.

Simple FEC

Chunk N 개.

모든 chunk 에 대해 exclusive or 한다.

그리고 그 값을 N+`1 자리에 넣는다.

임의의 chunk 가 사라졌을 때, 다시 전체에 대해서 exclusive or 값을 구해서 그 사라진 chunk 값을 구할 수 있다.

또다른 FEC 방법.

Original shceme 이 존재할 때

Redudancy 추가

1번을 보내고, 2번을 보낼 때, 1번의 data를 요약한 저음질(lower resolution) 데이터를 보낸다.

3번을 보낼 때, 2번 저음질 데이터를 보낸다.

로스가 생긴다면, 3번의 저음질 데이터로 땜빵하고 그냥 재생시킴.

Loss concealment technique (loss 가 안생긴 척 하는 방법)

아니면 아류방법으로, 2 번 한번 더 3번자리에 재생시켜도 되고

2 ~4 의 평균 값으로 3번을 재생 시키는 방법도 있음.

Interleaving to conceal loss

원래 오리지날 schem 을 shuffling 한다.

Pkt 랑 chunk 랑 같은건가?

섞인 pkt 들이 날라가니까

하나가 없어져도

원래 pkt 들이 다시 재정렬되면

일부분만 loss 되기 때문에 괜찮음.

== 조금씩만 잡음만 들림. 고통 분담.

Skype 나 카카오톡 들도 이런 방법 들을 사용한다고 예측.

다음 시간에 설명할 것 은 RTP 에 대해서 설명할게요

RTSP 는 컨트롤 하기 위한 것.

RTP 는 실제 데이터가 전송되는 방법

그림을 잠깐 확인하면 ( APP Layer- RTP –UDP –IP –Data link Physical)

원래 순서 안 맞는 것 APP 에서 다 책임져야하는데, RTP 가 time stamp 같은 것 대신 관리해줌, because app 마다 일일이 관리하기 어렵기 때문에 RTP 로 묶어서 관리. UDP 위에서 동작함.

TCP 도 APP 에서 책임지긴 하나? 동작은 TCP 에서만 해도?

무선은 시험에 안 들어감.