

네트워크 상태와 영상 특성 기반 실시간 적응형 비디오 스트리밍 시스템



부산대학교 정보컴퓨터공학부
지도교수 : 김종덕

팀명 : BitFlow
팀원 : 202155515 김남희
202155553 박은재

목차

1	과제 개요	2
1.1	개요	2
1.2	목표	2
1.3	기대 효과	3
2	과제 구성	4
2.1	요구사항 분석	4
2.2	시스템 구성	4
2.3	개발 도구 및 기술	5
2.4	현실적 제약 사항 분석 결과 및 대책	5
3	개발 일정 및 역할	7
3.1	개발 일정	7
3.2	역할 분담	7
4	참고 문헌	8

1 과제 개요

1.1 개요

스트리밍 서비스란 인터넷을 통해 음성이나 영상 등 미디어 콘텐츠를 실시간으로 재생하는 서비스로, 대표적으로 유튜브, 넷플릭스 등이 스트리밍 서비스를 제공하고 있다. 사용자는 Wi-fi, 모바일 데이터, 유선 인터넷 등 다양한 환경에서 스트리밍을 시청하는데, 이처럼 각기 다른 조건에서도 끊김 없이 안정적으로 스트리밍 서비스를 제공하기 위해서는 "적응형 비디오 스트리밍 시스템(Adaptive Bitrate Streaming, ABR)"이 필수적이다.

기존의 스트리밍 방식은 실시간 네트워크 상황과 영상의 특성을 고려하지 않고 스트리밍을 제공하였기 때문에 네트워크 상황이 바뀌었을 때 버퍼링이 걸리거나 속도가 느려지는 등의 문제가 발생하였다. 또한 다수의 시청자가 고품질 영상을 시청하는 경우, 동영상 트래픽이 폭발적으로 증가하면서 트래픽 관리 비용 또한 문제가 되고 있다. 실시간 비디오 스트리밍 시스템은 이를 해결하기 위해, 클라이언트 측에서 실시간 네트워크 상태 및 영상의 복잡도와 인코딩 특성을 분석하고 이를 서버에서 AI 모델에 적용하여 사용자에게 끊임 없는 최적의 스트리밍 서비스를 제공하고 기업은 효율적으로 트래픽을 관리할 수 있게 한다.

1.2 목표

1. 실시간 네트워크 모니터링 및 영상 복잡도 분석

- 클라이언트 측에서 ntopng을 통해 클라이언트-서버와 서버의 네트워크 환경(Wi-fi, 모바일 네트워크) 등 네트워크 상태를 실시간으로 모니터링한다. 웹 어플리케이션을 통해 이러한 네트워크 상태를 시각화할 수 있는 도구를 제공한다.
- 모션 벡터 분석과 인코딩 특성 기반 분석을 기반으로 영상의 복잡도를 측정하고, 이를 실시간 네트워크 모니터링 분석 결과와 함께 서버로 전송한다.

2. 실시간 비디오 스트리밍 서비스 제공

- 서버 측은 FFmpeg 라이브러리를 사용하여 HLS 프로토콜을 기반으로 다수의 시청자(클라이언트)에게 실시간 스트리밍 서비스를 제공한다. 동시에 클라이언트 측에서 전달받은 네트워크 품질 및 영

상 특성 보고서를 AI 모델에 적용하여 얻은 결과를 활용하여 실시간으로 스트리밍 품질을 조정하여 시청자가 끊김 없는 스트리밍 서비스를 제공받을 수 있도록 한다.

3. 실시간 스트리밍 품질 조정을 위한 AI 모델 개발

- 영상의 움직임을 나타내는 모션 벡터 분석과, 영상의 인코딩 특성을 분석하여 최적의 스트리밍 품질을 예측하는 AI 모델을 개발하여 클라이언트 측에서 측정한 분석 결과를 적용하여 서버에서 실시간으로 스트리밍 품질을 조정할 수 있도록 한다.

1.3 기대 효과

실시간 적응형 비디오 스트리밍 시스템을 적용하면 갑작스러운 네트워크 환경의 변화로 인해 연결이 불안정해지더라도 해상도를 낮추는 등의 대응을 통해 지연을 줄임으로서 어떠한 환경에서도 끊김 없는 최적의 스트리밍 서비스를 제공 받을 수 있을 뿐만 아니라 불안정한 환경에서는 고품질이 아닌 저품질의 영상을 제공하면서 사용자는 데이터 비용을 절약할 수 있다.

또한 동영상 화질이 높아질 수록 다수의 시청자들이 스트리밍 서비스를 시청할 시 트래픽이 폭발적으로 늘어날 수 있다. 실시간 적응형 비디오 스트리밍의 경우, 이러한 영상의 화질을 조정함으로써 트래픽을 효율적으로 관리할 수 있다.

2 과제 구성

2.1 요구사항 분석

(1) 안정성 확보 : 사용자의 Wi-fi, 모바일 네트워크 환경이 불안정해지더라도 스트리밍 서비스는 지속적으로 제공 받을 수 있도록 한다.

(2) 지연 시간 최소화 : 비디오 지연 시간이 너무 길 경우, 시청자가 사건이 일어나고 실제로 시청하기까지 오랜 시간이 걸리게 된다. 이를 방지하기 위해 비디오 세그먼트 길이를 조정하여 지연 시간이 6초 이내로 유지될 수 있도록 한다.

(3) 사용자 편의성 : 간단한 대시보드 UI를 통해 현재 네트워크 품질 상태 및 영상의 복잡도 분석 결과를 확인할 수 있도록 한다.

(4) 브라우저 호환성 : 실시간 스트리밍 서비스는 Chrome, Micro Edge, Firefox 등 주요 웹 브라우저에서 재생이 가능하도록 구현한다.

2.2 시스템 구성

(1) 서버

- 서버는 웹캠 또는 저장된 영상을 FFmpeg 라이브러리를 사용하여 인코딩 및 전송한다.

(2) 클라이언트 (시청자)

- 클라이언트는 스트리밍 영상을 재생할 수 있으며, 실시간으로 네트워크 상태와 영상의 복잡도를 분석한다.

- 실시간 네트워크 상태 분석은 ntopng을 사용하며, 클라이언트-서버 간 연결 상태와 클라이언트의 네트워크 환경(Wi-fi, 모바일 네트워크)에서 패킷 손실률, 왕복 지연 시간, 대역폭, 비트레이트 등을 측정한다.

- 영상 복잡도 분석의 경우 영상의 움직임을 나타내는 요소인 모션 벡터를 통해 프레임 간의 움직임을

추정하며, 영상의 세그먼트 크기, 비트레이트, 프레임 레이트, 해상도 등의 인코딩 특성을 반영하여 영상의 복잡도를 분석한다.

- 실시간 네트워크 분석 및 영상 복잡도 분석 결과를 서버로 전송한다.

(3) 서버

- 클라이언트 측으로부터 전달받은 네트워크 상태와 영상 복잡도 분석 결과를 AI 모델에 적용하여 최적의 비트레이트 및 해상도를 예측한다.

- 네트워크가 느리거나 불안정하면 비트레이트를 낮추어 저화질의 영상을 전송하고, 네트워크가 빨라지면 비트레이트를 높여 고화질의 영상을 전송하도록 한다.

- 영상 복잡도 분석 결과에서 영상의 복잡도가 높아질 수록, 예를 들어 스포츠나 게임 스트리밍의 경우는 비트레이트를 높여 부드러운 영상 재생이 가능하도록 한다.

2.3 개발 도구 및 기술

요소	내용
UI	HTML, CSS, Javascript - 비디오 스트리밍 시스템에 필요한 웹 어플리케이션 개발
백엔드 서버	Python, FaskAPI
AI 모델	tensorflow, keras - 네트워크 품질 및 영상 인코딩 특성 분석 모델 개발 OpenCV - 모션 벡터 분석
미디어 처리 도구	FFmpeg - 영상 인코딩 및 실시간 스트리밍을 위한 소프트웨어
네트워크 모니터링 도구	ntopng - 네트워크 트래픽을 실시간으로 분석하고 시각화
인프라	Github, Docker, AWS EC2 - 가상머신 및 웹 어플리케이션 배포
데이터베이스	Postgresql

2.4 현실적 제약 사항 분석 결과 및 대책

(1) 실시간성 구현의 어려움 (지연 시간 최소화)

-> 네트워크 데이터와 영상 특성을 AI 모델에 입력하여 최적의 비트레이트 및 해상도, 프레임 레이

트를 예측해서 영상 품질을 조정하도록 한다.

(2) 학습 데이터 부족으로 인한 AI 정확도 저하

-> 시뮬레이션을 통한 데이터 생성 또는 공개 데이터셋을 활용한다.

(3) 외부 인프라 사용 시 비용 발생 및 실시간 테스트 환경 구성의 제약

-> 초기 개발은 로컬 Docker 기반으로 테스트하고, 데모 시 클라우드를 사용한다.

(4) 상대적으로 적은 팀 구성원 수

-> 역할 분담을 명확히 하여 개발을 진행하고 서로 맡은 부분을 알려주면서 진행하도록 한다.

3 개발 일정 및 역할

3.1 개발 일정

업무	5월				6월				7월				8월				9월				10월			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
스터디 및 착수 보고서 작성																								
ffmpeg 스트리밍 및 웹 어플리케이션 구현																								
영상 복잡도 분석 알고리즘 개발																								
네트워크 모니터링 및 인코딩 분석 데이터 수집																								
실시간 적응형 스트리밍 구현																								
개선 사항 분석 및 수정																								
최종 평가 및 배포																								
최종 보고서 및 발표 준비																								

3.2 역할 분담

이름	역할 분담
김남희	영상 처리 및 분석 AI 모델 개발 - 영상 복잡도 분석 알고리즘 개발 - 스트리밍 품질 개선 딥러닝 모델 개발
박은재	풀스택 개발 - 프론트 및 백엔드 서버 개발 실시간 스트리밍 서비스 구현
공통	보고서 작성 및 서비스 테스트

4 참고 문헌

(1) 정소영, 딥러닝을 이용한 실시간 인코딩 효율 최적화, <https://brunch.co.kr/@kakao-it/285>

(2) AWS, 라이브 스트리밍에서 비디오 지연 시간, <https://aws.amazon.com/ko/media/tech/video-latency-in-live-streaming/>

(3) 양명섭 김기원 박승배. "가변 비트율을 통한 적응형 스트리밍 서비스 시스템의 설계." 초고속정보화센터논문집 2.- (2001): 27-33., <https://www.riss.kr/link?id=A3009406>

(4) 이중영. "가변 비트율을 이용한 적응형 스트리밍 서비스 시스템 설계 및 구현." 국내석사학위논문 全北大學校 大學院, 2004. 전북특별자치도, <https://www.riss.kr/link?id=T9349742>