

FlexiView: 엣지 지원 모바일 비디오 스트리밍을 위한 탄력적 비디오 품질 및 재생 속도 최적화

신동윤, 김영진*
인하대학교, *인하대학교

skc213@inha.edu, *yj.kim@inha.ac.kr

요약

모바일 비디오 스트리밍에 대한 수요 증가로 인해 변동하는 네트워크 조건에서 고품질의 비디오 전송 기술이 주목받고 있다. 본 논문에서는 엣지 지원 모바일 비디오 스트리밍 환경에서 데이터 크기, SSIM, VMAF 와 같은 청크 별 특성을 바탕으로 청크 단위의 비디오 품질과 재생 속도를 최적화하는 새로운 알고리즘인 FlexiView 를 소개한다. FlexiView 는 Lyapunov 및 Convex 최적화 기법을 사용하여 사용자 QoE 를 극대화하고 영상 끊김을 최소화하며, 비디오 품질과 재생 속도 변동성을 안정화 한다. 실측 트레이스 기반 시뮬레이션을 통해 FlexiView 는 기존의 적응형 비트레이트 및 재생 제어 알고리즘보다 뛰어난 성능을 입증하였으며, 이는 QoE 개선, 영상 끊김 방지, 시스템 일관성 유지 사이의 tradeoff 균형을 효과적으로 맞추므로써 달성된다.

I. 서론

모바일 비디오 스트리밍 시장의 급성장으로 비디오 전송 최적화가 중요한 과제가 되고 있다. 사용자들은 고품질 비디오 재생을 기대하지만, 모바일 네트워크의 대역폭 변동으로 인해 시청 경험이 악화될 수 있다. 이 문제를 해결하기 위해 비디오 품질과 재생 속도를 효율적으로 제어하는 연구가 많이 진행되고 있다.

비디오 콘텐츠는 여러 품질(예: 25Mbps, 7Mbps, 2Mbps)로 트랜스코딩되고, 각 품질 버전은 동일한 재생 시간(예: 4 초)으로 나뉜 청크로 제공된다. 이때 비디오 품질은 각 청크에 대하여 사용자가 느끼는 시각적 품질을 결정하며, 높은 품질은 더 나은 사용자 QoE 를 제공하지만 더 많은 네트워크 자원을 필요로 한다. 반면, 재생속도 제어는 비디오의 재생 속도를 조절하여 몰입감에 영향을 주며, 1 배속을 기준으로 낮아질수록 사용자 QoE 가 떨어지지만 재생 대기열(버퍼)의 소진 속도를 줄여준다. 이러한 두 요소들은 서로 밀접하게 연결되어 있으며, 특히 동적 네트워크 환경에서 두 요소의 균형을 맞추는 것이 어렵다.

전통적인 적응형 비트레이트(ABR) 알고리즘은 네트워크 상태와 버퍼 길이에 따라 비디오 품질을 조정하며 영상 끊김을 방지하는 데 중점을 둔다. 그러나 이 알고리즘은 재생 속도 제어를 고려하지 않는다. 재생 속도를 미세하게 조정함으로써 재생 대기열이 짧고 네트워크 상태가 좋지 않은 상황에서도 영상 끊김이나 급격한 비디오 품질 저하 현상을 줄일 수 있다. 비디오 품질과 재생 속도를 함께 최적화하면 더 높은 QoE 를 제공하고 영상 끊김을 줄이며 일관된 품질을 유지할 수 있다. 최근 딥러닝과 강화학습을 활용한 통합 제어 연구가 주목받고 있지만[1][2], 이들 연구는 청크 데이터 크기가 동일하다는 가정이나 청크 내 장면에 따른 QoE 민감도 변화를 충분히 고려하지 않는 한계가 있다.

청크 수준의 이질성을 평가하기 위해 SF, 애니메이션, 다큐멘터리 장르의 4K 해상도 및 VBR 인코딩된 비디오 세 가지를 다운로드하였다. Handbrake 비디오 트랜스코더를 사용해 YouTube 권장 비디오 설정에 따라 원본 비디오를 여러 품질로 트랜스코딩한 후, FFmpeg 소프트웨어로 각 비디오를 4 초씩 청크로 분할하였다. 각

청크에 대해 데이터 크기, 인접한 프레임 간 평균 structural similarity index measure (SSIM), 최고 품질의 원본 청크와 비교한 video quality assessment algorithm (VMAF) 세 가지 지표를 측정하였다. 데이터 크기는 청크를 다운로드하는 데 필요한 전송 시간을 결정하며, SSIM 은 청크 내 장면의 역동성을 측정하여 재생 속도에 따른 사용자 QoE 와 높은 상관관계를 갖는다. VMAF 는 사용자가 실제로 체감하는 시각적 품질을 반영하며, 비디오 품질과 사용자 QoE 간의 강한 상관관계를 보여준다. 그림 1 에서는 동일한 비디오의 데이터 크기와 SSIM 이 청크 인덱스에 따라 차이를 확인할 수 있으며, 그림 2 는 동일한 비디오 품질로 트랜스코딩했음에도 불구하고 청크마다 사용자가 느끼는 품질이 다르다는 것을 나타낸다. 이러한 결과는 청크별 특성을 고려한 비디오 품질 및 재생 속도 제어의 필요성을 강조한다.

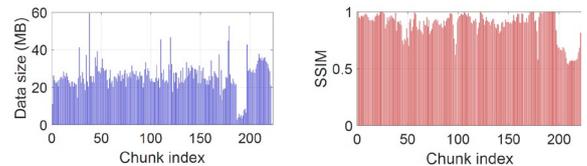


그림 1. Sintel (SF)의 높은 비디오 품질에 대한 청크 별 크기와 평균 SSIM.

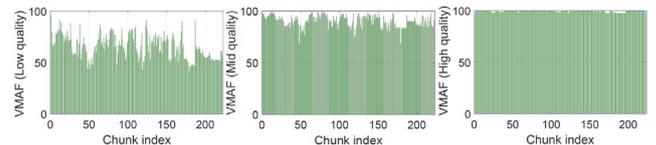


그림 2. Sintel (SF)의 서로 다른 비디오 품질에 대한 청크 별 VMAF.

본 논문에서는 사용자의 QoE 를 극대화하면서 영상 끊김을 방지하고 시스템의 일관성을 유지하는 탄력적 비디오 품질 및 재생 속도 제어 알고리즘인 FlexiView 를 제안한다. 가상 대기열 기반의 Lyapunov 및 Convex 최적화 기법을 사용하여 FlexiView 알고리즘을 개발하였으며, 새로운 청크의 전송이 시작될 때와 새로운 청크의 재생이 시작될 때 각각 비디오 품질과 재생 속도를 새롭게 업데이트 할 수 있다. 마지막으로, 트레이스 기반 시뮬레이션을 통해 FlexiView가 기존 알고리즘보다 사용자 QoE, 영상 끊김 등에서 효과적임을 입증한다.

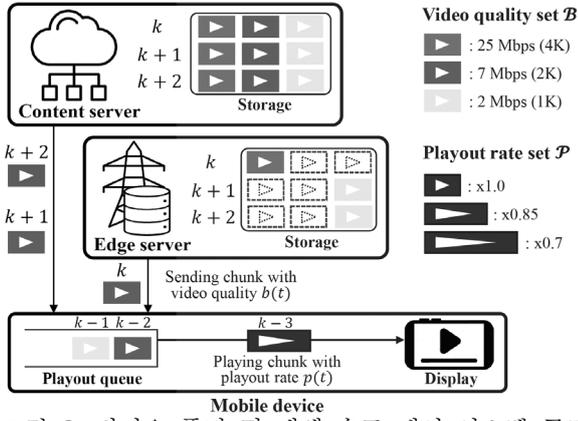


그림 3. 비디오 품질 및 재생 속도 제어 시스템 구조.

II. 본론

시스템 모델. 그림 3은 본 논문에서 고려하는 시스템 아키텍처를 보여준다. 사용자와 멀리 위치한 콘텐츠 서버는 비디오의 모든 비디오 품질에 대한 청크를 보유하고 있으며, 모바일 사용자와 가까운 위치에 있는 엣지 서버는 제한된 저장 용량으로 인해 일부 청크만 캐싱한 상황이다. 사용자는 모바일 비디오 스트리밍 서비스를 이용하고 있으며, 대상 비디오는 사전에 여러 품질로 트랜스코딩되어 $b \in \mathcal{B} = \{b_1, \dots, b_B\}$ 범위의 비디오 품질로 나누어지고, 각 품질은 $k \in \mathcal{K} = \{0, \dots, K-1\}$ 로 인덱싱된 d 초 단위 청크로 분할 되어있다. 청크 (b, k) 의 캐싱 상태는 $\theta(b, k) \in \{0, 1\}$ 로 표시되며, 청크 (b, k) 의 전송 가능한 속도는 네트워크 상태와 캐싱 상태에 의해 결정된다. 사용자의 QoE를 나타내는 $U(t)$ 는 비디오 품질과 재생 속도와 관련하여 다음과 같이 정의된다.

$$U(t) = U_b(b(t), k_b(t)) + U_p(p(t), k_p(t)),$$

여기서 $k_b(t) \in \mathcal{K}$ 와 $k_p(t) \in \mathcal{K}$ 는 t 시점에서 전송 중이거나 재생 중인 청크의 인덱스를 나타내고, $b(t) \in \mathcal{B}$ 와 $p(t) \in \mathcal{P} = [p_{\min}, 1]$ 는 *FlexiView* 알고리즘에 의해 선택된 비디오 품질과 재생 속도를 나타낸다. 중요한 점은 사용자 QoE는 동일한 $(b(t), p(t))$ 일지라도 전송 중이거나 재생 중인 청크가 무엇인지에 따라 달라진다는 점이다. 이는 그림 1과 그림 2에서 SSIM과 VMAF를 측정했을 때 확인할 수 있다.

이 모델을 바탕으로, 모바일 비디오 스트리밍의 사용자 QoE를 최대화하면서 시스템의 일관성을 유지하고 영상 끊김을 방지하는 최적화 문제를 다음과 같이 정의한다.

$$(P): \max_{b, p} \bar{U},$$

s.t.

- (C1): $b(t) \in \mathcal{B}$, $p(t) \in \mathcal{P}$ 는 청크 단위로 변경될 수 있다,
- (C2): 재생 대기열이 비어 있으면 영상 끊김이 발생한다 ($p(t) = 0$),
- (C3): 평균 변동 $\overline{\Delta b}$ 와 $\overline{\Delta p}$ 을 임계값 이하로 유지한다,
- (C4): 재생 대기열을 안정화 시킨다.

알고리즘 제안. *FlexiView* 알고리즘은 물리적 및 가상 대기열 기반의 Lyapunov 최적화 기법과 Convex 최적화를 바탕으로 실시간에서 최적의 $(b(t)^*, p(t)^*)$ 를 찾기 위해 제안되었다. 이 알고리즘은 세 가지 주요 상충 요소 간의 균형을 맞추며 작동한다: (i) 사용자 QoE를 향상시키기 위해 $b(t)$ 와 $p(t)$ 를 증가시키고, (ii) 영상 끊김을 피하기 위해 $b(t)$ 와 $p(t)$ 를 감소시키고, (iii) 시스템의 일관성을 유지하기 위해 $b(t)$ 와 $p(t)$ 를 이전

값으로 유지한다. 이 세 가지 요소 간의 상충 요소를 고려하여 *FlexiView*는 각 시점에서 사용자 경험을 극대화하면서도 시스템의 안정성을 유지하고 영상 끊김을 최소화하는 방향으로 동작한다.

시뮬레이션 결과. 트레이스 기반 시뮬레이션을 통해 *FlexiView* 알고리즘의 성능을 최신 알고리즘들과 비교 평가하였다. 대전에서 측정된 LTE 및 5G 네트워크 트레이스를 활용하며, 세 가지 4K 비디오 콘텐츠로 진행되었다. *FlexiView*는 기존 알고리즘과 비교하여 비디오 품질과 재생 속도를 개선하면서 영상 끊김을 줄이는 성능을 보였다. 특히 "*FlexiView w/o stab*"과 비교했을 때 시스템 일관성을 유지하면서도 성능이 개선되었으며, 네트워크 변동성이 심한 상황에서도 안정적인 성능을 제공하였다. "*Quality fixed(high)*"는 비디오 중단이 너무 많아 실제 환경에 적용하기 어려우며, *FlexiView*는 사용자 QoE와 시스템 일관성을 극대화하는 데 있어 우수함을 입증하였다.

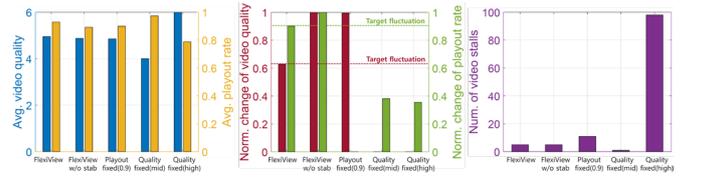


그림 4. *FlexiView*와 비교 알고리즘의 시뮬레이션 결과.

III. 결론

본 연구에서는 엣지 서버가 지원하는 모바일 비디오 스트리밍의 사용자 경험 품질(QoE)을 향상시키기 위해 비디오 품질과 재생 속도를 공동으로 최적화하는 알고리즘을 개발하였다. 특히, *FlexiView*는 청크 수준의 데이터 크기, SSIM, VMAF와 같은 특성에 반응하여 모바일 사용자가 느끼는 QoE를 향상시킨다. 심층적인 시뮬레이션을 통해 제안된 *FlexiView* 알고리즘이 사용자 QoE를 향상시키는 것, 영상 끊김을 방지하는 것, 그리고 시스템의 일관성을 유지하는 세 가지 상충 요소 사이에서 최적의 균형을 찾아냄을 입증하였다.

ACKNOWLEDGMENT

본 연구는 2024년도 정보통신기획평가원 (No.RS-2022-00155915 인공지능융합혁신인재양성(인하대학교), 2021-0-02201 사용자 프라이버시를 보존하는 비디오 캐싱을 위한 연합 학습 시스템, 2022-0-00448 인간처럼 회상이 가능한 인공 신경망 지속학습 플랫폼 개발, No. RS-2024-00398157 AI-Native 응용서비스 지원 AI 오케스트레이터 개발) 및 한국연구재단 (No. RS-2023-00240019)의 지원을 받아 수행된 연구임.

참고 문헌

- [1] P. K. Yadav et al., "Playing chunk-transferred DASH segments at low latency with QLive," Proc. of ACM MMSys, p. 51-64, (2021).
- [2] P. K. Mu et al., "AMIS: Edge computing based adaptive mobile video streaming," Proc. of IEEE INFOCOM, p. 1-10, (2021).