L3VTP: A Low-Latency Live Video Transmission Platform

Gang Yi
Tsinghua University
icarus.eagle@outlook.com

Dan Yang
Beijing University of Posts and Telecommunications
bupt-steven@foxmail.com

Mowei Wang
Tsinghua University
wang.mowei@outlook.com

Weihua Li
PowerInfo
powerinfo@263.net

Yi Li
PowerInfo
tiger_li@263.net

Yong Cui
Tsinghua University
cuiyong@tsinghua.edu.cn

CCS CONCEPTS

• **Information systems → Multimedia streaming**; • **Computer systems organization → Real-time system architecture**;

KEYWORDS

live video streaming, ABR, QoE, low latency

ACM Reference Format:
Gang Yi, Dan Yang, Mowei Wang, Weihua Li, Yi Li, Yong Cui. 2019. L3VTP: A Low-Latency Live Video Transmission Platform. In *SIGCOMM '19: ACM SIGCOMM 2019 Conference (SIGCOMM Posters and Demos '19), August 19–23, 2019, Beijing, China.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3342280.3342336

1 INTRODUCTION

Recently, a new breed of video services that support interactive live video streaming has become tremendously popular. Compared to the video-on-demand (VoD) cases, live video streaming requires a low end-to-end latency for real-time interaction between the broadcasters and the viewers while still maintaining low rebuffering ratio and high video quality. To ensure viewer's high quality of experience (QoE), adaptive bitrate (ABR) algorithms are leveraged to dynamically decide the bitrate level for future video content.

The existing video streaming system and ABR algorithms are inherently mismatch the properties of live video scenario. On one hand, the playout can not start until the entire video chunk has been downloaded in http-based adaptive streaming (DASH). It results in the end-to-end delay longer than one-chunk length, which is unacceptable in live video transmission. On the other hand, unlike the case for pre-record on-demand video, live video is generated in real time. Therefore, the ABR algorithm can only access few seconds of video on CDN and less information can be utilized to make the optimal bitrate decisions.

To address these challenges and promote the development of the community, we design and provide a **L**ow-**L**atency **L**ive **V**ideo **T**ransmission **P**latform (L3VTP) that can help to speed up the live ABR research loop. To the best of our knowledge, L3VTP (§2) is the first platform to accomplish that. As shown in figure 1, L3VTP contains five components: ABR algorithm, live system, simulator, dataset and QoE evaluation. Researches can evaluate their algorithms both on the simulator and the live system. Moreover, we held AITrans [1], a global ABR competition based on L3VTP. Through the competition experience, we summarize the pros and cons of the submitted ABR algorithms and verify the fidelity of our simulator. All the resources of L3VTP are open source [2].

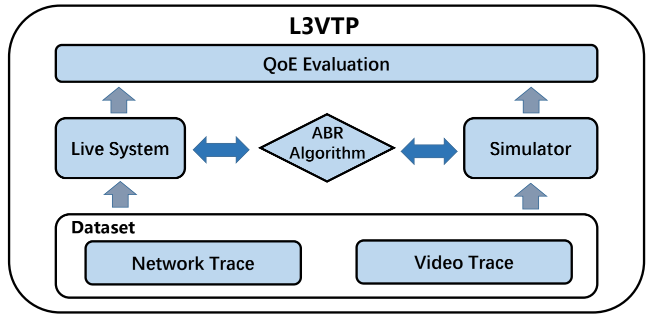

Figure 1: Architecture of L3VTP.

2 L3VTP DESIGN

2.1 System Design

To ensure low latency, we refine the video transmission granularity to frame and introduce two delay control mechanisms.

Frame-level push-based delivery: The typical workflow of a personalized live streaming system can be described as follows. First, the broadcaster side captures and pushes the live video to the transcoding server. Then the video will be transcoded to different bitrate levels and all these streams will be pushed to the CDN server. Finally, the client will request each video segment from the CDN server according to the decision of the ABR algorithm.

In our system, unlike the chunk-level pull-based DASH, CDN server pushes video frames to the client. The ABR algorithm makes the bitrate decision periodically according to the current state (e.g. measured throughput and buffer occupancy). Once the frames of the decided bitrate arrive at the CDN server, they can be pushed immediately to the client and playout frame by frame. Thus the latency of requesting each frame (e.g. half of RTT) and waiting the entire chunk downloaded for playout can be reduced. Moreover,

[1] https://www.aitrans.online/
[2] https://github.com/L3VTP/L3VTP

SIGCOMM Posters and Demos '19, August 19–23, 2019, Beijing, China

ACM ISBN 978-1-4503-6886-5/19/08…$15.00
https://doi.org/10.1145/3342280.3342336

the client will not face the risk that the video requested have not arrived at CDN so that the re-request latency can be avoided.

Latency control mechanisms: The latency in live video streaming is mainly caused by the playout buffer and the CDN buffer. Given a natural playout speed(1sec/sec), the latency will be accumulated when rebuffering occurs.Therefore we introduce two latency control mechanisms into the client video player, which provide the knobs to control the latency.

(1) **Adaptive playout.** The adaptive playout machanism[3] controls the video playout speed according to a triple $[S, T, Q]$ which can be represented as the *target buffer* T. In the triple, S represents the slow playout threshold ($S = \alpha * T$), and Q represents the fast playout threshold ($Q = \beta * T$). If the buffer level is greater than Q or less than S, the player will perform fast or slow playout respectively[3]. Moreover, the slow playout can help to reduce the risk of rebuffering by extend the playout time of the video frame in buffer.

(2) **Frame skipping.** The client downloadeds frame in order under default setting. The frame skipping event will be trigger when the end-to-end delay exceeds the predefined latency limit. In this moment, the video frame downloaded by the client is no longer the next frame, but the next I-frame.

2.2 Simulator

It is often costly and unscalable to evaluate the algorithm in real-world systems. Therefore, we build a simulator according to our live system, which decouples the operation environment and the control algorithm. It simulates the interaction between the client video player and the CDN server, which includes the CDN push, the client download and playback. The simulator takes the video trace and network trace as inputs to simulate the dynamics of video sources and the last-hop network bandwidth fluctuation. The bitrate and latency control logic is decided by the ABR algorithm.

2.3 ABR Algorithm

The ABR algorithm interacts with Live System or Simulator. It takes the observations from the environment as inputs, which include playout buffer occupancy, download time for every frame, frame size, rebuffering times, end-to-end delay, and so on. According to these states, the ABR algorithm decides on the bitrate for next chunk and the value of target buffer which controls the adaptive playout mechanism. Note that the decision can only take effect when it comes to the GoP(group of pictures) boundary.

2.4 Dataset

The dataset used in L3VTP consists of two parts, the video trace and the network trace. The video trace describes the video frame size, the frame type (i.e. I/P-frame) and the timestamp that the frame arrives at the CDN server. The timestamps reflects the network condition between the broadcaster side and the CDN server. The network trace records the network condition between CDN server and the client. We collect the video traces and the network traces daily from December 25th 2018 to January 8th 2019 at a CDN server in Japan, which is operated by a commercial live video company. The video traces contain video data of three scenarios including game, live show and sports.

[3] According to the experience of a commercial live streaming company, we set the speed of adaptive playout to 0.95x and 1.05x, with which the viewer are often cannot notice the change of playout speed.

2.5 QoE Evaluation

In our L3VTP, the ABR algorithm is expected to jointly decide the bitrate level and the target buffer to optimize a predefined quality of experience (QoE) metric. Based on the QoE model of pensieve [2], we construct our QoE metric with latency penalty as:

$$QoE = \sum_{n=1}^{N} \sum_{m=1}^{M} (\beta R_{n,m} - \gamma T_{n,m} - \delta L_{n,m}) - \sum_{n=1}^{N-1} \alpha |R_{n+1} - R_n|$$

for a live video with N GoPs and M frames in each GoP. $R_{n,m}$,$T_{n,m}$ and $L_{n,m}$ represent the bitrate, rebuffering time and latency of frame m in GoP n respectively. The coefficients can be set according to the application scenario and the user preference.

3 LIVE ABR ALGORITHM COMPETITION

To facilitate ABR algorithms development for live streaming, we held an ABR algorithm competition based on our platform. 138 teams in total from China, USA and Japan participated in this competition. The teams come from both academia and industry including Tsinghua, UCLA and Alibaba.

The contestants are asked to develop an ABR algorithm which will be tested with our simulator and system given network trace and video trace. Then performance of the ABR algorithms are evaluated by QoE model. The relative perfomance ranking of submitted ABR algorithms are almost consistent when tested with simulator and system, which shows the high fidelity of our simulator.

In our competition, most of the submitted algorithms are variants of the existing ABR algorithms. Their modifications for live streaming indicates that current algorithms can not be directly applied in the live scenario. Buffer-based algorithms (e.g. BBA [1]) lack of sufficient scheduling space due to the small playback buffer caused by the low-latency constraint. Objective-based algorithms (e.g. MPC [4] and Pensieve [2]) need the future chunk information as decision inputs. However, at one moment, only few seconds of video can be accessed ahead in CDN for live video.

By analyzing all the ABR algorithms and their rankings, we find two operations helpful to enhance ABR algorithms for live streaming: **Video Source Information Prediction** and **Network Condition Classification**. Since there is little information in CDN, it contributes to better decision to predict future source video information (e.g. the chunk size) based on the video information available. Accurate throughput prediction can dramatically improve the efficiency of bitrate adaptation for high QoE. Extracting the feature of network condition and classifying it into several categories can help to improve the accuracy of throughput prediction.

4 FUTURE WORK

Our ultimate goal is to improve the QoE of low-latency live video streaming. Our future work will include designing a customized ABR algorithm with deep reinforcement learning and a frame size predictor that can capture the dynamics of the video source.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive feedback.This work was funded in part by National Key R&D Program of China award No.2018YFB1800303 and National Nature Science Foundation of China(NSFC) award No.61872211.

REFERENCES

[1] Te-Yuan Huang, Ramesh Johari, Nick McKeown, Matthew Trunnell, and Mark Watson. 2015. A buffer-based approach to rate adaptation: Evidence from a large video streaming service. *SIGCOMM* 44, 4 (2015), 187–198.
[2] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. 2017. Neural adaptive video streaming with pensieve. In *SIGCOMM*. ACM, 197–210.
[3] Eckehard Steinbach, Niko Farber, and Bernd Girod. 2001. Adaptive playout for low latency video streaming. In *Proceedings 2001 International Conference on Image Processing*, Vol. 1. IEEE, 962–965.
[4] Xiaoqi Yin, Abhishek Jindal, Vyas Sekar, and Bruno Sinopoli. 2015. A control-theoretic approach for dynamic adaptive video streaming over HTTP. In *SIGCOMM*, Vol. 45. ACM, 325–338.