AutoTune: Game-based Adaptive Bitrate Streaming in P2P-Assisted Cloud-Based VoD Systems

Yuhua Lin and **Haiying Shen**

Dept. of Electrical and Computer Engineering

Clemson University, SC, USA

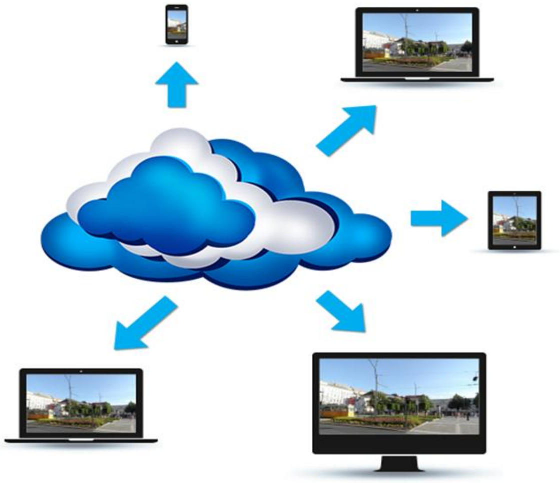
Outline

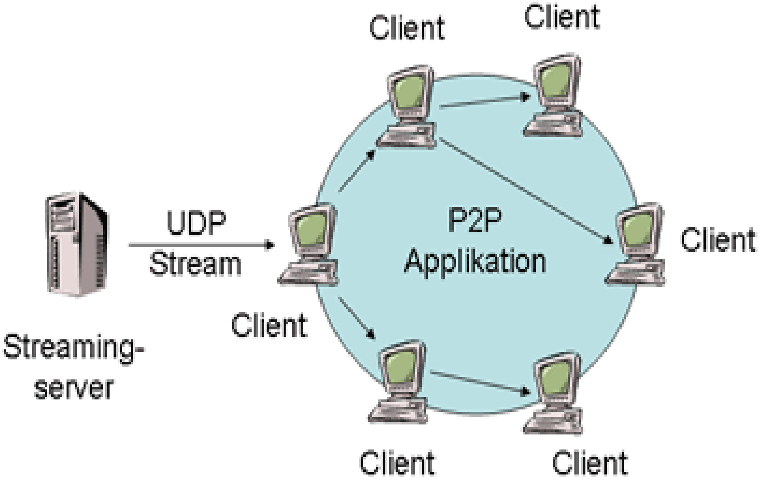
- Introduction
- System Design
    - Overview of AutoTune
    - Design of AutoTune
- Performance Evaluation
- Conclusions

Introduction

Cloud: stable and robust video streaming services

P2P video streaming: scalable, cheap



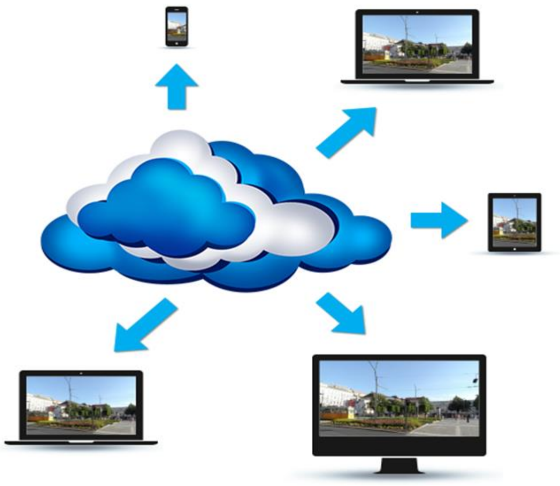


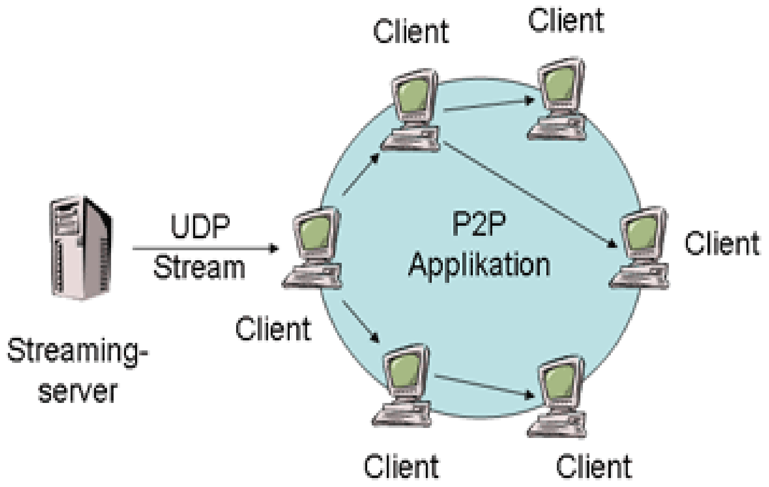
[1] http://ipcamlive.com/howdoesitwork

[2] http://www.csg.uzh.ch/publications/software/p2p-streaming.html

Introduction

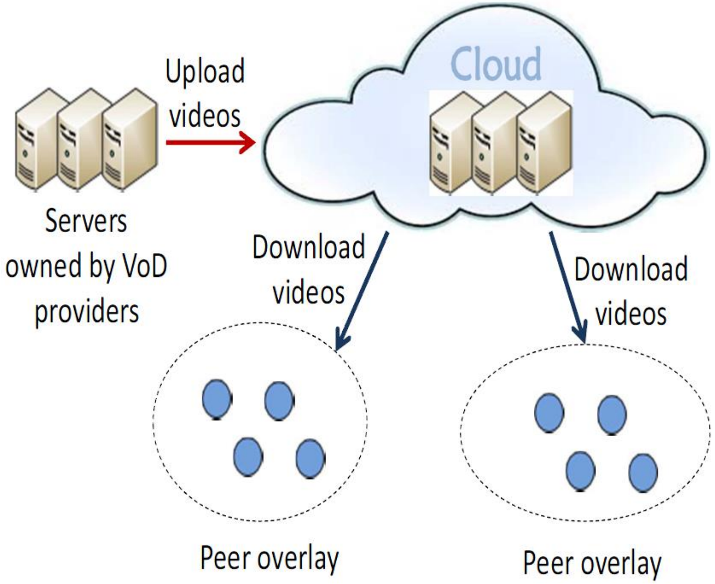
Hybrid P2P-assisted cloud-based video-on-demand systems (hybrid VoD)





Introduction

Hybrid VoD systems



Users:

- Watching same video are grouped in a P2P overlay
- Download video chunks from the cloud and peers

Introduction

Adaptive bitrate streaming in hybrid VoD systems

Purpose: improve video playback smoothness

Existing adaptive bitrate streaming methods:

Server-side adaptation

- Adjust user's video bitrate by examining bandwidth or buffer conditions of the server

Client-side adaptation

- Adjust video bitrate by estimating a user's bandwidth capacity based on the current level of its playback buffer

Introduction

Adaptive bitrate streaming in hybrid VoD systems

Drawbacks of existing adaptive bitrate methods:

Server-side adaptation

- Fails to guarantee user satisfaction, as it adapts a user’s video bitrate based on the server’s bandwidth capacity

Client-side adaptation

- User aims to maximize its own video bitrate based on its buffer condition, it leads to a large size of video downloads from the cloud

Introduction

Our proposed method: AutoTune

A game-based adaptive bitrate streaming method

Formulate the bitrate adaptation problem as a noncooperative Stackelberg game, where the VoD service provider and users are players

Reach the Stackelberg equilibrium, so that:

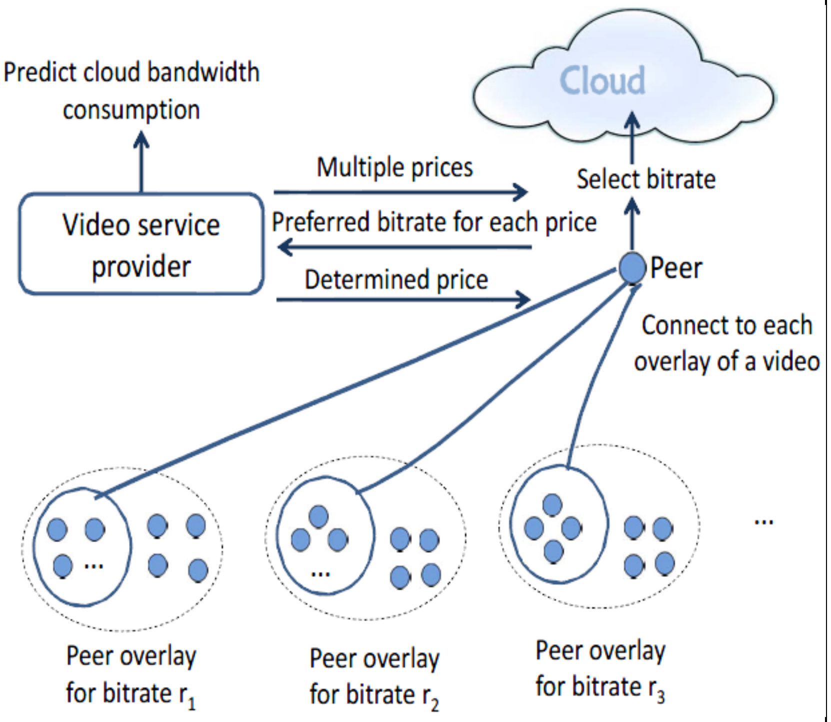
- Cloud bandwidth consumption is minimized
- Users are satisfied with the selected video bitrates

Outline

- Introduction
- System Design
  - Overview of AutoTune
  - Design of AutoTune
- Performance Evaluation
- Conclusions

Overview of AutoTune

1. VoD service provides a set of multiple unit prices for cloud bandwidth consumption
2. User chooses a new bitrate that maximizes its utility
3. VoD service provider chooses one unit price among multiple unit prices that maximizes its own revenue
4. Each user picks the bitrate corresponding to the price


Predict cloud bandwidth consumption
Cloud
Multiple prices
Select bitrate
Video service provider
Preferred bitrate for each price
Peer
Determined price
Connect to each overlay of a video
...
...
...
Peer overlay for bitrate r_1
Peer overlay for bitrate r_2
Peer overlay for bitrate r_3

Client Buffer Based Bitrate Adaptation

Decide a new set of possible bitrates a player can choose: PR_k

Increases bitrate when:

- Buffer has more than N_u^b sequential chunks to playback
- The last bitrate change was made more than t_l seconds ago

$$PR_k = \{r_j : r_j \geq r_i \wedge \frac{r_j - r_i}{r_i} < \frac{N^b - N_u^b}{N_u^b}\}, if \begin{cases} N^b > N_u^b \\ t_g > t_l \end{cases}$$

Client Buffer Based Bitrate Adaptation

Decide a new set of possible bitrates a player can choose: PR_k

Decreases bitrate when:

- Buffer has less than N_l^b sequential chunks to playback
- The last bitrate change was made more than t_l seconds ago

$$PR_k = \{r_j : r_j \leq r_1 \wedge \frac{r_i - r_j}{r_i} < \frac{N_l^b - N^b}{N_l^b}\}, if \begin{cases} N^b < N_l^b \\ t_g > t_l \end{cases}$$

Price Driven Bitrate Adaptation

Utility function of a user

$$F(k)(r_i, \alpha_i, s_i, p, r_u) = U_s(r_i, \alpha_i, s_i) - w_k U_p(r_i, r_u)$$

$U_s(\cdot)$: a user's satisfaction degree in watching a video of a specific bitrate

$U_p(\cdot)$: payment cost function on cloud bandwidth consumption

Price Driven Bitrate Adaptation

$U_s(\cdot)$: user's satisfaction degree

- Non-decreasing as higher bitrate makes a user more satisfied
- Marginal satisfaction is non-increasing as a user's level of satisfaction gradually gets saturated when video bitrate increases

$$U_s(r_i, \alpha_i, s_i) = \alpha_i \ln(1 + s_i r_i)$$

r_i : video bitrate
α_i : scale factor
s_i : satisfaction parameter

Price Driven Bitrate Adaptation

$U_p(\cdot)$: payment cost function

$$U_p = p(r_i - r_u)$$

r_u: bandwidth contribution from peers

Rationale: utility of a user decreases with a higher price

Combine all together, utility function of a user:

$$F(k)(r_i, \alpha_i, s_i, p) = \alpha_i \ln(1+s_i r_i) - w_k p(r_i - r_u)$$

Price Driven Bitrate Adaptation

Utility function of the VoD service provider

$$L(p) = p \sum_{n} (r_i - r_u)$$

Rationale:

VoD service provider aims to maximize its revenue, i.e., unit price times cloud bandwidth usage from all users

Optimal Bitrate Selection

1. Leader: VoD service provider notifies users a set of unit prices for estimated cloud bandwidth

$$V \; =< \; p_1, p_2, ..., p_m \; >$$

2. Follower: each user calculates optimal bitrate for each price that maximizes its utility F(k)

$$r_{ij} = \underset{r_i \in RP_k}{\operatorname{argmax}} \; F(k)(r_i, \alpha_i, s_i, p_j)$$

3. Leader: VoD service provider sets a price that maximizes its utility $p_l = \underset{p_j \in P}{\operatorname{argmax}} \; L(p_j) = \underset{p_j \in P}{\operatorname{argmax}} \; p_j \sum_n r_{ij}$

4. Follower: picks its optimal bitrate corresponding to p_l

Outline

- Introduction
- System Design
  - Overview of AutoTune
  - Design of AutoTune
- Performance Evaluation
- Conclusions

Performance Evaluation: Settings

PeerSim simulator and PlanetLab real-world testbed

- 10,000 nodes on PeerSim， 350 nodes on PlanetLab
- 10 cloud servers on PeerSim, 1 cloud server on PlanetLab
- 1,000 videos from 100Kbps to 3600Kbps
- Nodes join the system following the Poison distribution with rate of 5 players per second

Comparison methods

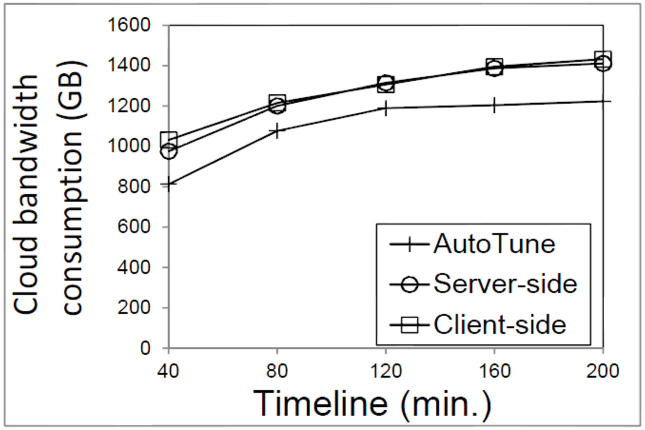
- Server-side bitrate adaptation [3]
- Client-side bitrate adaptation [4]

[3] A. Mansy and M. Ammar. Analysis of adaptive streaming for hybrid CDN/P2P live video systems. In Proc. of ICNP, 2011.

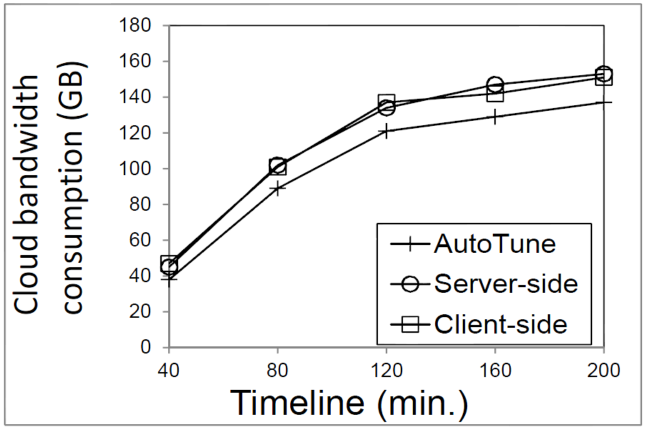
[4] K. Hwang, V. Gopalakrishnan, R. Jana, S. Lee, V. Misra, K. Ramakrishnan, and D. Rubenstein. Joint-family: Enabling adaptive bitrate streaming in peer-to-peer videoon-demand. In Proc. of ICNP, 2013.

Performance Evaluation: Results

- Cloud bandwidth consumption



Experimental results on PeerSim

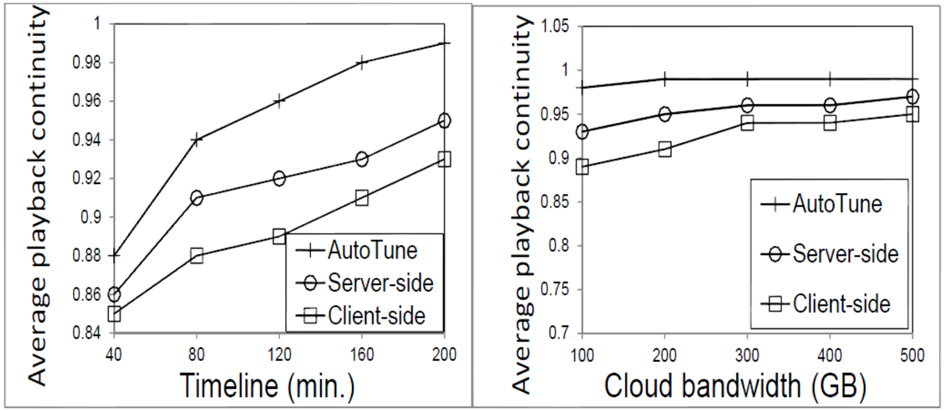


Experimental results on PlanetLab

- Observation: Server-side ≈ Client-side > AutoTune
- Reason: In AutoTune, VoD service provider encourages users to download chunks from peers by setting price on cloud bandwidth consumption; users minimize their cloud bandwidth consumption to increase the utility

Performance Evaluation: Results

- Video playback continuity: results from PeerSim
  - dividing the number of time slots without playback interruptions by the total number of slots

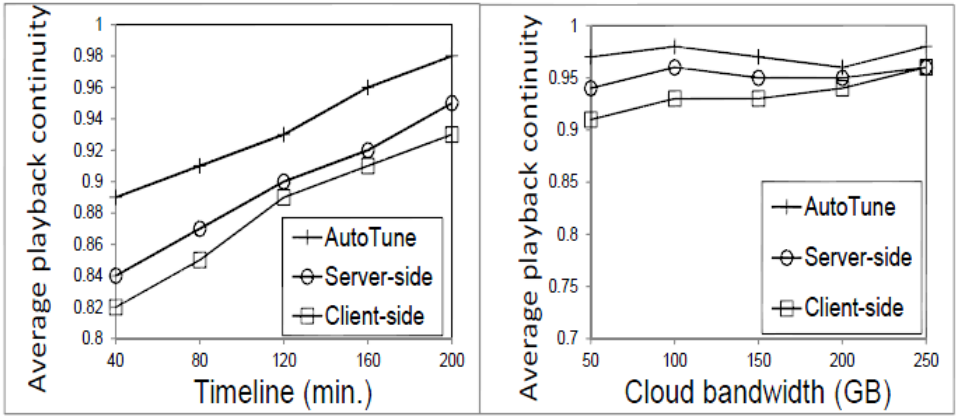


(a) Results at different time intervals.(b) Results at different cloud bandwidth consumption.

- Observation: AutoTune > Server-side > Client-side
- Reason: AutoTune achieves a tradeoff between minimizing cloud bandwidth consumption and guaranteeing users’ satisfaction; Server-side rejects bitrate increase requests when it has insufficient cloud bandwidth capacity

Performance Evaluation: Results

- Video playback continuity: results from PlanetLab
  - dividing the number of time slots without playback interruptions by the total number of slots

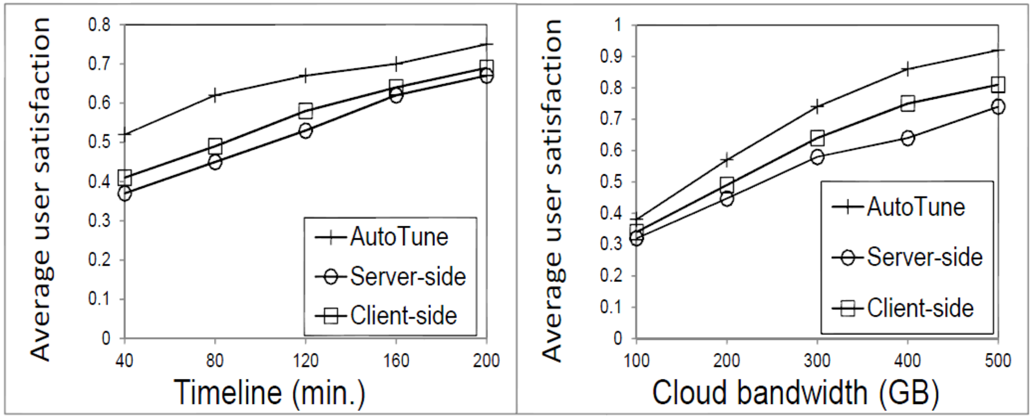


(a) Results at different time intervals. (b) Results at different cloud bandwidth consumption.

- Observation: AutoTune > Server-side > Client-side
- Reason: AutoTune achieves a tradeoff between minimizing cloud bandwidth consumption and guaranteeing users' satisfaction; Server-side rejects bitrate increase requests when insufficient cloud bandwidth capacity

Performance Evaluation: Results

- User satisfaction: results from PeerSim
  - $ln(1+r_i)/ln(1+3600)$

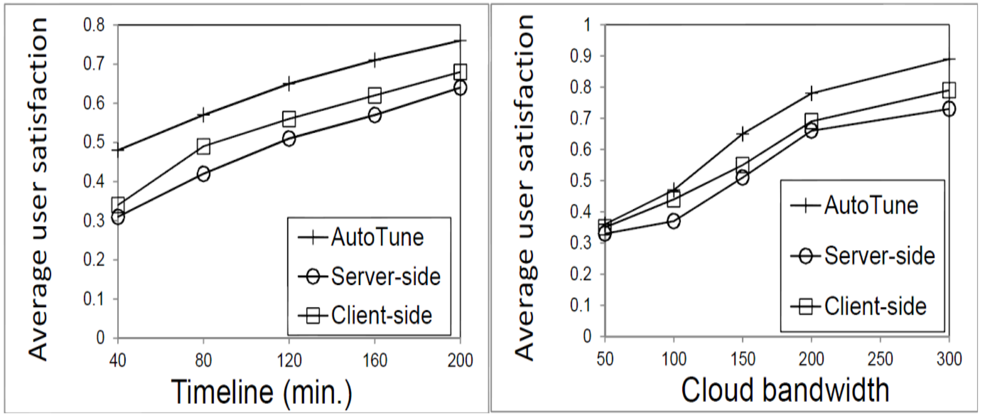


(a) Results at different time intervals.(b) Results at different cloud bandwidth consumption.

- Observation: AutoTune > Client-side > Server-side
- Reason: In AutoTune, each user selects a new video bitrate that guarantees its satisfaction and has high peer bandwidth contribution

Performance Evaluation: Results

- User satisfaction: results from PlanetLab
  - $ln(1 + r_i)/ln(1 + 3600)$



(a) Results at different time intervals.(b) Results at different cloud bandwidth consumption.

- Observation: AutoTune > Client-side > Server-side
- Reason: In AutoTune, each user selects a new video bitrate that guarantees its satisfaction and has high peer bandwidth contribution

Outline

- Introduction
- System Design
  - Overview of AutoTune
  - Design of AutoTune
- Performance Evaluation
- Conclusions

Conclusion

- AutoTune: game-based adaptive bitrate streaming method
- Experiments on the PeerSim simulator and the PlanetLab real-world testbed show the effectiveness of AutoTune:
  - Reduce cloud bandwidth consumption
  - Increase video playback continuity
  - Increase user satisfaction
- Future work: encourage peers to contribute bandwidth through incentives of better cloud service

Thank you!

Questions & Comments?

Haiying Shen

shenh@clemson.edu

Electrical and Computer Engineering

Clemson University