P2TSS: TIME-SHIFTED AND LIVE STREAMING OF VIDEO IN PEER-TO-PEER SYSTEMS

June 25, 2008

Sachin Deshpande
Jeonghun Noh*

Sharp Laboratories of America
Stanford University*
Motivation

• In live-stream multicast, e.g., live TV, viewers watch the most recent scenes together.

• Viewers may come late, yet want to watch video from the beginning.

• Most P2P streaming systems support live streaming only using peers’ uplink bandwidth.

• Peers can contribute:
  – Bandwidth
  – *Memory (storage)*
Lessons from Prior Arts

- Linear video access

- Use of peer-side storage
  - P2VoD requires big video buffer
  - P2Cast does not fully use peer buffer
  - Both impose limitation on peer storage availability

- Use of peer uplink capacity
  - P2VoD requires two connections on both uplink/downlink bandwidth

- Scalability issue
  - P2VoD requires help from a central server
Introduction to P2TSS

- P2TSS
  - P2P system serving both *live streaming* and *time-shifted streaming*
  - Time-shifted stream (TSS) is the same as the original stream except being delayed in time.
- Peers store a *portion of a video stream*.
- Cached video blocks are served to other peers for playback at a later time. (Similarity with P2P based Video-on-Demand)

Peer 1 caches Video [0m, 4m) from 9:00am

Source peer

Peer 2 watches from 0m at 9:10am
Caching Video

- Distributed caching
  - Peers locally determine *which video portion to cache.*
  - Peers stop caching video once their cache is full.

- Distributed Stream Cache (DSC)
  - Peer’s local buffer to hold a fraction of video
  - Independent of playback buffer
  - A finite size of cache (e.g., size of 2 to 4 minutes of video)

- Static contents in cache
  - no content change once the cache is full
  - An assumption that eases our analysis
  - Provides a bottom line performance

**Proposed Caching Algorithms**

1) Initial play-out position (IPP)
2) Live stream position (LSP)
IPP: Initial Play-out Position

- Peers cache from their initial play-out position
LSP: Live Stream Position

- Peers cache live stream with *another* video connection

![Diagram showing LSP: Live Stream Position with peers and their cached portions and playback trajectories.](image)
Setup Parameters

- Video session length: 7200s
- Peer uplink bandwidth: 3R (R is the video bitrate)
- Peer DSC size: 240s (common to all peers)
- 300 peers arrive at the system with inter-arrival exponentially distributed.
- Upon arrival at time T, a peer chooses its initial playout position x, randomly in $0 < x \leq T$.

Questions to be Answered

1) Distribution of data availability
2) Expected number of available peers
Analysis Sketch

- Distribution of data availability
  \[ \Pr\{x \text{ is available at time } t\} = \Pr\{\text{At least a peer covers } x \text{ by time } t\} \]

- Expected number of available peers
  \[ E\{M(t, x)\} = \sum_{k=1}^\infty \sum_{i=k}^\infty \Pr\{k \text{ of } i \text{ peers cover } x \mid N(t) = i\} \Pr\{N(t) = i\} \]
Model: Data Availability

No Peer Departure

Peer Departure
Expected Number of Available Peers

(No peer departure)

IPP

LSP

Model

At time 7200s

Simulation

Model

Model

At time 7200s
Searching Supplier Peers

• Peers *need to find* suppliers for continuous video playback

• Need for supplier switching
  – DSC size is finite
  – Current supplier may disconnect from the system

• Centralized search
  – **The (query) server** retains all the peers’ DSC status
  – Peers send queries to the server
Supplier Switching Overhead

- Trade-off between switching overhead and server stress (LSP is used)
  - Larger DSC reduces the number of supplier switching
  - Larger DSC results in longer cache filling time
P2VoD and P2TSS

Server Stress vs. Peer Bandwidth

- P2VoD [T. Do et al, 2004]
  - groups peers based on their arrival time.
  - Cache contents change over time. (time-sliding cache)
- P2TSS utilizes peers’ uplink bandwidth efficiently.
P2VoD and P2TSS

Server Stress vs. Group Size

- When the group size increases, the overhead for additional connection to fill DSC becomes significant.
- This overhead can be reduced by decreasing DSC size without hurting server stress.
Conclusions and Future Research

• Two distributed caching algorithms
  – We proposed caching algorithms with *light-weight control overhead*
    • Peers should easily know which portion to cache
    • as bottom-line performance algorithms
  – We provided the *analysis of video coverage characteristics* of the algorithms

• Future research topics
  – Distributed search for video blocks (*submitted to ISM ’08*)
  – Dynamic cache algorithm
  – Packet-level optimization