Impact of Quality Adaptation in SVC-based P2P Video-on-Demand Systems

Osama Abboud, Thomas Zinner, Konstantin Pussep, Sabah Al-Sabea, and Ralf Steinmetz

Carsten Griwodz
Osama Abboud


© 2010 author(s) of these slides including research results from the KOM research network and TU Darmstadt. Otherwise it is specified at the respective slide

February 22, 2011
Introduction

Video is a major part of Internet traffic [1]
- By 2014 almost 90% of Internet traffic

Peer-assisted solution
- Reduce server load by making use of client-side resources
- System is more scalable

But how to overcome weak peer contributions and heterogeneity?
- Scalable Video Coding (SVC) can relieve and solve many issues and problems

---

This paper

... is about

- Designing and evaluating a P2P VoD system that uses SVC to overcome resource heterogeneity and weak peer contributions

... addresses the questions

- Does SVC really help in systems with heterogeneous resources?
- How to measure the quality of an SVC-based VoD system?
- How should the SVC layer selection algorithm be configured?
- How often should these algorithms be executed?
Outline

Introduction

Scalable Video Coding

The Quality Adaptive VoD System
  - Quality Adaptation Algorithms
  - Peer Selection
  - Connection Management
  - Block Selection

Evaluation Metrics and Setup

Evaluation Results

Conclusion
Scalable Video Coding (SVC)

Video file encoded only once but with different quality levels
- Can be requested independently

Enables quality adaptation
- Video quality adjustable according to static and dynamic resources

Scalability in 3 dimensions
- Temporal: Frames per second
- Spatial: Resolution of the picture
- Quality: Quantization levels, sharpness

Picture sources: http://www.hhi.fraunhofer.de/index.php?id=2767&L=1
The P2P Video-on-Demand System

Peer-assisted architecture
- Mesh-based pull approach
- Hybrid server/P2P solution
  - Servers with modest resources are deployed
  - Inject the initial content, guarantee QoS
  - Tracker with contact information of the peers

Video streaming
- Video divided into chunks (time domain)
- Chunks divided into blocks (SVC 3D cube)
Quality Adaptation Algorithms

Select SVC layer according to
- Peer resources and network dynamics
- Different strategy depending on the stage of the streaming session
Initial Quality Adaptation (IQA)

Determines stream-able SVC layers
- Based on static peer resources
- Invoked at the beginning of video playback

Goal
- Avoid long startup times
- Match resources at session start

Uses static peer resources

Stages
- Spatial adaptation
- Bit-rate adaptation
- Complexity adaptation

Peer resources
- Screen resolution
- Available bandwidth
- Processing power

Initial Quality Set ($QS_0$)

Spatial Adaptation

Bitrate Adaptation

Complexity Adaptation

Final Decision

$QS_S$

$QS_{S,B}$

$QS_{S,B,C}$

User Preference
Progressive Quality Adaptation (PQA)

Adapt to real time changes of the network
- Activated periodically (every $T$ seconds)
- Based on real time network information

Goal
- Predict possible stalls before they happen
- Avoid stalls by temporary switching the layer

Uses real time information

Stages
- Net-status adaptation
- Bitrate adaptation
- Complexity adaptation

Filtered Quality Set ($QS_S$)

Real Time Information
- Block availability
- Throughput

Peer resources
- Processing power

Net. Status Adaptation

Bitrate Adaptation

Complexity Adaptation

$QS_{S,A}$

$QS_{S,A,B}$

Final Decision

User Preference

Progressive Quality Adaptation
PQA Stages

Net-status Adaptation
- Only request layers that are available within local neighborhood
- Determine the highest supported SVC layer
- Avoid waiting for rare blocks by temporary switching the layer

Bit-rate Adaptation
- Adjust layer according to throughput of high priority set
- Throughput observed through the fullness of the high priority set
- Avoid buffer under run by switching down the layer when throughput is low
- Switch layer up in case throughput high enough

Complexity Adaptation
- Use models that estimate required processing speed for decoding each layer
Peer Selection and Neighbor Management

Tracker manages: active peers and layers they currently stream
- Important in order to connect correct providers and consumers

Peers advertise currently streamed SVC layers
- After successful IQA/PQA to the tracker and neighbors
- During connection establishment phase with neighbors
- Buffer maps are extended to support SVC

The mechanism is bi-directional
- Peers are eventually clustered according to their resources
- Seeders/caches support both weak and strong peers.
Connection Management

Two separated peer sets
- Sender peer set
- Receiver peer set

Sender peer set
- Rank peers
- Trace their contribution
- Drop bad ones

Receiver peer set
- Limit the number according to upload bandwidth
- Assign upload slots according to how urgent a request is.
Block Selection

High priority set
- Sliding buffer window, updated with playback position

Download task dispatching
- Parallel download from multiple peers
- Keep all peers as busy as possible

Priority calculation
- High priority set: use greedy approach
  - Chunks close to playback position and base layer get highest priority
- Low priority zone: use non-greedy approach
  - Download blocks “soon most wanted” by receiver set
Outline

Introduction

Scalable Video Coding

The Quality Adaptive VoD System
- Quality Adaptation Algorithms
- Peer Selection
- Connection Management
- Block Selection

Evaluation Metrics and Setup

Evaluation Results

Conclusion
Evaluation Metrics

Session Quality
- Relative playback delay

$$\text{Relative\_delay} = \frac{\text{Delay}_{\text{init}} + \sum_{i=1}^{n} \text{Stall}_i}{\text{Time}_{\text{playback}}}$$

Video Quality
- Number of layer changes
- Relative received layer

$$\text{Quality}_{rel}(d, t, q) = \frac{d + t + q}{D_{\text{init}} + T_{\text{init}} + Q_{\text{init}}}$$

- Less layer changes or higher relative received layer → Better user experience
Evaluation Setup

Simulation Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation duration</td>
<td>200 minutes</td>
</tr>
<tr>
<td>Number of peers</td>
<td>90</td>
</tr>
<tr>
<td>Peer arrival pattern</td>
<td>Exponential</td>
</tr>
<tr>
<td>Number of servers</td>
<td>4</td>
</tr>
<tr>
<td>Server upload capacity</td>
<td>6 Mbps</td>
</tr>
<tr>
<td>Play-out buffer size</td>
<td>7 seconds</td>
</tr>
<tr>
<td>Neighborhood size</td>
<td>10</td>
</tr>
<tr>
<td>Video length</td>
<td>5 minutes</td>
</tr>
</tbody>
</table>

Peers divided into three sets according to resources

<table>
<thead>
<tr>
<th></th>
<th>Set 1</th>
<th>Set 2</th>
<th>Set 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Screen size</td>
<td>176 × 144</td>
<td>352 × 288</td>
<td>704 × 576</td>
</tr>
<tr>
<td>Upload speed</td>
<td>128 Kbps</td>
<td>320 Kbps</td>
<td>800 Kbps</td>
</tr>
<tr>
<td>Download speed</td>
<td>256 Kbps</td>
<td>560 Kbps</td>
<td>1200 Kbps</td>
</tr>
</tbody>
</table>
Impact of Quality Adaptation: Session Quality

![Graph showing the impact of quality adaptation on session quality]
Impact of PQA Frequency: Visualization

The darker, the higher is the layer. White indicates a stall.

- **PQA every 5 seconds**
- **PQA every 30 seconds**
Impact of PQA Frequency on Session Quality

Relative Playback Delay

Evaluation with IQA and PQA
Evaluation with PQA but without IQA
Impact of PQA Frequency on Video Quality
Number of Layer Changes

Evaluation with IQA and PQA
Evaluation with PQA but without IQA

number of layer changes

PQA interval [s]
Impact of PQA Frequency on Video Quality
Relative Received Layer

- Evaluation with IQA and PQA
- Evaluation with PQA but without IQA
Evaluation Conclusions

IQA and PQA help in achieving
- Better session quality
- More homogeneous performance across heterogeneous peers

PQA invocation interval exhibits a performance trade-off

PQA interval $\uparrow$ Session quality $\downarrow$ Video quality $\uparrow$

Best PQA interval depends on application scenario and user expectation
Conclusions and Next Steps

P2P Video streaming is envisioned to have more importance
- SVC is needed to
  - Support high quality streaming
  - Achieve homogenous performance at heterogeneous peers

Advanced adaptation algorithms were developed
- To enable an efficient provisioning of resources
- Performance, tradeoffs, and impact of adaptation were explored

Possible optimizations
- Adaptive PQA interval
- Prediction-based layer selection
- Map session and SVC quality metrics to Quality-of-Experience metrics
That’s all folks
Thank you for your attention. Questions?
SVC: Block-based Quality Scalability

Three-dimensional scalability

SVC Cube-Model
- Each GOP is modeled by a 3D-cube
- Block-combinations form layers
- Base layer is the most important

Must consider
- Interdependencies of blocks
- Deadline of blocks
- User preference
Evaluation Scenario

SVC Video File

- Used traces from a real nature video clip with medium activity

<table>
<thead>
<tr>
<th>SVC layer (d,t,q)</th>
<th>Picture size</th>
<th>Frame rate (fps)</th>
<th>Partial Bit-rate (Kbps)</th>
<th>Total Bit-rate (Kbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0,0,0</td>
<td>176 × 144</td>
<td>3.75</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td>0,1,0</td>
<td>176 × 144</td>
<td>7.5</td>
<td>30</td>
<td>90</td>
</tr>
<tr>
<td>0,2,0</td>
<td>176 × 144</td>
<td>15</td>
<td>30</td>
<td>120</td>
</tr>
<tr>
<td>0,3,0</td>
<td>176 × 144</td>
<td>30</td>
<td>30</td>
<td>150</td>
</tr>
<tr>
<td>1,0,0</td>
<td>352 × 288</td>
<td>3.75</td>
<td>180</td>
<td>240</td>
</tr>
<tr>
<td>1,1,0</td>
<td>352 × 288</td>
<td>7.5</td>
<td>90</td>
<td>330</td>
</tr>
<tr>
<td>1,2,0</td>
<td>352 × 288</td>
<td>15</td>
<td>60</td>
<td>390</td>
</tr>
<tr>
<td>1,3,0</td>
<td>352 × 288</td>
<td>30</td>
<td>60</td>
<td>450</td>
</tr>
<tr>
<td>2,0,0</td>
<td>704 × 576</td>
<td>3.75</td>
<td>270</td>
<td>510</td>
</tr>
<tr>
<td>2,1,0</td>
<td>704 × 576</td>
<td>7.5</td>
<td>150</td>
<td>660</td>
</tr>
<tr>
<td>2,2,0</td>
<td>704 × 576</td>
<td>15</td>
<td>180</td>
<td>840</td>
</tr>
<tr>
<td>2,3,0</td>
<td>704 × 576</td>
<td>30</td>
<td>160</td>
<td>1000</td>
</tr>
</tbody>
</table>