Video-on-demand broadcasting protocols

Jukka Leveelahti
17.4.2002
Tik-111.590
Multimedia Communications
Motivation

- Watch any movie at home whenever you like
- MPEG-2 at least 4 MB per second
- Too expensive 😞
- Two ways to reduce costs
  - Proactive: broadcasting
  - Reactive: many approaches
**Terms**

- **segment**
  - chunk of video, \(n\) of these in right order make entire video

- **consumption rate**
  - Processing rate of video in STB
  - \(b\), unit of measure for VOD server bandwidth

- **slot**
  - Time for STB to consume a constant-sized segment

- **channel**
  - Each stream in VOD server
  - Does not need to be of bandwidth \(b\)
  - Each video can be distributed over several channels
Client requirements

- when channel bandwidth > \( b \)
- or STB listens to multiple channels
- \( \Rightarrow \) We need local storage

- Size of storage and number of channels are the two things to minimize with clever broadcasting protocols
Staggered broadcasting protocols

- Starting times for video are staggered evenly across certain n of channels
- Video starts at every \( \frac{D}{n} \) (\( D=\)duration) mins = phase offset
- Not efficient for server, to cut phase offset double means doubling bandwidth
- Minimal requirements for client
- Can handle interactive VOD
- Example Canal Digital KIOSK
Pyramid broadcasting protocols

- each video is $n$ segments, $S_1, \ldots, S_n$
- available bandwidth divided evenly to $n$ channels $C_1, \ldots, C_n$
- $i^{th}$ segment of each video broadcasted on channel $C_i$
- Size of segments grow geometrically using parameter $\lambda$
Pyramid broadcasting protocols

- Client waits for $S_1$ on channel $C_1$ and starts consuming to receive all the time, receiving $S_i$ must start before $S_{i-1}$ finishes.
- Client will never experience a break when
  - $\alpha = b'/m$, where $b'$ is bandwidth of each channel.
  - Typical $\alpha$ is 2.5

<table>
<thead>
<tr>
<th>channel 1</th>
<th>$S_1$</th>
<th>$S_1$</th>
<th>$S_1$</th>
<th>$S_1$</th>
<th>$S_1$</th>
<th>$S_1$</th>
<th>$S_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>channel 2</td>
<td>$S_2$</td>
<td>$S_2$</td>
<td>$S_2$</td>
<td>$S_2$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>channel 3</td>
<td>$S_3$</td>
<td>$S_3$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$S_3$</td>
</tr>
</tbody>
</table>
Pyramid broadcasting - performance

- more efficient than Staggered Broadcasting
- client waiting time decreases exponentially with bandwidth
- 2h video with 10^6 bandwidth per video => 12 mins vs. 2 mins
- client requirements are high
- clients have to listen > 1 channels at once
- bandwidth per channel is very high
- requires large storage size
- optimized versions followed..
Permutation-based Pyramid Protocol

- each channel divided into $p$ subchannels for each video
- starts of segments evenly staggered on subchannels
- client listens only one subchannel at a time
- need of storage down to third comparing to basic Pyramid Broadcasting
- cost: more bandwidth for same waiting times
Skyscraper Broadcasting Protocol

- replaces geometric series for determining amount of data on each channel
- each video divided into \( n \) equally sized segments
- number of consecutive segments to place on each channel determined by series
  \[
  \{ 1,2,2,5,5,12,12,25,25,52,52,\ldots \}
  \]
- Equals \( \frac{b}{2} \) of about 1.5
- Each channel requires only bandwidth \( b \), can use much more channels
- Width of channel is constrained, no need of storage to store the last (large) block in last channel
Skycraper Broadcasting Protocol

- 1998 improvements: dynamical scheduling of channels and more efficient segment-to-channel series
- In total, low transfer rates and storage needs while reducing also waiting times (found in Pyramid Broadcasting)
- Low transfer rate at client causes waste of bandwidth in server

- opposite approach to Skyscraper Broadcasting
- Series is \(\{1,2,4,6,8,16,32,64,\ldots\}\)
- Very low waiting times
- Clients receive all data from all channels at once, leads to high transfer rate and high need of storage (up to half of video length)

- Goal: broadcast segments infrequently while maintaining an even transfer rate to the client.
- Uses series like predecessors: \( \{1, 3, 5, 15, 25, 75, 125, \ldots\} \)
- Big difference: segments don’t need to be consecutive on channels.
- Uses pairs of channels when assigning segments.

<table>
<thead>
<tr>
<th>Slot</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel 1</td>
<td>( S_1 )</td>
<td>( S_1 )</td>
<td>( S_1 )</td>
<td>( S_1 )</td>
<td>( S_1 )</td>
<td>( S_1 )</td>
</tr>
<tr>
<td>Channel 2</td>
<td>( S_2 )</td>
<td>( S_4 )</td>
<td>( S_2 )</td>
<td>( S_5 )</td>
<td>( S_2 )</td>
<td>( S_4 )</td>
</tr>
<tr>
<td>Channel 3</td>
<td>( S_3 )</td>
<td>( S_6 )</td>
<td>( S_8 )</td>
<td>( S_3 )</td>
<td>( S_7 )</td>
<td>( S_9 )</td>
</tr>
</tbody>
</table>
Pagoda Broadcasting Protocol

- Client waits for instance of $S_1$ on channel $C_1$
- While consuming $S_1$, starts receiving from every other channel dedicated to that video
- Each segment $S_i$ is broadcasted at least once every $i$ slots of time
- Client will have the segment ahead in buffer or receive directly from server when needed
- Still requires storage for about half of video

Notice: Pyramid protocols don’t work with interactive VOD
Harmonic broadcasting protocols

- first Juhn & Tseng (1997)
- Each video divided into $n$ equally sized segments $S_1, \ldots, S_i$
- These are continuously broadcasted in their own channels
- $S_i$ is broadcasted in channel $C_i$ with bandwidth $b/i$
- Sum of channel bandwidths is

\[
\sum_{i=1}^{n} \frac{b}{i} = b \sum_{i=1}^{n} \frac{1}{i} = bH(n)
\]

- $H(n)$ is the harmonic number of $n$, hence the name
Harmonic broadcasting protocols

- series grows very slowly
- can use hundreds of segments without not much bandwidth
- example: with 5b 1,5 mins for 2h video
- local storage is needed about 37% of the video
- contains a bug
- fixed with Delayed Harmonic Broadcasting Protocol
- with twice the waiting time..
Harmonic broadcasting protocols

- 1998 three variations
- Cautious Harmonic Broadcasting Protocol
  - $C_1$ not changed, $C_2$ alternates $S_2$ and $S_3$
  - $C_i$ from 3 to n, broadcasts $S_{i+1}$ at bandwidth $b/i$
- $b/2$ more bandwidth than Delayed Harmonic Protocol but waiting time only 1 slot
- Quasi-harmonic Broadcasting Protocol
  - segments are divided into fragments which are not broadcast in order
  - waiting time still 1 slot, bandwidth converges to $bH(n)$ as $n$ of subsegments increases
Harmonic broadcasting protocols

- Polyharmonic Broadcasting
  - forces client to wait \( m \) slots before consuming
  - clients can receive while waiting, segments can be broadcasted with lower bandwidth (compared to Harmonic Broadcasting)
  - can use \( m \) times as many segments, waiting time does not increase
  - uses less bandwidth for a given waiting time than Quasi-harmonic Broadcasting
- no interactive VOD with Harmonic Broadcasting
Summary - VOD server

- different protocols share the same strategy: if some videos are more popular than others and clients have local storage, then later parts of video can be broadcasted not as often as the earlier parts
- protocols can save bandwidth on VOD server
- and allow more videos
- or allow server to be cheaper
## Summary - client requirements

<table>
<thead>
<tr>
<th>Broadcasting Protocol</th>
<th>Storage requirement (% of video)</th>
<th>Bandwidth requirement (multiples of $b$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Staggered</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Pyramid</td>
<td>75</td>
<td>$4\text{-}5^{\alpha}$</td>
</tr>
<tr>
<td>Permutation-based</td>
<td>20</td>
<td>2\text{-}3</td>
</tr>
<tr>
<td>Skyscraper</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>Fast</td>
<td>50</td>
<td>6\text{-}8</td>
</tr>
<tr>
<td>Pagoda</td>
<td>45</td>
<td>5\text{-}7</td>
</tr>
<tr>
<td>Harmonic</td>
<td>40</td>
<td>5\text{-}6</td>
</tr>
</tbody>
</table>
Summary

- Comparing server and client requirements there is no clear winner.
- For example: Polyharmonic has lowest bandwidth requirements on server, but too many data channels per video.
- Pagoda is easy on server also, but client bandwidth too high.
- Staggered Broadcasting still only one for interactive VOD and no extra load on client.
Open questions and research

- interactive VOD
- protocols assume fixed bit rate, not the case with MPEG
- changes in video popularity are difficult to handle
- Staggered model still the easiest
Questions?

Thanks!