VAAL, Video Adaptation at Application Layer and Experiments using DCCP

Wassim Ramadan, Eugen Dedu et Julien Bourgeois

Laboratoire d'Informatique de l’Université de Franche-Comté (LIFC)
Montbéliard, France

WPMC
Recife, Brazil
13 October 2010
Rate adaptive video control: optimise network usage

Dynamic bandwidth vs static bitrates

The best bitrate

- if bitrate < bw: unoptimised
- if bitrate > bw: lost packets
  - either on network, if no congestion control
  - or at the sender, if congestion control
- bitrate = bw: the best
Rate adaptive video control: improve user experience

Demonstration
(look at the change at sec. 15)

Small bandwidth, so small bitrate
High bandwidth, so high bitrate
Goals of video adaptation

- Improve user experience
- Optimise network resource usage
1. State of the art
2. Our VAAL method
3. Experiments
4. Conclusions
State of the art

Video adaptation methods based on changed layer

- Low layers changed (unchanged application)
  - for multi-layer encoded video
- Cross-layer (application and transport/network layers changed)
  - iTCP, VTP etc.
- Application layer changed (our VAAL method)
Overview of VAAL

VAAL, Video Adaptation at Application Layer

At each moment:
- congestion control takes care of network conditions
- our VAAL algorithm adapts bitrate to network conditions
Details of VAAL

2 phases executed regularly (each 2 sec):

1. Discovery of network conditions
   - compute FEP (Failed Error Percentage)

2. Quality selection
   - if FEP=0, increase quality
   - if 0<FEP<5%, maintain quality
   - if 5%≤FEP, decrease quality to q':
     \[ q' \leq q(1 - FEP)r, \]
     where q is current quality and r = 105%

VAAL (sender side only)

Available qualities
- Video Q1
- Video Q2
- Video Q3
- Video Q4
- ...........

Quality selector
- Last used quality
- Selected quality

Application layer
- FEP
- filling buffer

Transport protocol layer

Lower layers
VAAL characteristics

- Easy to implement, since it uses a very simple algorithm
- Only the sender (server) is modified
- Only the application layer is modified
- Can either choose among several available qualities (video on demand), or change the quantisation parameter (videoconference)
- Needs a transport protocol with congestion control
  - network-friendly
Network and transport protocols used in experiments

Network topology

Sender \(\rightarrow\) Access Point \(\rightarrow\) Receiver

Transport protocol: DCCP (Datagram Congestion Control Protocol)

- New protocol more adapted to multimedia transmissions:
  - unreliable, but with congestion control
  - can use various congestion control algorithms, such as:
    - TFRC
    - TCP-like
  - allow the sender to find out the received packets
Checking if adaptation works

Quality variation for 5 flows in competition, flow 1

Bitrate of transmitted video (Mb/s)
Percentage of write buffer success (%)
Time (s)

Wassim Ramadan, Eugen Dedu et Julien Bourgeois

Video adaptation and experiments using DCCP
Number of sent and recv packets, 10 flows without gap

Static bitrate leads to either low bitrate, or many network losses (or similar results in lucky cases)

Wassim Ramadan, Eugen Dedu et Julien Bourgeois

Video adaptation and experiments using DCCP
Number of sent and recv packets, 10 flows with 30 sec. gap

VAAL (esp. flows 4 to 7) adapts bitrate to available bandwidth
VAAL has the greatest received/sent packet ratio
Contributions

- One of the rare papers which *analyse* video content adaptation
- The first one which uses DCCP in real experiments in wireless networks
- A new method, called VAAL, using buffer filling
  - simple to deploy (only app-level at sender is modified)
  - works with any transport protocol with congestion control

Conclusions

- By adapting the video bitrate to network bandwidth, VAAL improves network usage (reduces under/over-utilisation) compared to classical static bitrate

Perspectives

- Avoid zigzag quality changing (e.g. using a history)
- Present video quality metrics (PSNR)
### Network information

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Parameter value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet size</td>
<td>1024 Bytes</td>
</tr>
<tr>
<td>Sender wired card</td>
<td>100Mb/s</td>
</tr>
<tr>
<td>Receiver wireless card</td>
<td>802.11b/g</td>
</tr>
<tr>
<td>OSes</td>
<td>Ubuntu 64bits, 2.6.31</td>
</tr>
<tr>
<td>DCCP</td>
<td>Included in the kernel</td>
</tr>
<tr>
<td>Access point</td>
<td>LINKSYS, wireless-G</td>
</tr>
<tr>
<td>Wireless bandwidth</td>
<td>54Mb/s</td>
</tr>
<tr>
<td>Distance (AP ↔ receiver)</td>
<td>50cm</td>
</tr>
</tbody>
</table>
On the value of 5% chosen for local losses

“This refers to end-to-end IP packet-loss rate in video. Considering the packet loss in a terminal-jitter buffer and the packet loss in networks is extremely important. The value should be less than 10 [%].”

(ITU-T, Opinion model for video-telephony applications, ITU Recommendation G.1070, Apr. 2007)
About the simplicity of VAAL adaptation algorithm

for each period of time
   err = 0
   pkts = 0
   while (period not ended)
      pkts++
      write video packet
      if (write error)
         err++
      sleep
      errorRate = err/pkts
   if (errorRate == 0)
      increase bitrate
   else if (errorRate < 5%)
      maintain bitrate
   else
      decrease bitrate