Machine Learning Approach for Quality Adaptation of Streaming Video Through 4G Wireless Network over HTTP

Dhananjay Kumar
Anna University, Chennai, India
dhananjay@annauniv.edu

Arun Raj L.
B.S. Abdur Rahman Crescent University, Chennai, India
arunraj@bsauniv.ac.in
Outline

• Motivation
• System Model and Development
• Machine Learning Approach
• Algorithm Development
• Experimental Results
• Conclusion
Motivation

• Bit Rate Fluctuation in Internet over 4G Wireless
  – Internet service based on Best-effort model
  – Received signal strength fluctuation in wireless channel

Solutions?

➢ Change in underlying hardware and network
➢ Migration to 5G
➢ Content adaptation on-the-top of HTTP

Trigger actions in those trends to adapt contents
System Model

Fig. 1. The Client Server Model

Fig. 2. Multi-threaded Client-Server Communication
Adaptive Streaming System

- Media Capture
- Adaptation Procedure
- Content Encoding / Transcoding
- Stream Processing

Content Transcoding at Server

- Quality estimation through standard procedures (ITU-T G.1070) at client
- Trigger action based on machine learning (State Action Reward State Action (SARSA)) at client
State Action Reward State Action (SARSA) Approach

The Q-matrix is updated after each quality decision as follows.

\[
Q[s_{\text{cur}}, a_{\text{cur}}] \leftarrow Q[s_{\text{cur}}, a_{\text{cur}}] + \alpha \left[ V_q + \gamma Q(s_{\text{new}}, a_{\text{new}}) - Q(s_{\text{cur}}, a_{\text{cur}}) \right]
\]  (1)

where \( \alpha \) denotes learning rate, \( \gamma \) discount factor, and \( v_q \) associated immediate reward

**SARSA Base Quality Adaptation (SBQA):**

(1) **Softmax Policy**: The action is chosen according to the resulting distribution, which is the Boltzmann distribution given by

\[
P(a_j) = \frac{e^{Q(s,a_j)}}{\sum_{i=1}^{\|a_s\|} e^{Q(s,a_i)}}
\]  (2)

(2) **\( \epsilon \)-greedy Policy**: the agent selects at each time step a random action with a fixed probability, \( 0 < \epsilon < 1 \)

\[
Action = \begin{cases} 
\text{random_action_from_A(s)}, & \text{if } r < \epsilon \\
\arg\max_{a} \epsilon A(s)Q(s,a), & \text{otherwise}
\end{cases}
\]  (3)
ITU-T G.1070 Based Video Quality Evaluation

The video quality ($V_q$) is represented as

$$V_q = 1 + I_c I_t$$  \hspace{1cm} (4)

Where $I_c$ represents the basic video quality resulting from the encoding distortion, $I_t$ is the factor governed by degree of robustness due to packet loss.

$I_c$ is expressed in terms of bit rate ($b$) and frame rate ($f$) according to equations (5 - 8) as follows.

$$I_c = I_0 e^{-\frac{(\ln(f) - \ln(f_0))^2}{2D_{Fr}^2}}$$  \hspace{1cm} (5)

$$f_0 = v_1 + v_2 b$$  \hspace{1cm} (6)

$$D_{Fr} = v_6 + v_7 b$$  \hspace{1cm} (7)

$$I_0 = v_3 \left(1 - \frac{1}{1 + \left(\frac{b}{v_4}\right)^{v_5}}\right)$$  \hspace{1cm} (8)

$I_t$ depends on packet loss robustness factor ($D_{P_{plv}}$) and rate of packet loss ($p$) given by

$$I_t = e^{-\frac{p}{D_{P_{plv}}}}$$  \hspace{1cm} (9)

$$D_{P_{plv}} = v_{10} + v_{11} e^{-\frac{f}{v_8}} + v_{12} e^{-\frac{b}{v_9}}$$  \hspace{1cm} (10)
SBQA Using Softmax Policy (SBQA-SP) Algorithm

1. Initialize with number of packets $N$, learning rate $\alpha$ and discount factor $\gamma$, last state $s_{last}$ and Q-matrix $Q$.
2. Compute throughput $(Th)$ resulting from the capture of $N$ packets.
3. Identify current state $s_{cur}$ based on $Th$ value.
4. Read the resolution $res$, and frame per second $fps$ from the header in streamed video.
5. Determine current action $a_{cur}$ based on the current quality segment.
6. While $s_{cur} < s_{last}$ // till last state reached
7. Read the encoded bit rate $b$, and compute frame loss percentage $p$.
8. Calculate the reward (video quality) $V_q$ using (4)
9. Estimate the current throughput $Th_{cur}$ and based on $Th_{cur}$ identify new state $s_{new}$.
10. Compute new action $a_{new} \leftarrow \text{SoftMax}(Q,s)$.
    // Exploration policy function to get best possible future action
11. Update the $Q[s_{cur}, a_{cur}]$ based on (1)
12. $s_{cur} \leftarrow s_{new}$ // Update new state to the current state
13. $a_{cur} \leftarrow a_{new}$ // Update new action to the current action
14. Assign action $a_{cur}$ as feedback to the server
15. End
16. Go to Step 2 and continue till streaming occurs
17. End
SBQA Using $\epsilon$-Greedy Policy (SBQA-GP) Algorithm

1. Initialize fixed probability $\epsilon$ and max // Store maximum value (max) in $s^{th}$ row of Q-matrix.
2. Generate a random value $ran$ in the range 0 to 1.
3. If $ran < \epsilon$
4. $selectedAction = -1$
5. Else
6. For $i = 1$ to $Q_{length}$ // get $Q_{length}$ from Q-matrix
7. If $Q[s,i] >= max$
8. $selectedAction = i$ // action with max reward
9. $max = Q[s,i]$
10. End
11. If $selectedAction = -1$
12. Generate a random number $r$, in range of action.
13. $selectedAction = r$
14. Return $selectedAction$
Q-Learning Based Quality Adaptation (QBQA) Algorithm

1. Initialize the learning rate $\alpha$, discount factor $\gamma$, Q-matrix, and optimal bandwidth value $B_{opt}$.
2. Read the current buffer occupancy level $buf_k$ for $k^{th}$ segment and quality level $q_{k-1}$ for segment $k-1$.
3. For $i = 1$ to $t$ // Training Phase
4. Estimate the bandwidth $bw_k$.
5. Assign $s_k = \{bw_k, buf_k, q_{k-1}\}$ // Current State
6. $a_k = \text{Softmax}(Q, s_k)$ // Exploration policy function to get best possible action.
7. Calculate the quality factor related to bandwidth and buffer occupancy level using the equation
   $$R_{quality} = -1.5 \cdot \left| bw_k \cdot \frac{1+(buf_k/B_{opt})}{3-(bw_1/a_1)} - a_k \right|$$
8. Calculate the quality factor related to switch in quality using the equation
   $$R_{switches} = -|q_{k-1} - a_k|$$
9. Read the duration of video freeze $t_{stall}$, time elapsed from the last freeze $t_{play}$ and number of freezes $n$.
10. Calculate the quality factor related to video freezing using the equation
    $$R_{freezes} = \begin{cases} 
      -100 \cdot \frac{a_k}{bw_k} \cdot \frac{e^{t_{stall}/10}}{\ln(t_{play}+1)} & a_k = a_1 \\
      -100 \cdot \frac{a_k}{bw_k} \cdot \frac{e^{n+t_{stall}/10}}{\ln(t_{play}+1)} & a_k \neq a_1 
    \end{cases}$$
11. Calculate
    $$R_{total} = R_{quality} + R_{switches} + R_{freezes}$$
12. End
13. Determine the resultant state, $s_{k+1}$ using $\{bw_{k+1}, buf_{k+1}, q_k\}$.
14. Update the Q-matrix using
    $$Q[s_k, a_k] \leftarrow (1 - \alpha) Q[s_k, a_k] + \alpha [R_{total} + \gamma \max_b Q(s_{k+1}, b)]$$
15. Estimate the bandwidth $bw_k$ // Testing Phase begins
16. Assign $s_k = \{bw_k, buf_k, q_{k-1}\}$
17. $a_k = \max_a (Q(s_k, a))$
18. Send $a_k$ as feedback to the server
19. Repeat from Step 15 until streaming occurs.
Experimental Results

Test Parameters as per ITU-T J.247

<table>
<thead>
<tr>
<th>S.no</th>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Transmission Errors with packet loss</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>Frame Rate</td>
<td>5 fps to 30 fps</td>
</tr>
<tr>
<td>3.</td>
<td>Video Codec</td>
<td>H.264/AVC (MPEG-4 part10), VC-1, Windows Media9, Real Video (RV 10), MPEG-4 Part 2</td>
</tr>
<tr>
<td>4.</td>
<td>Video Resolutions and bit rates</td>
<td>QCIF: 16 - 320 kbps</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CIF: 64 - 2000 kbps</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VGA: 128 - 4000 kbps</td>
</tr>
<tr>
<td>5.</td>
<td>Temporal errors (pausing with skipping)</td>
<td>Maximum of 2 seconds</td>
</tr>
</tbody>
</table>
Experimental Results cont.

VQM

Inter packet delay

3SSIM
Experimental Results cont.

Fig.(a) Original Live Video

Fig.(b) Decoded at Receiver
Conclusions

- On-the-top (OTT) implementation with the last mile connectivity through 4G mobile
- ITU-T G.1070 model embedded in the algorithm
- The State Action Reward State Action (SARSA) approach achieves the system goal
- Experiments conducted on live / stored video stream and results validated through FR metrics
  - PSNR, SSIM, MS-SSIM, VQM, and 3-SSIM
- Implementation of multi-casting service will require additional mechanism
Acknowledgements

The University Grants Commission (UGC), New Delhi, India

Anna University, Chennai, India