Online Learning for QoE-based Video Streaming to Mobile Receivers
Nesrine Changuel, Bessem Sayadi, Michel Kieffer

To cite this version:
Nesrine Changuel, Bessem Sayadi, Michel Kieffer. Online Learning for QoE-based Video Streaming to Mobile Receivers. GLOBECOM 2012, Dec 2012, Anaheim, United States. pp.1-5. hal-00727798

HAL Id: hal-00727798
https://hal-supelec.archives-ouvertes.fr/hal-00727798
Submitted on 4 Sep 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Abstract—This paper proposes a cross-layer control mechanism to stream efficiently scalable videos to mobile receivers. Its goal is to maximize the quality of the received video while accounting for the variations of the characteristics of the transmitted content and of the channel. The control problem is cast in the framework of Markov Decision Processes. The optimal actions to apply to the system are learned using reinforcement learning. For that purpose, the quality of the decoded frames at receiver is inferred by an observation (i) of the quality of the various scalability layers and (ii) of the level of queues at the Application and Medium Access Control layers of the transmitter only. Delayed as well as absence of information on the channel state are considered. Experiments show that the performance of the proposed solution is only slightly degraded with delayed or missing channel state information. The performance degradation is larger when considering a basic bitstream slightly degraded with delayed or missing channel state information. The considered model is more general than that considered in [3], since layers of one or several frames may be transmitted in one time slot. This allows to change the speed at which encoded frames are transmitted depending on the channel conditions and on the state of the buffers. At MAC layer, HARQ ACK/NACK [13] may be used to infer the channel conditions. The associated delay is considered in [15]. A model-based RL algorithm is used in [14], [15]. Learning with constant reward delay is considered in [15]. A model-based RL algorithm is used in discrete- and continuous-valued state. Nevertheless, in the considered context, the scheduler is allowed to transmit more than one data unit (frames or GoPs) at each time slot leading to a time-varying delay between transmission and display.

Without channel state information, [16] shows that the observation of the level of the MAC buffer provides a satisfying estimate of the channel state. This prevents using delayed measurements. Moreover, the evaluation of the optimal layer filtering policy is performed offline. A way to tackle the problem of time-varying characteristics of the encoded videos and of the channel, is to learn and update on-line the optimal layer filtering policy. RL techniques are well-suited to update periodically the state-value function and the policy.

The remainder of the paper is organized as follows. Section II introduces the considered unicast video streaming. Section III recalls for transmission of real-time video over burst-error wireless channels is considered using constrained optimization of the encoding rate. The quantization parameters of the source coder are optimized dynamically to maximize the average quality while satisfying expected rate constraints using an accurate model of the rate-distortion (R-D) characteristics of the source.
II. System Description

Consider the video streaming system to mobile receivers sketched in Figure 1. The core network consists of a streaming server hosting a scalable video coder, a proxy, and a base station. Packets are transmitted through a wireless channel to a mobile client. Among the components of the base station (eNodeB [13]), we consider mainly the MAC buffer. The MAC scheduler of eNodeB, as well as its physical layer, its radio front-end, the wireless channel, the physical layer of the receiver, and the part of the MAC layer at receiver side managing ACK/NACK procedures are considered as belonging to the channel. Focusing on SNR scalability, our goal is to design a layer filtering algorithm to maximize the QoE of the decoded video at receiver side.

Figure 1. Scalable video transmission system to a mobile receiver

Streaming server: The video sequence is segmented into frames and encoded into L layers: a base layer and \( L - 1 \) enhancement layers. Frames are generated with a constant period of time \( T \). The encoding parameters (quantization steps, frame rate, etc.) are controlled by the streaming server, independently of the remainder of the chain.

Proxy: The \( L \) SNR layers are packetized and fed, via a lossless network, to the Post-Encoding (PE) buffer. The controller performs layer filtering within the proxy: for each frame, SNR layers may be sent, kept, or dropped. Layer filtering may also be performed in the base station (PDCP layer [13]). The layer filtering process should maximize the QoE at receiver side by taking into account most factors impacting it: frame type, number of SNR layers, lost packets due to PE and MAC buffer overflow, and effect of error and loss concealment.

Base station and channel: The base station contains a buffer dedicated to each user to perform rate and bandwidth allocation (MAC scheduling, see [17], [18]) among users. Packets transmitted by the layer filtering are fed from the PE buffer to the MAC buffer of the base station after being segmented into Packet Data Units (PDUs). One has to control the MAC buffer to avoid overflow in order to prevent PDUs from being dropped. PDUs are then transmitted to the mobile receiver via a wireless channel. When the channel state is used to control the layer filtering process, it has to be inferred, e.g., using some feedback from the mobile client or using the level of the MAC buffer at transmitter side.

Receiver: The mobile receiver stores correctly received PDUs in its own MAC buffer. Packet de-encapsulation and buffering in the buffer at APL layer is done as soon as all corresponding PDUs have been received. Complete or incomplete frames are then processed by the video decoder. Outdated packets are dropped, without being decoded. Some packet loss concealment may be put at work at the receiver side. Handover issues are not addressed: the streaming server is assumed to transmit video to a mobile receiver considered linked to the same base station during the whole streaming session.

III. MDP and Learning

The video streaming system is modeled in the MDP framework. Time is slotted into discrete-time intervals of length \( T \). The \( t \)-th time slot is the time interval \([t, t + 1)\). \( T \) may be equal to the frame period, corresponding to the cadence of the encoder, or to the period at which the MAC scheduler delivers PDUs.

An MDP is a 4-tuple \((S,A,P,r)\), where \( S \) is the set of states of the considered system, \( A \) is the set of actions, \( P(s_{t+1}|s_t,a_t) \) is the transition probability from \( s_t \in S \) at time \( t \) to \( s_{t+1} \in S \) at time \( t + 1 \), when the action \( a_t \in A \) is applied. Finally \( r (s_t,a_t) \) is a reward function indicating the immediate reward obtained when applying \( a_t \) in state \( s_t \). Provided that all components of the MDP are clearly defined, the optimum policy may be evaluated, e.g., by value or policy iteration. Alternatively, when some components of the MDP are difficult to obtain, or are time-varying, a good policy may be obtained on-line by RL, see [2].

In the context of wireless video streaming, the characteristics of the video sequence and of the channel are time-varying. The policy that would be obtained via policy or value iteration for some transition probability matrix and some reward function under some source and channel conditions would probably not be well suited to other conditions. RL aims at estimating a good policy without requiring an accurate knowledge of the \( P(s_{t+1}|s_t,a_t) \). There are several classes of on-line RL algorithms. This paper focuses on Temporal Difference (TD) learning [2], which aims at directly estimating the action-value function (or Q-function) \( Q(s,a) \), indicating the expected long-term reward starting from \( s \), taking the action \( a \). The optimal policy is then derived by selecting the action maximizing \( Q(s_t,a_t) \). Popular on-line algorithms in this category are SARSA and Q-learning [2].

With Q-learning, considered in what follows, the Q-function is updated at each time slot according to

\[
Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha \delta_{TD,t}
\]

with \( \delta_{TD,t} = r_t(s_t,a_t) + \gamma \max_{a' \in A} Q_t(s_{t+1},a') - Q(s_t,a_t) \), where \( a_t \) is the greedy action in state \( s_{t+1} \), which maximizes the current estimate of the Q-function; \( \alpha \in [0,1] \) is a time-varying learning rate parameter and \( \gamma \) is a discount factor indicating the relative importance of present and future rewards. \( Q(s,a) \) can be initialized arbitrarily for all \((s,a) \in S \times A \). At the beginning of the learning process, the controller should go through the states many
times in order to learn the optimal actions. During the exploration
steps, the Q-learning rule in (1) is performed by executing actions in
each state several times until Q converges.

IV. MODEL OF THE STREAMING SYSTEM

The states of the system consists of the frame type \( s^t \), the level of
the PE buffer \( s^m \), that of the MAC buffer \( s^m \), and the channel state
\( s^h \). All state components are gathered in \( s = (s^t, s^h, s^m, s^h) \in \mathcal{S} \).

The actions indicate the number of scalability layers to transmit, keep or
drop from the PE buffer.

A. States

The frame type \( s^t \) (I, P, or B) is useful to describe the impact of
a frame losses on the video quality. In a GoP, transitions between
frame types may be described by a stationary Markov process [19].
In what follows, the structure of the GOP is assumed constant for
the whole video sequence.

The state of the PE buffer is \( s^e \in \mathcal{S}^e \) and that of the MAC buffer is
\( s^m \in \mathcal{S}^m \). Here, \( s^e \) describes the number of encoded frames stored;
this helps controlling the delay introduced within the system. The
state of the MAC buffer indicates the number of PDUs or of bits
(PDUs are assumed to have the same size) in the buffer.

The channel state \( s^h \) describes the time-varying channel conditions,
such as the rate, probability of error, capacity, etc., assumed constant in
\( [t, t+1) \). Here, \( s^h \) corresponds to the channel rate modeled as
the realization of an \( N_t \)-state Markov chain as in [20]. At time \( t \),
the state \( s^h = h = \{1, \ldots, N_h \} \) represents a rate within the set
\( \mathcal{R} = \{R_0, \ldots, R_{N_h} \} \). The transition probability \( p_{h, \ell} = p(s^h = \ell | s^{h-1} = k) \) from state \( h \in \{1, \ldots, N_h \} \) to state \( k \in \{1, \ldots, N_h \} \)
has usually to be estimated on-line.

Three scenarios concerning the knowledge of the state of the channel
are considered: (i) instantaneously available, \( s^h \) is available at time
\( t \), which requires feedback with very short delay; (ii) delayed, only
\( s^h \) is available at time \( t \), with \( \delta > 0 \) some feedback delay, which is
more realistic; (iii) unknown, no channel state feedback is available.

B. Actions

In the proposed model, several frames may be transmitted in each
time slot, allowing to speed up or slow down the frame scheduling
according to the channel and buffer conditions. The layer filtering
process has to determine the number of layers among the \( F \) oldest
frames stored in the PE buffer to send to the MAC buffer.

The vector of actions \( a_t = (a_{t,f}) \in \mathcal{A}^{F \times L} \) with \( L \in \{1, \ldots, L \} \)
and \( f \in \{1, \ldots, F \} \) taken in \( [t, t+1) \) represents the filtering
decisions. For the \( f \)-th SNR layer and the index \( f \) of the frame in the
PE buffer \( (f = 1 \) is the oldest one and \( f = F \) is the earliest),
\( a_{t,f} = 1 \) indicates a transmitted layer, \( a_{t,f} = -1 \) a dropped layer,
and \( a_{t,f} = 0 \) indicates that the layer is temporarily kept in the buffer.

A layer may be decoded only if the corresponding higher-
importance layers have already been decoded. When some layer is
dropped from the PE buffer, the actions are designed in such a way
that all refinement layers of the dropped layer belonging to the same
frame are also dropped.

C. Estimation of the reward

RL requires for each time slot some reward \( r_t(s_t, a_t) \) provided
by the system to update \( Q(s_t, a_t) \). Ideally, \( r_t(s_t, a_t) \) should be (i)
related to the user QoE (level of satisfaction) after applying action \( a_t \)
when the state is \( s_t \) and (ii) fed back instantaneously by the receiver.
QoE information may be obtained considering the PSNR, the SSIM,
or other metrics [21], [22], [23]. Automatic QoE measurement tools
[24] may be particularly useful in this context. Unfortunately, even
with such tools, due to buffering, the action \( a_t \) will have an impact
on the user QoE only after some delay \( \delta_t \). This reward evaluation
delay is time-varying, since the PE buffer is allowed to temporarily
keep frames or to transmit several frames in the same time slot. Let
\( r_{t}^{L_t}(s_t, a_t) \) be the reward provided by the receiver after a delay \( \delta_t \)
when the state at time \( t \) is \( s_t \) and the action \( a_t \).

To address the problem of delayed rewards, the QoE (and the reward) \( r_{t}^{L_t}(s_t, a_t) \) obtained by the receiver at time \( t + \delta_t \) is predicted
at time \( t \) at transmitter side \( \hat{r}_t(s_t, a_t) \). To facilitate prediction, we
assume that the QoE can be evaluated frame by frame (which is
not true for a video, since the motion plays an important role). We
assume further that an overflowed MAC buffer drops all entering
frames and that an overflowed PE buffer drops its oldest frame (the
drop action is forced for that frame). Finally, retransmissions and
adaptive modulation and coding schemes are used at MAC layer to
ensure the delivery of all PDUs from the MAC buffer to the receiver.

Several cases have now to be considered.

1) Transmission/drop of a single frame: Consider that layers from
a single frame are transmitted from the post-encoding buffer to the
MAC buffer. In absence of overflow or empty buffers, the reward
evaluation delay remains constant \( \delta_t = \delta_{t-1} \) and the QoE evaluated
at the encoder is equal that evaluated at the decoder

\[
\hat{r}_t(s_t, a_t) = r_{t}^{L_t}(s_t, a_t) = \sum_{\ell=1}^{L_{t}} \max(0, a_{t,\ell} \cdot q(s_0^t, \ell)) \tag{2}
\]

where \( q(s^t, \ell) \) is the additional QoE measure provided by the
transmission of layer \( \ell \) from a frame of type \( s^t \).

Assume now that a single frame is still transmitted and that layers are
dropped either by the layer filtering process, or due to buffer
overflow. When the base layer remains, (2) is still valid. When it is
dropped, concealment is performed at receiver side, and \( \delta_t = \delta_{t-1} \).
Several concealment techniques may be used. Here frame copy [25]
is considered and the QoE of the current frame is assumed to be equal
to that of the previous frame reduced by a factor \( \lambda(s^t) \) depending
on the type of the lost frame (lost I frames will have more impact
on the next frames than lost P frames). One then gets

\[
\hat{r}_t(s_t, a_t) = \hat{r}_{t-1}(s_{t-1}, a_{t-1}) - \lambda(s^t) \tag{3}
\]

2) Temporarily kept frames: When frames are neither transmitted
to the MAC buffer, nor dropped (intentionally or as a consequence of
post-encoding buffer overflow), the reward evaluation delay decreases
\( \delta_t = \delta_{t-1} - 1 \) for the next frame transmitted from the PE buffer to
the MAC buffer. As a consequence, estimating \( r_{t}^{L_t}(s_t, a_t) \) is quite
difficult, since no frame is transmitted at time \( t \). The impact of the
QoE at the receiver will be via the next transmitted frames, for which
no decision has been considered at time \( t \). Thus, we consider that the
reward is the average QoE of the next frame, i.e.,

\[
\hat{r}_t(s_t, a_t) = \frac{\hat{r}_{t-1}(s_{t-1}, a_{t-1}) - \lambda(s^t) + \sum_{\ell'=1}^{L_{t-1}} \sum_{\ell=1}^{L} q(s_0^t, \ell)}{L + 1} \tag{4}
\]

The first term in (4) corresponds to a dropped next frame and is equal
to (3), the second term corresponds to a number of layers transmitted
going from 1 to \( L \) with rewards as in (2).

3) Transmission of several frames: When layers of several frames
are transmitted from the PE buffer to the MAC buffer during the
same time slot, the reward evaluation delay increases \( \delta_t = \delta_{t-1} + 1 \)
for the next frame transmitted, since more frames are put in the MAC
buffer. This decision will impact the QoE of several frames at receiver
side. It is again quite difficult to evaluate precisely \( r_{t}^{L_t}(s_t, a_t) \). The
transmission of layers of several frames should not lead to jitter in the
B. Simulation conditions

stored in the MAC buffer provided that it does not overflow. Here, NAL units are ordered based on their quality level and are the minimum among the rewards obtained for each individual frames obtained by RL and that of the BBSE, considering

3

Figure 3. PSNR of the decoded Mother&Daughter sequence, control policy obtained by RL and that of the BBSE, considering 3 channel scenarii.

a

This prevents sending during a time slot all layers from the first frame and only a single layer from the next frames.

V. Experimental results

The performance of the proposed layer filtering process has been evaluated on the Foreman.qcif and Mother & Daughter.qcif sequences at 30 fps using the H.264/SVC encoder (JSVM 9.19) [26] with three MGS scalability layers per frame (L = 3). The period at which the controller is operating is \( T = 1/30 \) s. IPPP GoPs of 16 frames are considered. To avoid the drift due to SNR layer filtering, all frames are encoded as key pictures for which motion compensation is performed using only the base layer of the previous frames.

A. Reference Basic BitStream Extractor (BBSE)

The BBSE provided in the JSVM [26] serves as reference. It extracts SVC layers according to a specific priority and accounting for the level of the MAC buffer. The priorization is done according to high-level syntax elements: dependency, temporal, and quality ids. Here, NAL units are ordered based on their quality level and are stored in the MAC buffer provided that it does not overflow.

B. Simulation conditions

The channel is described by a two-state Markov model, with a bad (B) state with channel rate \( R_B \) and good (G) state with channel rate \( R_G \). The state transitions occur with a period \( T \) and with probabilities \( P(G|G) = 0.9 \) and \( P(B|B) = 0.8 \), leading to the stationary probabilities \( P(G) = 0.66 \) and \( P(B) = 0.33 \). As indicated in Section IV-A, the state of the channel may be instantaneously available, delayed, or unknown, depending on the considered scenario.

The PE and MAC buffers contain at most \( B^P = 25 \) frames and \( B^M = 500 \) PDUs respectively. The PDUs are assumed to be static with size 336 bits, which is consistent with the 3GPP radio link control protocol specification [27]. To get a model with reduced state space and accelerate the convergence of RL, the state of the PE buffer is quantized to two intervals \([0, 22]\) represented by \( s_1 = 1 \), indicating a satisfying occupancy and \([22, 25]\), represented by \( s_2 = 2 \), indicating a buffer close to overflow. As the MAC state transitions depend on the encoded frame size contrary to the PE buffer state, a finer quantization is considered for the state of the MAC buffer. Moreover, when the channel state is unknown, the control has to rely on the observation of the state of the MAC buffer only. MAC buffer states are quantized into five intervals. The fifth is smaller than the others to anticipate overflow and prevent PDUs from being dropped.

The number of possible actions is kept small to limit the learning complexity. The layers of at most two frames in the PE buffer may be fed to the MAC buffer, i.e., \( F = 2 \). The action for each layer is organized as \( a_t = (a_{t,1}, \ldots, a_{t,3}, a_{t,1}, \ldots, a_{t,3}, a_{t,1}) \in A \) with \( a_{t,0} = \{ -1, 0, 1 \} \). When only the base layer of a frame is transmitted, the other layers are dropped, see Table V-B. For example \( a^{(9)} \) indicates the transmission of two layers of the oldest frame and of the highest priority layer of the next frame.

The reward involves the PSNR of the two last frames, but any other video quality metric may be used in the proposed learning process. The value of the PSNR reduction \( \lambda(s_{1, t}) \) depends on the frame type. Off-line measurements are performed using different sequences with frame copy concealment leading to \( \lambda(I) = 15 \) dB when an I frame is lost and \( \lambda(P) = 8 \) dB when a P frame is lost.

C. Results

On-line Q-learning is performed over 5000 time slots (on Foreman and Mother\&Daughter sequences by repeating the sequences from the beginning after 300 frames). \( Q(s, a) \) is initialized to zero for each state-action pair. The discount factor is set to \( \gamma = 0.9 \).

Considering the three levels of knowledge of the channel state, Figures 2 and 3 compare the performance of the streaming server using the policy obtained by RL and that of the BBSE. Different channel rates are considered form 50 to 500 kbit/s for Foreman and from 40 to 250 kbit/s for Mother\&Daughter. In Figures 2 and 3, a separate learning is performed for each value of the rate.

As shown in Figures 2 and 3 and in Table II when the channel state is instantaneously known, the proposed RL-based scheme outperforms the BBSE in most cases for both video sequences. The gain of the proposed scheme compared to the BBSE is mainly due to more accurate SNR layer selection, which better accounts for the

\[
\begin{align*}
\alpha^{(1)} & = (0, 0, 0, 0, 0, 0) \\
\alpha^{(2)} & = (1, -1, -1, 0, 0, 0) \\
\alpha^{(3)} & = (1, 1, 0, 0, 0) \\
\alpha^{(4)} & = (1, 1, 0, 0, 0) \\
\alpha^{(5)} & = (1, 1, 1, 1, 0, 0) \\
\alpha^{(6)} & = (1, 1, 1, 1, 1, 1)
\end{align*}
\]

\[
\begin{align*}
\alpha^{(7)} & = (1, 1, 1, 1) \\
\alpha^{(8)} & = (1, -1, -1, 1, -1) \\
\alpha^{(9)} & = (1, -1, -1, 1, -1) \\
\alpha^{(10)} & = (1, 1, -1, 1, -1) \\
\alpha^{(11)} & = (-1, -1, -1, 0, 0, 0) \\
\alpha^{(12)} & = (1, 1, 1, 1, 1, -1)
\end{align*}
\]

Table I

Considered actions when \( L = 3 \) and \( F = 2 \).
The contribution of each layer to the video quality, and for the foresighted policy obtained by RL.

To evaluate the robustness of the proposed approach to variations of the characteristics of the system, the policy obtained by learning with a video sequence, here \texttt{coastguard.qcif} is used, and applied to Foreman when the channel state is unknown, as illustrated by the red dashed curve in Figure 4. The behavior in presence of rate variations is similar to that observed using only Foreman for learning, showing the robustness to variations of the transmitted content.

To evaluate the robustness to variations of the channel rate, the policy learned for \( R_c = 250 \) kbps is applied for different channel rates when the channel state is unknown for the Foreman sequence, see Figure 4. The results obtained when the policy is learned at each channel rate and when it is learned using only \( R_c = 250 \) kbps are very close, except when the channel rate becomes very different from that used for the learning process. The RL based control process is thus quite robust to moderate variations of the channel characteristics.

VI. CONCLUSION

A RL solution for scalable video transmission over a time-varying wireless channel is proposed. Experiments show that with delayed or without channel state information, the performance obtained with the policy obtained by RL is only slightly degraded compared to a case where the channel state information is available. The performance degradation is larger using a BBSE.

The robustness to variations of the characteristics of the channel and of the video sequence has been shown experimentally. The QoE metric considered in this paper is the PSNR, but other quality metrics could readily be used in the learning process, provided that they allow the QoE of the receiver to be predicted at the transmitter. Implementation of the proposed layer filtering process in a prototype LTE network including a functional eNodeB and UE is planned.

VII. ACKNOWLEDGMENTS

The authors would like to thank Prof. M. van der Schaar for valuable discussions on dynamic programming.

Table II

<table>
<thead>
<tr>
<th>Channel state</th>
<th>Foreman PSNR</th>
<th>Rate</th>
<th>Mother &amp; Daughter PSNR</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Known</td>
<td>0.4</td>
<td>60</td>
<td>0.39</td>
<td>22</td>
</tr>
<tr>
<td>Delayed</td>
<td>0.54</td>
<td>70</td>
<td>0.43</td>
<td>33</td>
</tr>
<tr>
<td>Unknown</td>
<td>0.91</td>
<td>120</td>
<td>0.84</td>
<td>42</td>
</tr>
</tbody>
</table>

Figure 4. PSNR as a function of channel rate for Foreman sequence when optimal policy is learned with Foreman at each channel rate (plain), with Coastguard at each channel rate (dashed red) and with Coastguard at \( R_c = 250 \) kbps only (dotted blue).

REFERENCES


