Abstract—We present an autonomous learning agent for adaptive video streaming in best effort networks. The agent learns an optimal control strategy in regards to the delivered quality of experience without the need for implementation of a complex heuristics.

I. INTRODUCTION

Video streaming functionality is present in most, if not all, web enabled devices. For a device to achieve delivery of a high quality streaming service it needs to continuously reproduce the video with sufficient bit-rate and without any errors. However, in best-effort networks multiple sources are competing for the same resources and therefore no guarantees are given that resources will be available when needed [1]. Since video streaming is a data-intensive process it is particularly susceptible to variations in throughput. If the resources are not sufficient the video playback will freeze or, otherwise, the video will have to be streamed with lower bit-rate.

To address the variability in available resources, adaptive streaming technologies are developed such as HTTP streaming [2] and SVC [3]. These technologies allow for continuous adaptation of the bit-rate as the video is being streamed so that a controlled degradation in quality, or quality of experience (QoE), can be achieved.

Adaptive streaming clients available today predominantly demonstrate a strategy of streaming at the highest possible bit-rate, for as long as possible [4]. Only when the buffer becomes depleted they switch to lower bit-rate levels. This approach does not take into account the subjective perception of quality [5], nor the degradation to QoE that comes from frequent changes between quality levels [6]. These ‘greedy’ strategies can further lead to over-provisioning of bit-rate with little to no positive effect on the QoE, due to the nonlinear dependence between bit-rate and user perception (Fig 1). Further, this leads to a higher probability of buffer depletion and playback freeze, which overall results in a lower QoE.

In this paper we propose a solution that addresses these issues by developing a QoE estimation function [7] for adaptive video streaming that incorporates: 1) the subjectively perceived quality; 2) the impairments from playback freeze; and 3) the effect of the frequency and amplitude of change in quality. Then we propose a design for an intelligent streaming agent that optimizes its decision strategy based on the subjective QoE delivered to the customer.

The agent uses a reinforcement learning [8] method to infer the optimal decisions trained in an environment of simulated background traffic. This approach does not require design of a control heuristic for the agent, which can adapt its strategy based on the understanding of the subjectively perceived quality. The intelligent adaptive streaming client hence provides better utilization of the network resources and higher QoE, for a wide range of network conditions.

II. METHOD

The HTTP adaptive streaming architecture is composed of a HTTP server, with access to a video database, and a client application, running on a connected device (Fig. 2). The client requests chunks of video from the server and reproduces the video on the device display. The client can choose to request chunks at different quality levels (L1, L2, etc), based on its estimate of network throughput and its own control strategy.

Our agent, residing on the client, evaluates its performance based on a broader view of delivered QoE given in (1).

\[
\text{QoE} = w_{\text{subjective}} \cdot (\text{bit-rate}, \text{video_si}, \text{video_ti}) + w_{\text{buffer}} \cdot \{(\text{len}_1, \text{T}_1), (\text{len}_2, \text{T}_2)\} + w_{\text{freeze}} \cdot \{(\text{delta}_1, \text{T}_1), (\text{delta}_2, \text{T}_2)\}
\]

This work has been partially supported by ARTEMIS project DEMANES (Design, Monitoring and Operation of Adaptive Networked Embedded Systems, contract 295372).
relative value of degradation. A typical subjective quality curve obtained with the Maximum Likelihood Difference Scaling method [9] is presented in Fig. 1.

The $f_{\text{freeze}}$ function calculates the degradation incurred by a freeze in playback. The value is based on research done on the psychological effects of this type of impairment. The degradation value given is proportional to the length of the freeze and to the frequency of these occurrences. The $f_{\text{freeze}}$ inputs a list of pair values. The first ($len_i$) is the length of the event and the second ($Ti_i$) is the time at which it occurred. This way the recent events have bigger effect and the older ones have smaller (decayed by $e^{-\lambda t}$). The total impairment is a sum of the effect of each event from the beginning. The same approach is taken for the $f_{\text{volChange}}$, where the $delta_i$ is the distance between the levels and $T_i$ is the moment of occurrence. Since the three types of impairments have different amount of impact they are weighted differently: $w_s$, $w_l$ and $w_f$. These weights can be adjusted according to results from subjective trials.

Fig. 3. Reinforcement learning loop

Generally, a reinforcement learning framework consists of an agent working in an environment (Fig. 3). The agent executes actions over the environment and receives state changes and reward feedback from it.

Fig. 4. Design of the RL Intelligent agent

In our case the actions are the download requests at different bit-rates; the reward is the QoE estimation; and the state is a combination of the video buffer condition, network throughput estimation and the position in the video stream (Fig. 4).

The estimation of the network throughput is implemented by a set of filters: Exponentially Weighted Moving Average (Fig. 4). 

III. Results

The agent is trained in an environment of simulated background traffic. The values for the weights of the QoE function were selected as 1 for $w_s$, 2 for $w_l$ and 10 for $w_f$. The background traffic is modeled with a self-similar process, where the number of file transfers is sampled from a Poisson distribution, and the length of each transfer sampled from a Pareto distribution. The Hurst parameter for the background traffic is set to 0.7. The agent learns to avoid freezes quickly, since this is heavily penalized in the QoE function. The estimated QoE incurred over after each episode is given in Fig. 5. We trained the agent on 1000 episodes, and found that the efficiency of the inferred control strategy improves considerably as new training episodes are included.

Fig. 5. Performance of the RL intelligent agent

IV. Conclusion

Our approach provides a flexible solution for the problem of adaptive streaming, but could also be used more generally in other control problems where it is hard to model the system deterministically. Another direct benefit is the inclusion of human perception (subjective) factors in the decision. Future work should explore the depth of predictive capabilities and more complex QoE non-linear dependencies between the given factors, as well as training on network traces.

REFERENCES