OPTIMIZING QOE OF MULTIPLE USERS OVER DASH: A META-LEARNING APPROACH

Liangyu Huo[1]*, Zulin Wang*[1]*, Mai Xu*[1]*, Zhiguo Ding*[2]*,Xiaoming Tao*[3]

[1]School of Electronic and Information Engineering, Beihang University, Beijing, China
[2]School of Electrical and Electronic Engineering, The University of Manchester, Manchester, M13 9PL, UK
[3]Department of Electronic Engineering, Tsinghua University, 100084 China

ABSTRACT

Dynamic adaptive video streaming over HTTP (DASH) plays a key role in video transmission over the Internet. The conventional DASH adaptation approaches concentrate on optimizing the overall quality of experience (QoE) for all client sides, neglecting the QoE diversity of different users. In this paper, we formulate the QoE optimization of multi-user preferences as a multi-task deep reinforcement learning problem, in which QoE refers to the metrics of visual quality, fluctuation and rebuffing events. Then, we propose a meta-learning framework for multi-user preferences (MLMP) as a new DASH adaptation approach. Finally, the simulation results show that the proposed approach outperforms state-of-the-art DASH adaptation approaches in satisfying the different users' QoE preferences regarding the three metrics.

***Index Terms*—** DASH, adaptation approaches, user preferences, meta-learning, reinforcement learning

1. INTRODUCTION

In the next five years, mobile traffic is expected to grow exponentially [1], in which 75% of the data are in the form of video streams. Recently, dynamic adaptive video streaming over hypertext transfer protocol (DASH) [2] has become a dominant standard for video streaming over the Internet. In the DASH protocol, video segments are encoded with different bit-rates to form an adaptation set of representations. DASH adaptation is needed to select one representation based on the network throughput to optimize the quality of experience (QoE). However, this is challenging due to the three following reasons: (1) conflicts in different metrics of QoE exist, such as visual quality, fluctuation and rebuffing events; (2) the network throughput may dramatically change; and (3) the user preference on different metrics of QoE is complex, thus making it difficult to model.

To solve the above challenges, several DASH adaptation approaches have been investigated such as [3–7]. However, most of them either cannot be generalized to varying network conditions or requires an accurate estimation of the future throughput. Hence, several approaches based on reinforcement learning and deep reinforcement learning (DRL) [8] have been proposed to optimize multiple metrics of QoE in DASH adaptation, which consider the future network throughput by learning from past experiences [9–13]. However, people in the multi-user scenario may have their own preferences on the different metrics of QoE [14, 15], such that the individual preference should be taken into account. This is out of scope for the existing single-model RL approaches [16].

In this paper, we propose a novel DASH adaptation approach, in which a meta-learning framework for multi-user preferences (MLMP) is developed to optimize the QoE across different users. Specifically, we first formulate the QoE optimization of multi-user preferences as a multi-task DRL problem, as a type of meta-learning. Then, we design the MLMP framework to solve the proposed problem, where the common knowledge and diversity of each user's preference can be learned in a hierarchy pattern. To our best knowledge, this paper is the first attempt to apply meta-learning framework for optimizing the QoE in DASH.

2. SYSTEM MODEL

2.1. QoE metrics for DASH

As depicted in [11], a video sequence is encoded to different representations at various bit-rates and resolutions, which are divided into several segments. A DASH client requests and downloads these segments sequentially. Note that, if the client-side buffer occupancy is emptied before the next segment is downloaded, a rebuffing event will take place. Details about the mathematical model of DASH can be found in [11].

Following [17, 18], we consider three metrics influencing the QoE for DASH: instantaneous visual quality, quality fluctuation and rebuffing events. Specifically, we choose structural similarity (SSIM) [19] as the visual quality metric, denoted by q_t. Let ϕ_t and L_t denote the rebuffing time and the current segment, then the overall QoE can eventually be computed as follows:

$$\mathrm{QoE}_{L_t} = \alpha \cdot q_t - \beta \cdot |q_t - q_{t-1}| - \gamma \cdot \phi_t. \qquad (1)$$

In (1), α, β and γ are the weights for the instantaneous visual quality, the fluctuation and the rebuffing time, respectively.

This work was supported by NSFC under Grants 61471022, 61876013 and 61573037, and by the Fok Ying Tung Education Foundation under Grant 151061.

Mai Xu is the corresponding author of this paper.

2.2. Multi-task DRL modeling for DASH adaptation

Following the assumption in [11], we can formulate DASH adaptation as a 5-tuple markov decision process, with *state space* $\mathcal{S}$, *action space* $\mathcal{A}$, *reward* function $\mathcal{R}: \mathcal{S} \times \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, *transition probability* $\mathcal{P}: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ and *discount factor* η. Then, we can apply DRL for optimizing the QoE. As depicted in Figure 1, at each time step t, the *agent*, i.e., the video client, observes *environment state* s_t and then selects a certain representation with quality q_t to download. Hence, the *action* at each time step is also denoted as q_t.

To estimate the QoE of (1), the *state space* $\mathcal{S}$ is composed of the four following elements: (1) the throughput for the last few time steps (for predicting the current throughput [11]); (2) the last rebuffing time; (3) the quality of the last downloaded segment; and (4) the current buffer occupancy. Given *state* $\mathbf{s}_t$, the *agent* yields a policy $\pi(\mathbf{s}_t)$, with which the *action* q_t is made upon downloading the next segment. Then, a new *state* $\mathbf{s}_{t+1}$ is obtained through the probability $p(\mathbf{s}_{t+1}|\mathbf{s}_t, q_t)$.

Equation (1) provides a basic form of the *reward* function, in which α, β and γ are introduced to balance the trade-off between visual quality, fluctuation and rebuffing events. At the client sides, different users may have different preferences regarding these three metrics. Thus, α, β, and γ should vary across different users. We assume that $(\alpha_m, \beta_m, \gamma_m)$ are the values of α, β and γ for user m, encoding the QoE preference of this user. Accordingly, we formally define the immediate *reward* function of user m as follows:

$$r_m(q_{t-1}, \pi(\mathbf{s}_t), \phi_t) = \alpha_m \cdot q_t - \beta_m \cdot |q_t - q_{t-1}| - \gamma_m \cdot \phi_t. \quad (2)$$

Following the mechanism of DRL, the accumulated *reward* of multiple users from the initial *state* of $\mathbf{s}_1$ can be defined as follows:

$$R(\pi) = \underset{m \in \mathcal{M}}{E} \left\{ \sum_{t=1}^{t_{\max}-1} \eta^t r_m(q_{t-1}, \pi(\mathbf{s}_t), \phi_t) \right\}. \quad (3)$$

In (3), η and $t_{\max}$ is the *discount factor* and the maximum number of time steps, while $\mathcal{M}$ is the set of all users. Since optimizing the accumulated *reward* of one user preference is single-task DRL, optimizing that of multi-user preferences in (3) can be seen as multi-task DRL. However, it is intractable to directly optimize (3) due to the conflict preferences on the different QoE metrics [16]. In the next section, the MLMP framework is proposed to tackle this deficiency.

3. MLMP FOR DASH ADAPTATION

Now, we present our MLMP approach for DASH adaptation, which is based on a recent multi-task DRL approach: meta-learning shared hierarchies (MLSH) [20].

3.1. Review of MLSH

As a meta-learning framework for simultaneously mastering more than one task, Frans *et al.* [20] proposed the MLSH approach for simulation robotic control. In MLSH, some of neural network parameters are updated during the interaction with a specific task and are re-initialized once switching to another task. The remaining parameters are shared and updated in the whole distribution of tasks. The former characterizes a stochastic policy, i.e., the master policy, while the latter characterizes a set of policies, i.e., the sub-policies. Consequently, by continuously sampling tasks from the distribution of related tasks, the master policy learns to switch sub-policies, and the sub-policies learn to generate final *actions* to execute in the *environment*. As a result, sub-policies gradually collect primitives, such as moving to different directions in the navigation task or selecting the correct goal in the bandit task.

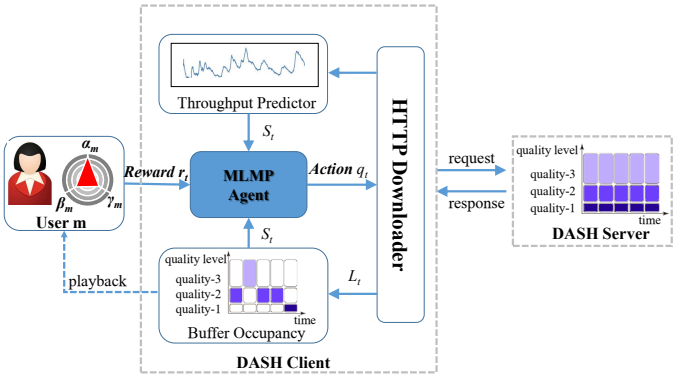

Fig. 1: DRL model for DASH adaptation at time step t.

3.2. Framework of MLMP

The overall framework is shown in Figure 2. The input is the network status at time step t, as depicted in Section 2.2. They are seen as the *state* of the DRL, which can be collected by a *state* extractor and then fed into the MLMP *agent*.

Inspired by MLSH [20], our *agent* consists of a two-level hierarchical DRL network. To be more specific, the low-level networks, i.e., the sub-policies (characterized by parameter ψ), learn to extract the common knowledge of the QoE optimization in DASH. On the other hand, the top-level network, i.e., the master policy (characterized by parameter θ), learns to perceive the current user preference. The objective of the master policy is to sequentially select a proper sub-policy according to the changing network condition, which guides our MLMP *agent* to the current preference. The selected sub-policy finally produces the *action* as the output of our MLMP approach, given the current *state* $\mathbf{s}_t$.

In MLMP, the input *state* for the master policy is the same as that for the sub-policies. Meanwhile, the master policy selects the sub-policies at a fixed frequency of N time steps. In other words, at each N time step, i.e., t = N, $2N$, ..., the master policy is activated, and then, the deep learning network observes the current *state* $\mathbf{s}_t$. The deep learning architecture of the master policy network is shown in Figure 2. It has two fully connected layers (size: 64) and one layer of long- and short-term memory (LSTM) cells (size: 64). Then, the LSTM feature (denoted by h_t) is processed by an-

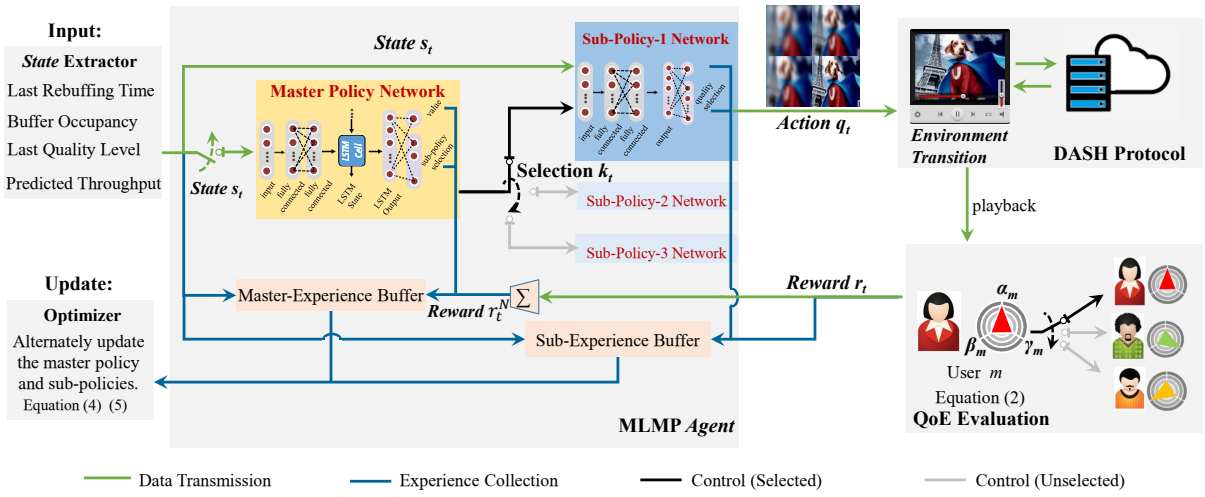

Fig. 2: Overall framework of the MLMP approach.

other fully connected layer (size: 4) to yield the final policy $\pi_\theta(\mathbf{s}_t)$. Here, π_θ is modeled by the probability distribution over the sub-policies (three in this paper) generated by the softmax function. Given the probability distribution, a sub-policy index $k_t \in \{1, 2, 3\}$ is obtained to select and activate the corresponding sub-policy. Subsequently, the selected sub-policy is activated at each time step to take as input the same *state* as that for the master policy. The deep learning architecture for learning the sub-policy is also shown in Figure 2, which includes two fully connected hidden layers (size: 64) and one fully connected output layer (size: 9) for yielding the *action*. Note that the LSTM cells are not included for learning the sub-policy, since the temporal correlation of the network throughput has been learned for the master policy. We denote the output quality of the selected sub-policy at time step t as $\pi_\psi(\mathbf{s}_t)$, which generates a probability distribution over all available segments with different quality levels. This is used to choose and download the next segment with appropriate quality for each user in DASH adaptation. Then, an instantaneous *reward* r_t is returned, based on the preference of the current user m (i.e., $(\alpha_m, \beta_m, \gamma_m)$). Subsequently, the experience, which refers to $\{\mathbf{s}_t, q_t, \mathbf{s}_{t+1}, r_t\}$, is stored in the master-experience buffer and sub-experience buffer. Finally, both the master policy and sub-policies can be updated by DRL training, which is discussed in the following section.

Algorithm 1: Algorithm for training MLMP.

Initialize parameter ψ for sub-policies.
for epoch $e = 1$ **to** E **do**
  Initialize parameter θ for the master policy.
  Sample preference $m \in \mathcal{M}$ and initialize the *environment state*.
  for iteration $w = 1$ **to** $T_{\text{master}} + T_{\text{joint}}$ **do**
    Initialize LSTM feature $h_0 = 0$, the current selected sub-policy index $k_0 = 0$ and the gathered *reward* $r_0^N = 0$.
    for time step $t = 1$ **to** $t_{\max} - 1$ **do**
      if t can be divided evenly by N **then**
        Extract $\mathbf{s}_t$ from the *environment*.
        Obtain policy $\pi_\theta(\mathbf{s}_t)$ and LSTM feature h_t using the master policy network with $\{\mathbf{s}_t, h_{t-N}, \theta\}$.
        Select sub-policy k_t according to the ϵ-greedy policy.
        Store the experience $\{\mathbf{s}_{t-N}, \mathbf{s}_t, h_{t-N}, k_{t-N}, r_t^N\}$ in the buffer and reset the gathered *reward* $r_t^N = 0$.
      end if
      Extract $\mathbf{s}_t$ from the *environment*.
      Obtain policy $\pi_\psi(\mathbf{s}_t)$ using the selected sub-policy network with $\{\mathbf{s}_t, \psi\}$.
      Select q_t according to the ϵ-greedy policy of $\pi_\psi(\mathbf{s}_t)$.
      Calculate $\mathbf{s}_{t+1}$ with regard to q_t and $\mathbf{s}_t$.
      Estimate the instantaneous *reward* r_t through (2) for q_t.
      Update the gathered *reward*: $r_t^N \leftarrow r_{t-1}^N + r_t$.
      Store the experience $\{\mathbf{s}_t, \mathbf{s}_{t+1}, q_t, r_t\}$ in the buffer.
    end for
    Update parameter θ according to (4) and (5).
    if $w > T_{\text{master}}$ **then**
      Update parameter ψ according to (4) and (5).
    end if
  end for
end for
Return: The trained parameter vector: $\{\theta, \psi\}$.

3.3. DRL training for MLMP

This section presents how to train and update the DRL model for selecting segments with different quality in DASH adaptation. In the multi-user scenario, we randomly sample a user m from $\mathcal{M}$ at each iteration in succession to estimate the *reward* r_t. Since the sub-policies aim at learning to obtain the common knowledge for the QoE optimization, they are shared for the preferences of different users. In other words, parameter ψ is optimized across all preferences without any re-initialization on ψ. Unlike the sub-policies, the master policy attempts to learn a preference-specific policy, such that an *agent* can be guided to be adaptive to the current preference. That is, parameter θ is trained for each preference and re-initialized until a new preference occurs. Moreover, we collect the *reward* r_t from *environment* for the master policy for the subsequent N time steps. Finally, the instantaneous *reward* r_t and the gathered *reward* $r_t^N = \sum_{t'=t}^{t+N-1} r_{t'}$ are used to update parameters ψ and θ, respectively.

Inspired by [20], we divide the update process into two sub-processes as follows: a guidance process (denoted by T_{master}) to only update the master policy and a joint process

Table 1: Performances on the three QoE metrics for our MLMP and D-DASH approaches.

	User-1				User-2				User-3			
	Quality (SSIM)	Fluctuation (Delta SSIM)	Rebuffing (Frequency)	Overall QoE	Quality (SSIM)	Fluctuation (Delta SSIM)	Rebuffing (Frequency)	Overall QoE	Quality (SSIM)	Fluctuation (Delta SSIM)	Rebuffing (Frequency)	Overall QoE
MLMP	0.974	**0.004**	0.008	**0.925**	0.969	0.012	**0.002**	**0.900**	**0.976**	0.007	0.009	**0.971**
D-DASH (MLP)	0.974	0.009	0.005	0.875	0.969	0.008	0.004	0.861	0.974	0.007	0.004	0.970
D-DASH (LSTM)	0.974	0.006	0.006	0.905	0.968	0.004	0.008	0.794	0.973	0.007	0.004	0.969
Ratio	1.00	0.042	0.009		1.00	0.023	0.047		1.00	0.003	0.001	

(T_{joint}) to update both the master policy and the sub-policies. To this end, both the instantaneous and gathered *reward* need to be optimized as follows:

$$\max_{\psi} \sum_{t=1}^{t_{\max}-1} \eta^t r_t + \max_{\theta} \sum_{t'=1}^{\lfloor t_{\max}/N \rfloor - 1} \eta^{t'} r_{t' \cdot N}^{N}. \quad (4)$$

Finally, based on (3), the overall optimization formulation can be rewritten as optimization over the distribution of multi-user preferences as follows:

$$\max_{\psi} \mathop{E}_{m \in \mathcal{M}} \left\{ \sum_{t=1}^{t_{\max}-1} \eta^t r_t + \max_{\theta} \sum_{t'=1}^{\lfloor t_{\max}/N \rfloor - 1} \eta^{t'} r_{t' \cdot N}^{N} \right\}. \quad (5)$$

Consequently, parameters θ and ψ can be learned in MLMP for optimizing the QoE of multi-user preferences. Algorithm 1 summarizes the procedure of training MLMP in DASH adaptation.

4. SIMULATION RESULTS

4.1. Simulation Settings

Following [9], we use the extensive trace-based simulation to evaluate the performance in DASH adaptation, in which the realistic DASH *environment* is modeled and the trace is from the realistic video dataset. We set the visual quality q_t (also in terms of SSIM) to be 9 levels and quantify the network throughput by 8 discrete levels with 50% switching probability. Refer to [9, 11] for other more details about DASH simulation and network throughput setting. Note that we use these same settings for a fair comparison with [11].

In our MLMP model, we set $N = 20$ time steps as the time period that the master policy selects a new sub-policy. For each iteration, the *agent* interacts with the *environment* until $t_{\max}$ (2000 in this paper) is reached for learning the experience. During the iterative interaction, we use the proximal policy optimization (PPO) algorithms [21] to update the parameters θ and ψ, for faster convergence. We follow [20] to set other hyperparameters of the DRL model.

In our simulation, we implement two versions of D-DASH [11], i.e., D-DASH with a multi-layer perception network (MLP) and D-DASH with an LSTM network. For more details about D-DASH (MLP) and D-DASH (LSTM), refer to [11]. Finally, to evaluate the performance on optimizing the QoE preferences of multiple users, we randomly sample one preference at each epoch. We denote the QoE preference of user m as $\mathbf{p}_m = (\alpha_m, \beta_m, \gamma_m)$.

4.2. Performance Evaluation of MLMP

Now, we evaluate the performance of our MLMP approach in satisfying multi-user preferences regarding the three QoE metrics, i.e., visual quality, fluctuation and the frequency of rebuffing events. For a fair comparison, we train MLMP, D-DASH (MLP) and D-DASH (LSTM) with the same number of time steps $t_{\max}$ in each episode. Since $(1, 2, 50)$ represents a general preference in [11], we use three typical QoE preferences with different bias : $\mathbf{p}_1 = (1, 10, 50)$ for user 1 who cares for fluctuation; $\mathbf{p}_2 = (1, 2, 250)$ for user 2 who is intolerable to rebuffing events; and $\mathbf{p}_3 = (5, 2, 50)$ for user 3 who favors the high instantaneous visual quality.

Table 1 tabulates the results of the three QoE metrics and the overall QoE for our and D-DASH approaches. Specifically, we measure the visual quality by the average SSIM value of all video segments. The fluctuation is assessed via the difference between the SSIM values of two successive segments. Rebuffing events are represented by the frequency of rebuffing. As shown in Table 1, our MLMP approach performs better than the other two D-DASH approaches in terms of the overall QoE scores for all three users, especially for user 1 and user 2. Meanwhile, Table 1 also reports the ratios of the weighted scores of different metrics to the weighted visual quality score. We can see from this table that our MLMP approach performs better in the QoE metric with the higher ratio. For example, user 1 does not like quality fluctuation (its ratio is 0.042) and pays less attention to rebuffing events (its ratio is 0.009). Accordingly, MLMP achieves considerably better performance in quality fluctuation than D-DASH. The cost is the loss of worse rebuffing events. This indicates that our MLMP approach is "good at" satisfying the multi-user preferences on the three QoE metrics. In addition, among the three approaches, our MLMP obtains the minimum standard deviation for all three users in terms of fluctuation, rebuffing events and quality, which demonstrates the robustness.

5. CONCLUSION

In this paper, we have proposed the MLMP approach for optimizing the multi-user QoE in DASH adaptation via generating the *actions* of selecting the proper video segments. First, we formulated optimizing the QoE of multi-user preferences on the three metrics, i.e., visual quality, fluctuation and rebuffing events, as a multi-task DRL problem. To produce a policy that is adaptive to the different preferences of multiple users, we presented the MLMP approach for generating a policy that is able to select the optimal video segments in DASH adaptation. Finally, the simulation results implied that MLMP is capable of meeting the QoE preferences of multiple users and is superior to other state-of-the-art approaches. The future work would be to quantify the diverse QoE preferences of multiple users or assess the QoE more accurately.

6. REFERENCES

[1] V. N. Index, "Cisco visual networking index: Global mobile data traffic forecast update, 2016-2021 white paper."

[2] T. Stockhammer, "Dynamic adaptive streaming over http–: standards and design principles," in *Proceedings of the second annual ACM conference on Multimedia systems*. ACM, 2011, pp. 133–144.

[3] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive," *IEEE/ACM Transactions on Networking (TON)*, vol. 22, no. 1, pp. 326–340, 2014.

[4] T. Y. Huang, R. Johari, N. Mckeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: evidence from a large video streaming service," in *ACM Conference on SIGCOMM*, 2014, pp. 187–198.

[5] Y. Sun, X. Yin, J. Jiang, V. Sekar, F. Lin, N. Wang, T. Liu, and B. Sinopoli, "Cs2p: Improving video bitrate selection and adaptation with data-driven throughput prediction," in *Proceedings of the 2016 ACM SIGCOMM Conference*. ACM, 2016, pp. 272–285.

[6] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "Bola: Near-optimal bitrate adaptation for online videos," in *INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, IEEE*. IEEE, 2016, pp. 1–9.

[7] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over http," in *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4. ACM, 2015, pp. 325–338.

[8] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[9] F. Chiariotti, S. D'Aronco, L. Toni, and P. Frossard, "Online learning adaptation strategy for dash clients," in *Proceedings of the 7th International Conference on Multimedia Systems*. ACM, 2016, p. 8.

[10] M. Claeys, S. Latré, J. Famaey, T. Wu, V. Leekwijck, D. Turck *et al.*, "Design of a q-learning-based client quality selection algorithm for http adaptive video streaming," in *Proceedings of the 2013 Workshop on Adaptive and Learning Agents (ALA), Saint Paul (Minn.), USA*, 2013, pp. 30–37.

[11] M. Gadaleta, F. Chiariotti, M. Rossi, and A. Zanella, "D-dash: a deep q-learning framework for dash video streaming," *IEEE Transactions on Cognitive Communications & Networking*, vol. PP, no. 99, pp. 1–1, 2017.

[12] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with pensieve," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. ACM, 2017, pp. 197–210.

[13] L. Yu, T. Tillo, and J. Xiao, "Qoe-driven dynamic adaptive video streaming strategy with future information," *IEEE Transactions on Broadcasting*, vol. PP, no. 99, pp. 1–12, 2017.

[14] I. Ketykó, K. De Moor, T. De Pessemier, A. J. Verdejo, K. Vanhecke, W. Joseph, L. Martens, and L. De Marez, "Qoe measurement of mobile youtube video streaming," in *Proceedings of the 3rd workshop on Mobile video delivery*. ACM, 2010, pp. 27–32.

[15] R. K. Mok, E. W. Chan, X. Luo, and R. K. Chang, "Inferring the qoe of http video streaming from user-viewing activities," in *Proceedings of the first ACM SIGCOMM workshop on Measurements up the stack*. ACM, 2011, pp. 31–36.

[16] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," *Psychology of learning and motivation*, vol. 24, pp. 109–165, 1989.

[17] T. Zhao, Q. Liu, and C. W. Chen, "Qoe in video transmission: A user experience-driven strategy," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 285–302, 2017.

[18] C. Chen, L. K. Choi, G. de Veciana, C. Caramanis, R. W. Heath, and A. C. Bovik, "A dynamic system model of time-varying subjective quality of video streams over http," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3602–3606.

[19] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity." *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[20] K. Frans, J. Ho, X. Chen, P. Abbeel, and J. Schulman, "Meta learning shared hierarchies," *arXiv preprint arXiv:1710.09767*, 2017.

[21] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.