applied sciences

MDPI

Article

A Deep Reinforcement Learning Quality Optimization Framework for Multimedia Streaming over 5G Networks

Alberto del Río [1,*], Javier Serrano [1], David Jimenez [1], Luis M. Contreras [2] and Federico Alvarez [1]

[1] GATV Research Group, Signals, Systems and Radiocommunications Department, Universidad Politécnica de Madrid, 28040 Madrid, Spain

[2] Global CTIO Unit, Telefónica I+D, 28050 Madrid, Spain

* Correspondence: arp@gatv.ssr.upm.es

Abstract: Media applications are amongst the most demanding services. They require high amounts of network capacity as well as computational resources for synchronous high-quality audio–visual streaming. Recent technological advances in the domain of new generation networks, specifically network virtualization and Multiaccess Edge Computing (MEC) have unlocked the potential of the media industry. They enable high-quality media services through dynamic and efficient resource allocation taking advantage of the flexibility of the layered architecture offered by 5G. The presented work demonstrates the potential application of Artificial Intelligence (AI) capabilities for multimedia services deployment. The goal was targeted to optimize the Quality of Experience (QoE) of real-time video using dynamic predictions by means of Deep Reinforcement Learning (DRL) algorithms. Specifically, it contains the initial design and test of a self-optimized cloud streaming proof-of-concept. The environment is implemented through a virtualized end-to-end architecture for multimedia transmission, capable of adapting streaming bitrate based on a set of actions. A prediction algorithm is trained through different state conditions (QoE, bitrate, encoding quality, and RAM usage) that serves the optimizer as the encoding values of the environment for action prediction. Optimization is applied by selecting the most suitable option from a set of actions. These consist of a collection of predefined network profiles with associated bitrates, which are validated by a list of reward functions. The optimizer is built employing the most prominent algorithms in the DRL family, with the use of two Neural Networks (NN), named Advantage Actor–Critic (A2C). As a result of its application, the ratio of good quality video segments increased from 65% to 90%. Furthermore, the number of image artifacts is reduced compared to standard sessions without applying intelligent optimization. From these achievements, the global QoE obtained is clearly better. These results, based on a simulated scenario, increase the interest in further research on the potential of applying intelligence to enhance the provisioning of media services under real conditions.

Keywords: deep reinforcement learning; 5G; adaptive multimedia; A2C; quality of experience; agent-based simulation; smart optimization

check for updates

Citation: del Río, A.; Serrano, J.; Jimenez, D.; Contreras, L.M.; Alvarez, F. A Deep Reinforcement Learning Quality Optimization Framework for Multimedia Streaming over 5G Networks. *Appl. Sci.* **2022**, *12*, 10343. https://doi.org/10.3390/app122010343

Academic Editors: Manuel Herrera, Marco Pérez-Hernández, Rakib Abdur and Jun Hong

Received: 15 September 2022
Accepted: 9 October 2022
Published: 14 October 2022

1. Introduction

In recent years, there has been a proliferation of new services owing to improvements in the communication back-end. That is, the 5th generation networks (5G) stand out over their generation predecessor exhibiting an increase in bandwidth and a decrease in latency. However, the characteristic that allows to expand the new ecosystem of services is the capacity of connected devices to the same network cell.

For general content optimization, the task requires the active participation of background services to ensure minimum QoE conditions, trying to maximize it according to the network parameters. This service works in the background, monitoring all the parameters necessary for modeling and characterizing the environment. The environment states are generated by consuming data from the conditions of the environment, and the optimizer predicts the best action to execute. That is, by analyzing the environment data, the optimizer applies a policy of actions that seeks to maximize the QoE of the service. The complete system follows the reference control for self-adaptive systems such as MAPE (Monitor-Analyze-Plan-Execute) [1].

In this paper, Deep Reinforcement Learning (DRL) algorithms are considered for the application of AI in the automation of network operations to ensure the provision of QoE. Specifically, this paper shows the work that has been carried out in which the algorithm applies a series of actions that consist of establishing the target bitrate, maximizing the QoE based on the state of the network. It is used to prevent bothersome situations and bottlenecks from spreading throughout the network and to localize its sources to enable fast and precise corrective actions. Reinforcement Learning (RL) addresses the problem by training agents on which actions should be taken, evaluating the states and actions based on defined reward functions that reflect the idea of quality and efficiency that we want to translate into our automated system. These algorithms include two Neural Networks (NN) in the form of agents to evaluate the correct functioning of training and validation. That is, thanks to Deep Learning (DL) techniques, we can value and emphasize the actions that the RL technique is performing. As training progresses, these corrections will become more accurate. NNs are based on Convolutional Neural Networks (CNN) which have shown high performance in training processes that require a large volume of data. These NNs map the environment observations (inputs) to the bitrate decision (output).

This work investigates the potential application of an algorithm called Advantage Actor–Critic (A2C), as it is one of the leading algorithms in DRL, allowing a multimedia service application through multiple agents. These parallel agents perform the training synchronously, that is, the update of the results of the central node is carried out when all the agents have finished their episode. However, this research focused on the use of a single agent, to enable training in standard resource environments.

The work presented in this paper contains the initial design and test of a self-optimized cloud streaming vertical, and finally its validation in the laboratory prior to deployment in a real environment. The work is presented as follows. Section 2 introduces the multimedia content used in this work, the components of the service, an explanation of the workflow, and the optimizer theoretical background. The previous sections are accomplished in Section 3, exposing the AI algorithms applied and the results obtained. Finally, Section 4 ends with the conclusions of the work and future work expansion.

Related Work

The communications ecosystem is constantly growing. It has facilitated new remote business, both in terms of versatile innovation and the entry of new members. That is, new industries that have seen an economic benefit from the possibility of serving millions of devices efficiently. One of the hottest research topics is applying intelligence to the network. Intelligence can be applied mainly in two stages: application level (closer to end-user) or network level (infrastructure). There are several ways to use Artificial Intelligence (AI) at the application level. With regard to the application level, our work relies on image analysis tools and the use of relevant metrics, e.g., Quality of Experience (QoE). QoE, in the video domain, is a subjective measure used in different works [2] as an estimate of perceived quality based on the analysis of image characteristics.

However, the future of communications calls for a virtualized world based on microservice systems. This underscores the importance of adding smart services also at the network level. In the work [3], a cloud architecture is analyzed on which to deploy a service virtualization platform (SVP) to facilitate the development, deployment and operation of media services on 5G networks. These edge-to-cloud architectures represent the new deployment paradigm in which the different services are assembled and interconnected. The simultaneous application of 5G frameworks with QoE modeling in multimedia services enables works such as [4], where it subjectively compares not only the results obtained in the laboratory, but also the empirical results gathered during the testbed.

The multimedia adaptation of streaming requires the full participation of an intelligence. That is, a service capable of monitoring and acting accordingly. Certain works such as [5] began these studies with the application of RL to content adaptation. In more recent studies, thanks to 5G tools and QoE estimation, the possibilities of applying RL algorithms to optimize QoE were analyzed. The first of them [6] focuses on reducing buffering times in streaming, while other works advocate an adaptation of the bitrate with different videos scaled in bitrate [7] with the same objective. Both works aim to improve the QoE of users avoiding video stalls in off-line content consumption environments. However, QoE is estimated through other parameters, such as bitrate and buffer size. These scenarios are not meant for live content, but for on-demand content consumption.

The first major difference with respect to our work relates to the influence that network problems can have on QoE. In this case, QoE is affected by stalls in the video experience. The work estimates the reduction of QoE based on the times when a rebuffering process is needed. On the other hand, we present a work that actually measures the QoE, with the objective of not only guaranteeing the reduction of video artifacts, in our case consisting of blockiness, block loss, and blurring, but to also improve the experience by providing the highest possible QoE.

The scope of remote content production is one of the keys to analyze in terms of service optimization. Although 5G networks can support high binary rates, there are cases where conditions are not adequately met and service experience is affected. For example, in remote production cases [8], dedicated point-to-point connections are established between the content provider and the central server to try to guarantee the best possible conditions. However, due to network saturation, this point-to-point connection may not be in the most optimal conditions. The result is poor end-user QoE. The proposed idea for solving similar problems is to apply optimization techniques in which, through AI techniques, we can estimate which are the most appropriate corrective actions to maximize QoE for the end-user.

2. Methodology

This section presents the analyzed multimedia content, integration of system components, and workflow simulation.

2.1. Content Sampling

The purpose of evaluating the content to be transmitted is to analyze the viability of the transmission. The concept of feasibility lies in the fact that we can guarantee a high QoE. In recent years, Ultra-High-Definition (UHD) content has proliferated [9,10] due to the increase in devices with the ability to reproduce it. However, increasing the resolution does not always imply an improvement in image quality. Quality measurement is generally evaluated by bitrate [11]. It is common for certain occasions in content production, even in UHD, to not use high bitrates.

The work is based on a preliminary study on remote production [8]. Although the work can be implemented in any video, in our case, the content is not generated in the research laboratory. Using uncontrolled content requires higher AI training, but is more useful in terms of demonstration purposes. Multimedia is produced by local German television [12] and is received through a radio antenna placed in the laboratory to receive multimedia content. It is received in raw format, but using image processing equipment, we encode the video using the H.264 codec. Its content schedule is varied and will serve as a demonstration of image quality using traditional types of video.

In this demo, we are looking for high-quality streaming with playability on the widest range of devices possible. Instead of looking for UHD content and encoding it to lower the bitrate, we used high-bitrate Full-High-Definition (FHD) content. In this case, we are referring to an average bitrate of 10 Mbps in 16:9 format. Normally, the content at this resolution transmitted in the Spanish nation is transmitted at around 4 Mbps, so we can see that our transmission offers a higher quality of content.

Traditional content is reproduced at 24 frames per second (fps) in Europe [13], which, with 10 Mbps streaming, makes it easier for the application components of the service to be deployed in any instance. This work proposal aims to guarantee an optimal QoE under every network condition, using standard dedicated server components. Increasing the bitrate and resolution further requires specific hardware components (graphics cards) to support lossless real-time encoding.

2.2. Application Layer

The environment is developed by components to implement the application of intelligence on the streaming service. The following elements are necessary to meet the service requirements, that is, the satellite content receiver, content transmitter, video quality analyzer, and intelligence optimizer for application control. The fundamental components are the following:

- Satellite. The antenna receives the content via satellite. This content is processed (received in raw format, converted to H.264) and relayed within the laboratory.
- Optimizer. In charge of optimizing the service. Provides intelligence to the entire use case through DRL algorithms. It receives data from the communication bus to obtain probe data and, in addition, transcoder status values. These values are used to establish the set of states so that the current situation of the environment is monitored and the optimizer has sufficient data to predict actions accordingly.
- Transcoder. Performs bitrate control tasks, adjusting the streaming according to the optimizer's needs. It uses the H.264 encoder [14] for image processing, since the content received in the media production is based on the same codec. Apart from being one of the most used codecs in the industry, we maintain the same encoding type, which makes it easy to avoid changes in the content. This component is based on open source services, including multimedia libraries such as FFmpeg [15].
- Receiver. Multi-service component that allows both content visualization and analysis thanks to an integrated quality probe. This component simulates an end user on stage who views and evaluates the content in real time. The visualization part is based on a multimedia playback framework such as VLC [16], which is capable of receiving streaming content. The content analysis is performed by the quality probe through a non-reference analysis, determining the QoE of the received content through internal AI capabilities. The QoE metric is based on the Mean Opinion Score (MOS), used in many standardized subjective video quality assessment methodologies and ITU-T recommendations such as P.910 [17], among others. Given the subjectivity of the metric, the value scale was trained in an environment controlled by a complete evaluation procedure by different users with different content.

2.3. Workflow Simulation

Prior to development in a real scenario, it is necessary to not only test the workflow in a controlled environment, but also to pretrain the DRL optimizer algorithm. To perform this simulation, the proposed scenario (Figure 1) is deployed over a non-orchestrated network. Since one of the main objectives of this simulation is to test the optimization feature of a real scenario, we need to generate network problems in the workflow. We consider a network to be the communications infrastructure itself over which the multimedia content will flow, so the initiative is to create possible corruptions in the network and, therefore, in the reception of the content to train the algorithm against possible future inconveniences.

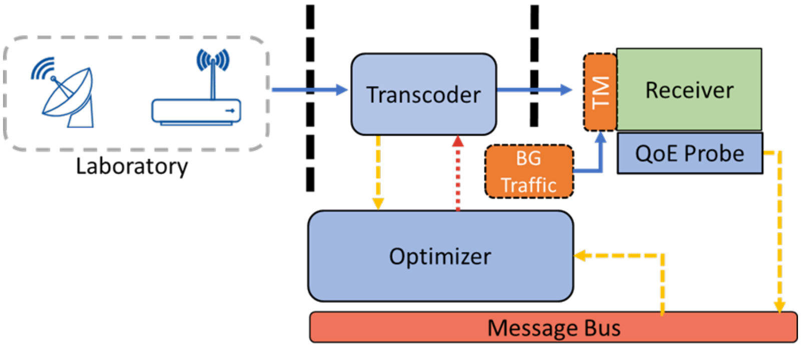

Figure 1. Simulated general optimization workflow.

In this case, it is considered an in-network adaptation to a given network bottleneck. For this purpose, a Traffic Manager (TM) acts as a controlled environment composed of a network interface, where there is a limit on the link bandwidth. The network interface will act as an input to the point that contains the components where the network problem could affect the service and possibly obtain a network bottleneck to train the algorithm under all conditions.

To create network problems, it is necessary to deploy a background traffic generator that will simulate different hosts that consume bandwidth and compete for the limited bandwidth available in the TM. This tool is based on the well-known open-source tool iPerf [18]. The bandwidth transmitted by this background service is generated randomly, so that both the network and the optimizer have no control over these values, simulating a close to reality networking problems environment.

The objective is to perform an optimization of the multimedia service, evaluating the QoE in the receiver, and optimizing the bitrate. The content will be received at the end of the flow, which will feed the optimizer to predict the actions that enable better QoE in the environment. These actions are reflected in the transcoder, which will adapt the bitrate to the actions taken. During training, the receiver is expected to have low-quality video at certain times, including image artifacts such as block loss. By using background traffic, the environment will face cases where the conditions are extremely poor, and in initial phases the optimizer will not know how to properly act (learning from different conditions). More details about this problem are presented in Section 2.4. Once trained, we can assess whether, through training, we have managed to alleviate video problems caused by various conditions in the network, obtaining the best possible QoE.

2.4. Algorithm

The AI service applied in these tests is based on real-time optimization. Information for the optimizer is obtained from both the probe and the transcoder, which completes the set of states of the environment. This development at the code level allows us to represent the environment and to be able to run actions on the components of the service.

Generally, AI algorithms require a dataset to create a model. The service we present is created in real time with the environment data, from the quality measurements offered by the probe to the transmission values of the transcoder. The techniques capable of applying intelligence are those consistent with RL, through which we expand thanks to the use of DL. This combination of techniques is called DRL.

Optimization of the service is performed on the quality measured in the probe, exerting actions on the transcoder. Based on the environment values, the optimizer analyzes the policies and makes the most optimal decision, even seeking to solve possible bottlenecks in the network.

DRL algorithms combine reward maximization techniques through action policies, with the addition of NN (Actor and Critic) as agents that interact with the environment to improve training efficiency. Based on the effects of the actions, the environment returns a reward. The algorithm developed and trained for this case is called A2C [19]. A high-level diagram is presented in Figure 2.

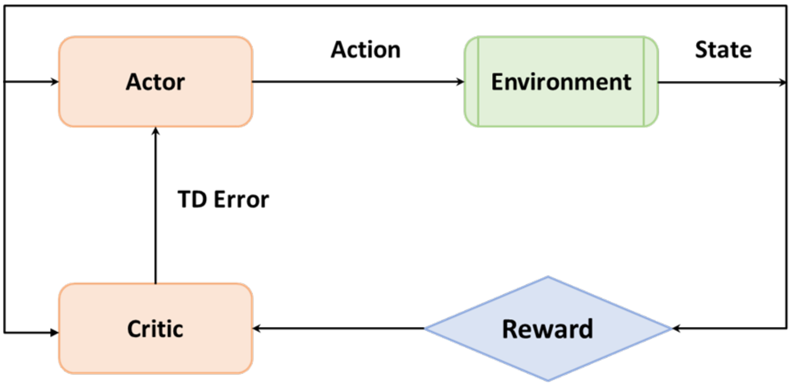

Figure 2. A2C architecture overview.

In recent years, new RL algorithms have proliferated, but among them, A2C has been one of the most prominent due to the use of NNs with RL techniques. We emphasize that its choice for this work was due to the possibility of training agents in parallel, giving the option as future work of expanding the number of parallel sites for more efficient training, as well as the final staging of the trained model.

Going into detail, the Advantage Function informs the algorithm how well (or poorly) an agent has behaved when performing a certain action compared to previous actions. This also helps inform the model that training is appropriate and that no further steps are required. That is, find a balance between a possible training bias while maintaining a data variance.

The Actor–Critic is a set of DRL algorithms that combine the ideas of algorithms focused on obtaining the value function (and the action-value function) and those of maximizing the optimal policy function (check Equations (1) and (2) below). These equations are called Bellman equations [20]. They help agents assess the current environment situation and not wait for long-term results. These equations introduced the state value function (V_π) as the expected return over all possible actions for a specific state, and the state-action value function (q_π) as the expected return over all possible actions in some state, both following the policy π. Other important variables from these equations are the expected value over the policy distribution ($\mathbb{E}_\pi$), the set of states and the current one (S and s), the set of actions and the current one (A and a), the accumulated reward (R_t), the returned reward for a specific trajectory (G_t), and the discount factor (γ).

$$\begin{aligned} V_\pi(s) &= \mathbb{E}_\pi[R_t + \gamma G_{t+1} | S_t = s] \\ &= \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)[r + \gamma v_\pi(s')] \\ q_\pi(s,a) &= \mathbb{E}_\pi[R_t + \gamma G_{t+1} | S_t = s, A_t = a] \\ &= \sum_{s',r} p(s',r|s,a)[r + \gamma \sum_a \pi(a|s) q_\pi(s',a')] \end{aligned} \tag{1}$$

$$\begin{aligned} v_*(s) &= \max_\pi v_\pi(s) \\ q_*(s,a) &= \max_\pi q_\pi(s,a) \end{aligned} \tag{2}$$

2.5. Neural Networks

Two NNs are included in the agent's position, one being the Actor, who is responsible for taking the actions evaluating the entire system, and the other NN corresponding to the Critic, which evaluates (criticizes) the actions taken by the Actor (configuration in Table 1). Both NNs share the structure of CNN architecture as input, with fully connected dedicated layers as output per network. In addition, there are some hidden layers with ReLu activation. The input of both neural networks consists of the set of states, as well as the number of states to hold in each training step (states batch).

As a difference between networks, the Actor network ends in a Dense network with SoftMax activation layer, to select the action to be taken based on the policy values. Instead, the Critic network follows a linear activation to estimate the value function for a given state. It is important to note that the Critic network role is to evaluate the actions of the Actor, helping in terms of training. Figure 3 displays the architecture used in training.

Table 1. Neural Networks hyperparameters.

Model	Input	Learning Rate	Output Activation	Output
Actor	5 × 8 (states × states batch)	0.01%	SoftMax	Action policies (Probabilities)
Critic	5 × 8 (states × states batch)	0.1%	Linear	V(s) (Value estimation)

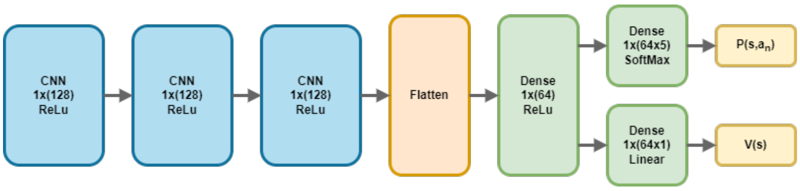

Figure 3. Neural Networks architecture implementation.

2.6. *Reinforcement Learning*

This algorithm was selected for the case evaluation due to the possibility of training numerous parallel agents in the environment, which would facilitate the possible expansion of new content reception sites. That is, according to the current workflow, it is a single site that receives and evaluates the content, but due to this algorithm, it would be easy in future work to expand the capacities with an indefinite number of parallel sites, limited only by the capacity of resources.

The streaming use case in which we train and apply the algorithm focuses on optimization of the streaming bitrate in order to obtain the best MOS received by the probe in the position of an end-user. The environment is defined by the parameters (states) that model the status of the use case workflow. These states are dependent on the received MOS, the emitted and received bitrate, the encoding quality, and the RAM usage of the transcoder. This set of states is chosen because they can appropriately reflect both the quality received and the current situation of the components.

To perform the optimization, and generally for all RL algorithms, the analysis of the environment provides a policy of actions. The set of actions should reflect the amount of options that the algorithm can produce. The set developed for this case consists of a selection of network profiles in which we configure the target bitrate to which the transmission must adapt. Actions are applied to the site transcoder via Rest API, changing the internal throughput of the streaming. The profile set consists of: 10 Mbps, 7.5 Mbps, 5 Mbps, 4 Mbps, 2.5 Mbps, 1 Mbps. Both states and actions are summarized in Table 2.

Table 2. Set of states and actions.

	Features
Environment States	Received MOS; Input Bitrate; Output Bitrate; Encoding Quality; RAM Usage
Set of Actions	10; 7.5; 5; 4; 2.5; 1 (Mbps)

Finally, the algorithm returns a reward based on the actions taken and the resulting states. This reward is calculated from a combination of specific reward functions, depending on the received streaming MOS; the emitted and received bitrate; the selected profile; and the smoothness (avoid very aggressive bitrate adjustments) of profile selection. First actions are randomly performed, and lower rewards will be returned. The optimizer will analyze possible actions and learn the best suitable option based on the information as it explores possible actions over time.

We present a detailed pseudocode of A2C in Algorithm 1 in which, the process behind how the algorithm models the optimal policy of actions of the NN actor is detailed step-by-step. That is, for a set of episodes, the algorithm performs a fixed number of steps for training. Each episode takes an action which returns a reward, and through the computation of the Advantage Function, the weights of the NN are updated (Critic network using Stochastic Gradient Descent).

Algorithm 1 Advantage Actor Critic (A2C)

```
Input Episodes size $\alpha$ , Steps size $\beta$
Output $\pi_{\theta_A}$, approximate optimal policy of the Actor network
  Initialize both Actor–Critic networks $\theta_{Act}$ and $\theta_{Cri}$
  for each episode do
      Initialize $s_t$
      for each step in the episode do
          Use the actor to get $\pi(\cdot|s_t)$, the probability of each $a$ in state $s_t$
          Take action $a_t \sim \pi(\cdot|s_t)$ and observe $r_t$, $s_{t+1}$
          Obtain $V_{\theta_C}(s_t)$ and $V_{\theta_C}(s_{t+1})$ using the Critic network
          Obtain $A(s_t, a_t) = R + \gamma V_{\theta_C}(s_{t+1}) - V_{\theta_C}(s_t)$
          Update the critic network: $\theta_C = SGD(A(s_t, a_t))$
          Update the actor network: $\theta_A$
  return $\pi_{\theta_A}$
```

3. Results and Analysis

Rewards were divided into different types to better reflect the assessment of the environment. Therefore, the total system reward is the sum of several independent rewards. Each of them represents the evaluation of a feature from the optimization system. In the reward creation process, we defined a range of importance between them so that the ones used to balance training do not outnumber those most important. The reward function that returns the maximum value is the MOS one (QoE metric), since it better represents the behavior of the system. The bitrate function also helps to determine possible problems in the flow, second in reward importance, by possible bitrate loss along the way. Therefore, in this bitrate reward function, we defined it as the difference between input and output bitrates.

In order to equilibrate the model, the other two rewards were developed as the development progressed in time. Smooth reward focuses on avoiding abrupt changes between profiles, while profile rewards overcomes conservative behavior for medium profiles. This is produced because medium profiles do not offer maximum reward for MOS, because of using lower bitrates, avoiding possible image problems. These reward functions included the predicted action parameter, the current one and the last one. In other words, a' represents the previous action, so that the current reward evaluates the evolution over the preceding profile, and a represents the action chosen at that step.

$$
\begin{aligned}
RewardMOS &= \pm 2 * e^{1.5*(\pm(MOS-2.5))} \\
RewardBitrate &= -e^{2*(1+\frac{|bitrate_{in}-bitrate_{out}|}{|bitrate_{in}|-|bitrate_{out}|})} \\
RewardSmooth &= 12 * \ln \frac{|a'-a+2|}{|a'+1|-|a+1|} \\
RewardProfile &= 2^{4-a} \\
TotalReward &= Rwd\_MOS + Rwd\_Bitrate + Rwd\_Smooth + Rwd\_Profile
\end{aligned}
\tag{3}
$$

Finally, to offer some conclusions and based on previous test training, we present the model that returns the best increase of the reward function while maintaining a good QoE. Model validation is unique to the simulated environment, but it would be easy to apply it to other clients in another environment, as long as the same data are included for the state set, and the environment rewards correctly define the new environment. In addition, the fundamental requirement is that the action policy is subject to the network profiles mentioned above with a defined set of bitrates.

An important point is to define the number of iterations to average the rewards for each step. It marks the line when individual iterations reward is accumulated. Each training step was set by 25 iterations, allowing enough accumulated reward to differentiate progress over time and offer learning to the model. In this configuration, iterations represent the optimizer's actions cycle and the effect produced on the system. In total training, 12,500 iterations were performed, which took approximately 24 h.

Figure 4 shows how negative values are returned on average in the initial steps, but after training, the average reward function improved by more than 300% from the beginning, indicating that the algorithm optimizes properly following the reward function, which delivers a better score with training.

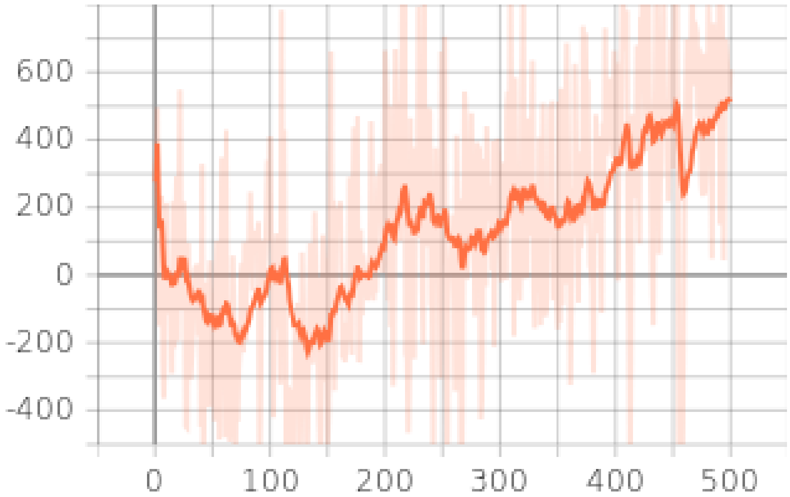

Figure 4. Training reward chart.

Reward graphic presents sufficient justification for the demonstration of improvement over training. Good quality represents higher reward, but there is another aspect that can decrease the general reward: image artifacts. When the probe analyzes the content and detects enough image artifacts for bad visualization, the returned MOS is 1. Figure 5 on the left represents the histogram values of MOS during training. Although there is more representation for higher quality values in general, there is a large percentage of minimum quality values.

In contrast, Figure 5 on the right side represents the same graphic histogram, but using the optimizer. Not only has the distribution moved to higher values, but there are also a few 1.0 MOS values. This demonstrates that the optimizer has learned to adapt the bitrate under the conditions to improve the general QoE. From a very poor QoE range (1–2 MOS) in the figure of almost 25% of the total, the optimizer has reduced the percentage to 10%. Finally, 90% of the values obtained represent good quality (higher than 3.5) from the 65% obtained without optimizer.

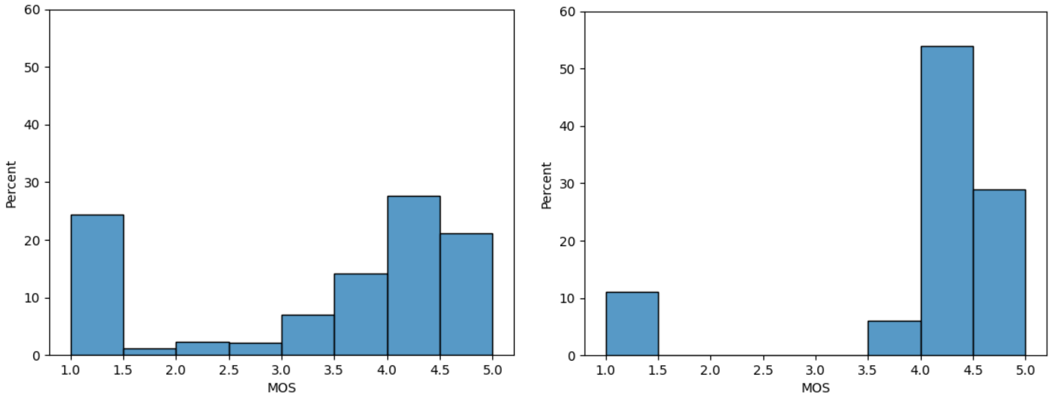

Figure 5. MOS distribution: (**left figure**) Values obtained for the application without optimizer. (**right figure**) Optimizer applied to the service.

With the actual experimental setup, the background traffic generator configuration creates uncontrolled randomness on the impairments. This leads to a lack of sufficient bandwidth for the proper provision of the content service during minimal time slots. As a result of this, image artifacts are generated, and the QoE (MOS) results in very low scores, despite adjusting the streaming bitrate according to the profiles. This justifies why there are still low MOS values (10%) with the optimizer enabled.

Analysis

The presented results compare the non-optimized case with respect to the optimized case. It is important to note that, in contrast to the existing methods in the literature [6,7], this work is built on a simulation that randomly introduces temporal network impairments. This means that network issues are not stationary and there is no indication when they can occur. Moreover, it demonstrates the ability to adapt to the problems of the proposed DRL-based optimizer.

In addition, in this dynamic environment, network conditions can also change with the availability of more resources. This fact is leveraged by the optimizer by increasing the profile, leading to a higher QoE. After activation of the optimizer, not only is evident the reduction of very poor quality video segments (from 25% to 10%), but also the fact that 90% of video segments have a very good QoE (more than 3.5).

4. Discussion

This section includes the main conclusions obtained during the experimentation. Furthermore, future work is included on how to continue the line of research, based on the premises of this article.

4.1. Conclusions

The objective of the presented research was to analyze the potential use of intelligence in the service-layer architecture of new-generation communication networks for optimizing quality on multimedia services. The work presented in this paper demonstrates the possibility of developing a DRL model through which we can guarantee an adequate QoE. Having a series of previously configured profiles and managing the bitrate adaptation only as an action are enough to create a control pattern in content transmission systems. This is demonstrated from an uncontrolled setup with no previously available dataset for training.

The developed reward functions were sufficient to represent an optimization in the system. The initial training phases returned negative rewards due to their exploratory nature. However, the last moments of training offered high reward values. This demonstrated the optimizer ability to adapt to the various impairments situations induced, offering the best possible action under every condition. Finally, an improvement over the system itself is observed in the quality of the received image, that is, the MOS. During the training phase, there were 25% values for a bad MOS range. However, this percentage was improved using the optimizer, reducing up to 10% of the values. Furthermore, when the optimizer was enabled and trained 90% of the values were in the high-quality range of MOS.

However, this is still a simulated scenario prior to real deployment in a real 5G environment. Commercial ideas cannot be approached due to initial research. Furthermore, we demonstrate the optimization task with one single network link and one single optimized transcoder. Despite these drawbacks, we can confirm the overall good performance of combined RL and DL, making it possible to automate the optimization of multimedia content delivery over virtualized networks.

4.2. Future Work

One of the main points of future work is to adapt this work to parallel multisite optimization. The A2C algorithm performs synchronous training between all parallel training agents, so that, in case of problems with any transcoder or delay in sending information, other actions to be executed on the transcoder can be stopped until the completion of the update point of the delayed training agent. In principle, an upgrade to the A3C algorithm [21] with asynchronous training could solve this problem, each transcoder being independent in terms of training.

Furthermore, we consider only the QoE of the end user as the objective of the optimization process. In future work, the addition of resource consumption and energy efficiency should be considered to create a multi-objective vector for a more accurate optimization process.

Finally, the ability to optimize in parallel different streams opens a new research field, where we can consider different privileges per user (premium user, standard user, etc.). This consideration also opens the door to research from a business perspective, where we can apply different billing models depending on the consideration of the user and the optimization of the workflow.

This future work is planned to be integrated with research and innovation within B5G (5G and Beyond 5G) and 6G topics by adding intelligence to network management and optimization.

Author Contributions: All authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

5G	5th Generation
QoE	Quality of Experience
MOS	Mean Opinion Score
MEC	Multiaccess Edge Computing
AI	Artificial Intelligence
DRL	Deep Reinforcement Learning
RL	Reinforcement Learning
DL	Deep Learning
NN	Neural Network
CNN	Convolutional Neural Networks
A2C	Advantage Actor–Critic
MAPE	Monitor-Analyze-Plan-Execute
SVP	Service Virtualization Platform
FHD	Full-High-Definition
UHD	Ultra-High-Definition
FPS	Frames Per Second
TM	Traffic Manager
RAM	Random Access Memory
API	Application Programming Interfaces

References

1. Arcaini, P.; Riccobene, E.; Scandurra, P. Modeling and Analyzing MAPE-K Feedback Loops for Self-Adaptation. In Proceedings of the 2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, Florence, Italy, 18–19 May 2015; pp. 13–23.
2. López, J.P.; Martín, D.; Jiménez, D.; Menéndez, J.M. Prediction and Modeling for No-Reference Video Quality Assessment Based on Machine Learning. In Proceedings of the 2018 14th International Conference on Signal-Image Technology Internet-Based Systems (SITIS), Las Palmas de Gran Canaria, Spain, 26–29 November 2018; pp. 56–63.
3. Alvarez, F.; Breitgand, D.; Griffin, D.; Andriani, P.; Rizou, S.; Zioulis, N.; Moscatelli, F.; Serrano, J.; Keltsch, M.; Trakadas, P.; et al. An Edge-to-Cloud Virtualized Multimedia Service Platform for 5G Networks. *IEEE Trans. Broadcast.* **2019**, *65*, 369–380.
4. Nightingale, J.; Salva-Garcia, P.; Calero, J.M.A.; Wang, Q. 5G-QoE: QoE Modelling for Ultra-HD Video Streaming in 5G Networks. *IEEE Trans. Broadcast.* **2018**, *64*, 621–634.
5. Charvillat, V.; Grigoraş, R. Reinforcement learning for dynamic multimedia adaptation. *J. Netw. Comput. Appl.* **2007**, *30*, 1034–1058.
6. Cui, L.; Su, D.; Yang, S.; Wang, Z.; Ming, Z. TCLiVi: Transmission Control in Live Video Streaming Based on Deep Reinforcement Learning. *IEEE Trans. Multimed.* **2021**, *23*, 651–663.
7. Mao, H.; Chen, S.; Dimmery, D.; Singh, S.; Blaisdell, D.; Tian, Y.; Alizadeh, M.; Bakshy, E. Real-world Video Adaptation with Reinforcement Learning. *J. Abbr.* **2020**, *10*, 142–149.
8. Keltsch, M.; Prokesch, S.; Gordo, O.P.; Serrano, J.; Phan, T.K.; Fritzsch, I. Remote Production and Mobile Contribution Over 5G Networks: Scenarios, Requirements and Approaches for Broadcast Quality Media Streaming. In Proceedings of the 2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Valencia, Spain, 6–8 June 2018; pp. 1–7.
9. UHD: 4K Is Here and Now, but What about 8K? Available online: https://www.ibc.org/trends/uhd-4k-is-here-and-now-but-what-about-8k/3827.article (accessed on 27 July 2022).
10. Transparency Market Research. Available online: https://www.transparencymarketresearch.com/4k-content-market.html (accessed on 27 July 2022).
11. Xie, F.; Pourazad, M.T.; Nasiopoulos, P.; Slevinsky, J. Determining bitrate requirement for UHD video content delivery. In Proceedings of the 2016 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 7–11 January 2016; pp. 241–242.
12. Servus TV Deutschland. Available online: https://es.astra.ses/channels/servus-tv-deutschland (accessed on 3 August 2022).
13. A Beginner's Guide to Frame Rates. Available online: https://www.premiumbeat.com/blog/beginners-guide-to-frame-rates/ (accessed on 16 August 2022).
14. ITU. H.264: Advanced Video Coding for Generic Audiovisual Services. Available online: https://www.itu.int/rec/T-REC-H.264 (accessed on 10 June 2022).
15. FFmpeg: A Complete, Cross-Platform Solution to Record, Convert and Stream Audio and Video. Available online: https://ffmpeg.org/ (accessed on 10 June 2022).
16. VLC Media Player. Available online: https://www.videolan.org/vlc/ (accessed on 10 June 2022).

17. Union, Int. Telecommun. Subjective video quality assessment methods for multimedia applications. ITU-Recommendation P.910. 1999. Available online: https://www.itu.int/rec/T-REC-P.910 (accessed on 10 June 2022).
18. iPerf—The Ultimate Speed Test Tool for TCP, UDP and SCTP. Available online: https://iperf.fr/ (accessed on 10 June 2022).
19. Mehta, D. State-of-the-Art Reinforcement Learning Algorithms. *Int. J. Eng. Tech. Res.* **2020**, *8*, 717–722.
20. Dave, H. Understanding the Bellman Optimality Equation in Reinforcement Learning. Data Science Blogathon. Available online: https://www.analyticsvidhya.com/blog/2021/02/understanding-the-bellman-optimality-equation-in-reinforcement-learning/ (accessed on 3 August 2022).
21. Mnih, V.; Badia, A.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous Methods for Deep Reinforcement Learning. In *International Conference on Machine Learning*; PMLR: New York, NY, USA, 2016; Volume 48, pp. 1928–1937.