User-Centric Markov Reward Model on the Example of Cloud Gaming

Tobias Hossfeld
University of Würzburg
Chair of Communication Networks
Würzburg, Germany
tobias.hossfeld@uni-wuerzburg.de

Poul E. Heegaard
NTNU - Norwegian University of Science and Technology
Trondheim, Norway
poul.heegaard@ntnu.no

Martín Varela
Profilence
Oulu, Finland
martin@varela.fi

Michael Jarschel
Technische Hochschule Ingolstadt
Faculty of Computer Science
Ingolstadt, Germany
Michael.Jarschel@thi.de

***Abstract*—Markov reward models are commonly used in the analysis of systems by integrating a reward rate to each system state. Typically, rewards are defined based on system states and reflect the system's perspective. From a user's point of view, it is important to consider the changing system conditions and dynamicity while the user consumes a service. In this paper, we consider online cloud gaming as use case. Cloud gaming essentially moves the processing power required to render a game away from the user into the cloud and streams the entire game experience to the user as a high definition video. According to the available network capacity, the video streaming bitrate is adapted. We conduct experiments on Google Stadia and provide a Markov model based on the measurement results to investigate a scenario where users are sharing a bottleneck link.**

The key contributions are proper definitions for (i) system-centric reward and (ii) user-centric reward of the cloud gaming model, as well as (iii) the analysis of the relationships between those metrics. Our key result allows a simple computation of the user-centric rewards. We provide (iv) numerical results on the trade-off between user-centric rewards and blocking probabilities to access the online cloud servers. We use Kleinrock's power metric to identify operational points. This work gives relevant and important insights in how to integrate the user's perspective in the analysis of Markov reward models and is a blueprint for the analysis of other services beyond cloud gaming.

***Index Terms*—Markov reward model, cloud gaming; Quality of Experience (QoE), user-centric reward**

I. Introduction

It is common practice to use Markov reward models to analyze the utility of a system. Markov models provide the system state probabilities $x(i) = P(X = i)$. The reward per state r_i is then considered when evaluating the expected reward of the system. This is mainly a system-centric point of view, since the system state and the corresponding reward is considered: $\sum_i x(i) r_i$ is the expected (system-centric) reward. The reader should note that although the rewards may reflect user-centric utilities (e.g. available video bitrate in that system state, and hence video quality), the analysis is nevertheless system-centric, as opposed to user-centric.

From a user-centric point of view, it is important to understand that the QoE of a user is determined for the entirety of the session the user is consuming. A good example of this is over-the-top adaptive video streaming; as network conditions change, the video bitrates are adapted to the networking situation. Naturally, the QoE is impacted by those bitrate (i.e., system) changes. For a user-centric evaluation of systems, we need to be able to quantify therefore the reward reflecting the QoE of the user. From a system-centric perspective, we would consider the probability that the system serves all customers with a certain bitrate. It is more complex to analyze the user-centric reward instead of the system-centric reward.

As a use case in this paper, we consider cloud gaming, concretely, Google Stadia. The cloud servers implement admission-control on the application layer; if the available bandwidth is below a certain threshold ($10\,\mathrm{Mbps}$), then a user is not allowed to enter the system. This is reasonable, as the QoE would be not sufficient for a good user experience. Depending on the available bandwidth, Google Stadia delivers the video contents to the users in higher resolutions and higher video bitrates. Our contribution builds upon a measurement study on Google Stadia and is the user-centric Markov reward model for such a service. We consider the research questions:

- How to analyze user-centric reward models in order to understand system-level performance?
- Is there a significant difference between system-centric and user-centric views in practice for realistic cases?

Section II provides background on Markov reward models, as well as cloud gaming QoE. Section III discusses our measurement study on Google Stadia. Section IV provides the modeling framework for user-centric rewards. Section V presents definitions for the expected system-centric reward. In addition, we show how the accumulated system-centric reward is related to the expected system-centric reward of an individual user. Section VI defines the expected user-centric reward for cloud gaming. The key contribution of this paper is then the derivation of closed formulas for the expected user-centric rewards. We will show the relation between system-centric and user-centric rewards for cloud gaming. Section VII presents numerical results and gives answers to the research questions above. We show the potential of the Markov reward model to investigate different implementations of cloud gaming and to find operational points in practice. Section VIII concludes this work and gives an outlook on relevant future work.

This work was partly funded by the Bavarian Ministry of Economic Affairs, Regional Development and Energy in the project 5GQMON under grant number DIK0169/02.

II. Background and Related Work

A. Background on Reward Models

Markov reward models are commonly used in literature to analyze communication networks and distributed systems. The authors of [1] use Markov reward model to analyze the availability of systems which are modeled as continuous-time Markov chains (CTMC). Each model state of the CTMC corresponds to a system state. Then, the Markov reward models associate a non-negative real-valued reward rate with each state. The stochastic process $\{X(t), t \leq 0\}$ describes the system at time t with a state probability vector $\mathcal{X}(t)$. The corresponding reward rates per state are summarized in the reward vector $\mathcal{R}$. The measures of interest are the expected (instantaneous) reward of the system at time t and especially in the steady-state. Accumulated rewards [2] are also relevant, e.g. to capture availability of systems in terms of the expected accumulated reward for finite intervals of time or the expected time-averaged accumulated reward over an infinite time interval [3]. For accumulated rewards, the time δ_i in a state i with reward r_i is considered and the accumulated reward is then $r_i\delta_i$. With proper definitions for reward rates, the performance, QoS, utility, performability, reliability, etc., of systems can be investigated [3]. For example, Markov reward models are used for the analysis of cloud computing [4], networks-on-chips [5], safety critical systems such as smart grids [6], vehicle-to-infrastructure communications [7]. For the analysis of network survivability [8], reward models are also essential e.g., the survivability of telecommunication network systems under fault propagation [9]. Recently, Markov reward models have also been used with different focus, e.g., to analyze whether small solar panels can drastically reduce the carbon footprint of radio access networks [10]. In this work, we shift the focus towards the QoE of a user of a service.

The Markov reward models above have in common that the *system* is analyzed with different measures and reward rates. This lies in the nature of the Markov model describing the system state and the assignment of rewards to system states. However, we are interested in analyzing a system from a user-centric perspective. In our cloud gaming use case, we consider a system with a shared bottleneck link. Accordingly, the video bitrate received by the player from the cloud rendering the game is a proper reward rate to take into account the user-perspective. Nevertheless, the Markov model still reflects the *system state* and common measures like the expected reward or the expected accumulated reward do not explicitly take into account an individual user. Our goal is quantify the *expected user-centric reward* which takes into account the changing system conditions and the dynamicity of the system while the user consumes the service.

In our previous work [11], we have used a Markov reward model to analyze the QoE of online authentication services considering the impatience of users. The access to online services such as shopping carts, online banking, online authentication, web, etc., is considered. The user requests an online authentication service, and may have to wait until the request is served due to limited resources. However, during waiting, the user may decide for abandonment due to impatience. The QoE of an user is mainly shaped by the waiting time of that user, which is in turn determined by the system state when the user arrives at the system (assuming FIFO scheduling discipline). For the cloud gaming use case, the analysis is much more complex, because the state in which a customers arrives is not sufficient to determine the QoE. We need to take into account the changing system conditions of the system while the user consumes the service due to adaptive video bitrate streaming.

The modeling framework of this paper can be also applied to the QoE analysis of other multimedia and Internet services.

B. Background on Cloud Gaming

Cloud Gaming is intended to remove the need for end-users to have high-powered rendering hardware at their disposal in order to play a video game. Local user input is transmitted to the cloud where the entire game experience is rendered and subsequently delivered back to the user via video streaming. There are several advantages, both practical and economic, to this approach for players and game developers alike.

So why is Cloud Gaming not already the prevalent form of consuming games in 2020? The two main reasons for that can be found in operational costs of a Cloud Gaming platform as well as network quality. When OnLive made the first attempt to commercialize the concept in 2010 the cloud concept in itself was not yet in the mature state it is in today, and the operational costs, along with network issues, made the service non-viable. Fast-forward to today and things have changed considerably. The improvements in hardware and Cloud services, along with the ongoing deployments of better access technologies like 5G networks as well as the recent trend towards edge computing will further serve to mitigate the technological hurdles for Cloud Gaming.

Nonetheless, Cloud Gaming remains one of the most demanding applications for networking today due to its high bandwidth and low latency requirements. This is where mechanisms for QoE optimization come into play and understanding them is key to improve overall network performance and in turn, QoE.

C. Cloud Gaming QoE

For the Markov reward model of online cloud gaming, we want to utilize a QoE model to define proper user-centric rewards. However, there is no commonly accepted QoE model for cloud gaming so far, although several works investigate QoE and QoE influence factors for online cloud games. [12] gives an introduction to online video games including cloud gaming and summarizes QoS and QoE influencing factors. Due to the streaming component, the network bitrate and the resulting video bitrate as well as other video quality metrics are relevant QoE influence factors. [13] reviews QoE in cloud gaming models and also identifies bitrate as a crucial parameter of QoE. [14] investigates the video quality of commercial cloud gaming services of the first wave in 2013. [15] evaluated the impact of delays and packet loss

on the QoE. [16] conducted subjective experiments on cloud gaming and found that cloud requires a certain minimum network bandwidth to maintain an acceptable visual quality, but increasing the bandwidth beyond a certain threshold did not lead to a significant increase in QoE. [17] examined different video qualities and video bitrates and found that their test subjects experienced higher flow and immersion for high video quality. [18] conduct subjective QoE studies and provide QoE estimation models as functions of video bitrate and frame rate depending on the game type and player skill. Similarly, [19] quantified the impact of frame rate and bit rate of cloud gaming QoE. However, no commonly accepted QoE models exists so far. As of this writing, there are various on-going ITU-T standardization activities targeting gaming QoE, e.g. ITU-T Rec. G.1072 (G.OMG) aiming at predicting cloud gaming QoE, which are summarized in [20].

From the literature, we conclude that the video bitrate a user experiences during a cloud gaming session is the most crucial QoE influence factor in this work. Thus, our reward model needs to consider the video bitrate of an individual user.

III. Cloud Gaming Measurements: Google Stadia

Google Stadia uses the WebRTC protocol at the application layer to transmit real-time audio and video from the game rendering server to the user's client. The WebRTC payload is carried by a QUIC connection consistently using UDP port 44700 on the server side during our measurement campaign. A second QUIC connection is established using UDP port 44732, that carries the control inputs from the client to the server. Video encoding is selected based on the capabilities of the client, its screen resolution and quality of the connection between client and server. Stadia has a preference for Google's VP9 codec, if a VP9 decoder is present in hardware at the client, otherwise it defaults to the H.264 codec. The highest supported screen resolution is 3840x2160. A frame rate of 60 frames per second is maintained in all cases to ensure responsiveness.

A. Measurement Setup

The measurements were taken on an off-the shelf PC running Windows 10 with the game client running Google Chrome to access the Stadia service. Wireshark was used to capture the traffic. The PC was connected to a TP-Link Archer 7 AC1750 V2 router running OpenWRT 19.07 that doubled as the uplink router as well as network emulator. The built-in user-space tool tc-netem was used to interact with OpenWRT's Linux kernel packet scheduler and introduce artificial delay and packet loss. A fairly consistent 10ms round trip time (RTT) was observed by constantly pinging the game server during an active session without impairments.

B. Results

A goal of the measurement study was to determine the quality of service limits under which the Stadia service could still function. However, Google actually preempts this determination by performing a connectivity check, i.e., an active measurement between client and server, before a new session is started. If the check fails, the users are denied access to the service. We experimentally determined the minimal quality requirements for the service to work for each dimension individually. In terms of delay, an RTT of $75\,\mathrm{ms}$ at most is tolerated, while a minimum bandwidth of $10\,\mathrm{Mbps}$ is required. Introduced packet loss from client to server was accepted up to a limit of 35% and up to 15% in the opposite direction.

Figure 1 shows the throughput measurements over time of an example $150\,\mathrm{s}$ long run using a 1920x1080 screen without artificial impairments on the connection. The game in this particular measurement was Shadow of the Tomb Raider. The figure illustrates the different phases of a Stadia game session. From $0\,\mathrm{s}$ until about $15\,\mathrm{s}$ the connection test is performed. $10\,\mathrm{Mbps}$ probes are sent from server to client (labeled 'check down'). Only about $0.2\,\mathrm{Mbps}$ are sent in the opposite direction ('check up'). Afterwards what we have dubbed the "interactive phase" starts and runs until the end. Here the two QUIC connections are reflected: WebRTC connection for video and audio in downlink ('stream down') and uplink ('stream up'). The control connection throughput is nearly constant throughout the run duration varying between $1\,\mathrm{kbps}$ and $2\,\mathrm{kbps}$ in both directions. The throughput pattern shown for the WebRTC session is entirely specific to this game session as it depends on the user's input and what is actually shown on the screen as Stadia adapts the video encoding to the content. We observe an initial ramp-up of downlink traffic to just below $30\,\mathrm{Mbps}$ after the connection check is passed. However, it quickly drops again to less than $1\,\mathrm{Mbps}$ at $25\,\mathrm{s}$. This is due to the static content being presented to the user here, i.e., the game developer's and publisher's logos, while the game is loading. We ramp up again to almost $30\,\mathrm{Mbps}$ once the game menu is reached. However, that spike only lasts until an actual game is started from the menu. The throughput is then quickly reduced to just below $1\,\mathrm{Mbps}$ during the loading of the gameplay sesion as once more only a semi-static loading screen is displayed. From the throughput pattern we can derive how long it took the Stadia server to load the game, i.e., about $70\,\mathrm{s}$. What follows is the interactive gameplay phase where the throughput jumps back up to just below $30\,\mathrm{Mbps}$. Gameplay is stopped by the user after $30\,\mathrm{s}$. Once again this is followed by a brief loading screen and corresponding drop below $1\,\mathrm{Mbps}$ throughput as well as subsequent ramp up when entering the menu where the user ends the session at $150\,\mathrm{s}$.

In Figure 2, we now switch to a 4k resolution screen and focus on the throughput of the WebRTC connections exclusively. The game played in this session is GRID. The uplink traffic throughput displays a similar behavior as in the previous session, staying between $0.2\,\mathrm{Mbps}$ and just below $1\,\mathrm{Mbps}$ throughout the session depending on the content transmitted. This is not surprising as the uplink of the WebRTC connection mainly carries acknowledgements. In the downlink connection we observe two main differences. First the throughput pattern is different over time as the user is presented with different content. GRID uses animations and videos rather than static screens during loading phases and this

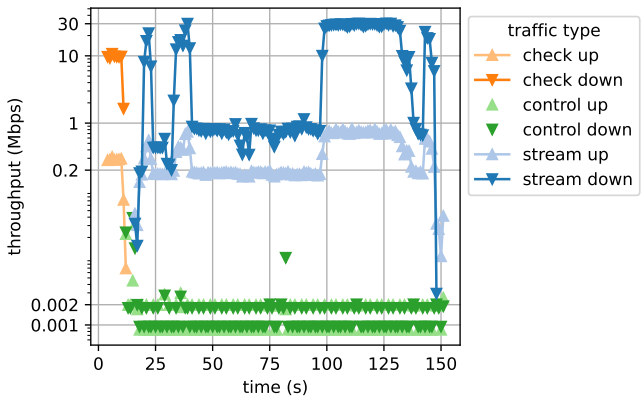

Fig. 1: Example of throughput measurements of Google Stadia without any impairments and a resolution of 1920x1080.

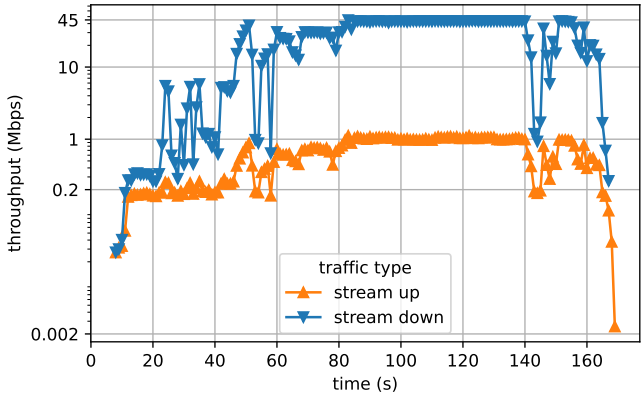

Fig. 2: Example of throughput measurements of Google Stadia without any impairments and a 4k resolution.

is reflected in the throughput. The second difference lies in the sustained throughput during the gameplay in the interactive phase. The 4k video transmission now requires just below 45 Mbps compared to the 30 Mbps of the previous session.

In Table I, we give the measured mean throughput results for multiple runs in four scenarios for the gameplay phase. The first two columns represent the 4k and 1080p session types without connection impairments we have discussed in Figure 2 and 1. The third column represents a scenario with added delay and a 75 ms RTT overall, which was the highest latency tolerated by the connection check. The fourth column represents a scenario with an artificially added asymmetric packet loss of 15% on the downlink. This was also the highest loss ratio tolerated by the connection check. We observe in both cases a significant reduction of the downlink throughput to roughly 10 Mbps. The video resolution is reduced by Stadia to 720p to facilitate this reduction in throughput.

C. Conclusion for the Markov Model

The cloud gaming system can therefore be modeled as a loss system. If the available network capacity of a user is less than 10 Mbps, the user is blocked and not allowed to enter the cloud gaming server. Assuming a bottleneck access link with capacity C, then the users share the capacity and are assigned the corresponding video bitrates (i.e. reward). The corresponding system model is an M/M/n loss system as discussed in Section IV-B.

TABLE I: Measurement Results Summary on Google Stadia.

Scenario	No Imp. 4K	No Imp.	75 ms RTT	15% Loss
Codec	VP9	H.264	H.264	H.264
Resolution	3840x2160	1920x1080	1280x720	1280x720
Ctrl. Up	0.85 kbps	0.85 kbps	0.85 kbps	1.14 kbps
Ctrl. Down	0.94 kbps	0.94 kbps	0.94 kbps	0.94 kbps
Stream	43.05 Mbps	28.6 Mbps	10.23 Mbps	11.36 Mbps

IV. Modeling Framework for User-Centric Rewards of Cloud Gaming

A. Notation and Definitions

We consider a system with shared or limited resources. The system state is reflected by the number i of users in the system, which determines the system behavior. With a finite system capacity, arriving users will be rejected when the capacity is reached and the system blocks the user. The probability that the system is in state i at time t is $x(i,t)$. In the steady state, it is $x(i) = \lim_{t\to\infty} x(i,t)$.

The system is described as a Markov model with transition rates q_{ij} from state i to state j. The transition rate for leaving state i is $q_i = \sum_{i\neq j} q_{ij}$. The rates are summarized in the rate matrix $\mathcal{Q}$ with $q_{ii} = -q_i$. This allows a compact representation of the system transition behavior.

To each state we assign *reward rates*, r_i, which is the individual reward of a user in state i. In our analysis, we consider the video bitrate an individual user is assigned to when the system is in state i. The definition of the rewards allows to take into account a user-centric perspective. In the case of cloud gaming, the video bitrate drives the QoE of a user. The reward rates are summarized in the reward vector $\mathcal{R} = (r_0, r_1, \dots)$.

The expected user-centric reward of a customer arriving in state i and staying in the system for time t is denoted as $\overline{R}_{t|i}$. For the cloud gaming use case, the users stay in the system for time B, which is a random variable. Then, we define $\overline{R}_i$ as the expected user-centric reward of a customer arriving in state i and staying in the system for time B. Finally, the expected user-centric reward of an arbitrary customer is $\overline{R}$.

Additionally, we quantify the expected system-centric reward $\overline{S}$ by considering the steady state probabilities of the system and the reward rate per state. Note that $\overline{S}$ is not taking into account which states an individual user is passing.

B. Modeling Cloud Gaming System as M/M/n Loss System

First, the system model of the cloud gaming system is described as M/M/n loss system. Then, the reward model is provided for two different video bitrate assignment strategies (`FULL`, `SAME`).

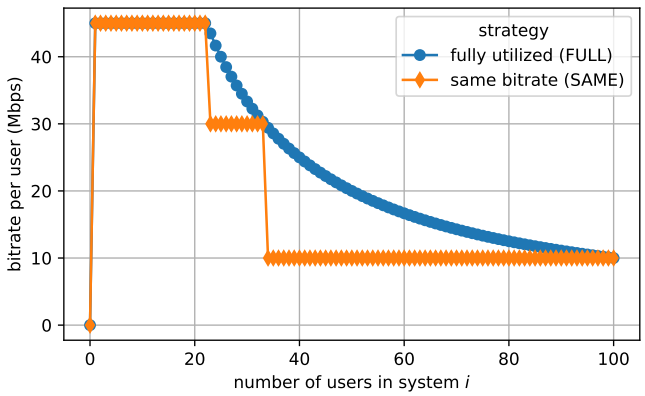

Fig. 3: Assignment of video streaming bitrates depending on the number of users in the system.

1) System Model: Users are arriving according to a Poisson process with rate λ. The mean time of customers in the system is $\mathrm{E}[B] = 1/\mu$ and the service times are exponentially distributed with parameter μ. Then, the offered load is $a = \lambda \cdot \mathrm{E}[B] = \lambda/\mu$. Hence, the state transitions are $q_{i,i+1} = \lambda$ and $q_{i,i-1} = i\mu$, as depicted in Figure 4.

The steady state probabilities $x(i)$ are computed according to the well known Erlang formula for loss systems.

$$x(i) = \frac{\frac{a^i}{i!}}{\sum_{k=0}^{n} \frac{a^k}{k!}} \qquad i = 0, 1, \ldots, n \tag{1}$$

The blocking probability is the Erlang-B formula

$$p_B = x(n) = \frac{\frac{a^n}{n!}}{\sum_{k=0}^{n} \frac{a^k}{k!}} \tag{2}$$

and follows from the PASTA property ("Poisson Arrival Sees Time Average" [21], [22]) and the underlying Poisson arrival process. The steady state probability $x_A(i)$ of the system as seen by an arriving user is identical to the steady state probability $x(i)$ at an arbitrary, random point in time.

2) Reward Model: In state i, there are i users in the system sharing the capacity C of the bottleneck link. For our numerical results, we consider $C = 1\,\mathrm{Gbps}$. Up to 22 users can be served with maximum bitrate of $45\,\mathrm{Mbps}$ for $C = 1\,\mathrm{Gbps}$. At most 100 users can join the system with minimum bitrate of $10\,\mathrm{Mbps}$. Figure 3 illustrates the assigned video streaming bitrate per user depending on the number i of users in the system. We distinguish two different strategies:

- `SAME`: Every user is assigned the same bitrate in the system. If there are e.g. 23 users in the system, then all users get $30\,\mathrm{Mbps}$ and only $690\,\mathrm{Mbps}$ are utilized.
- `FULL`: This strategy assigns some of the users higher bitrates to fully utilize the bottleneck's capacity. We model this strategy as processor-sharing and every user gets then a fraction of time the higher bitrate.

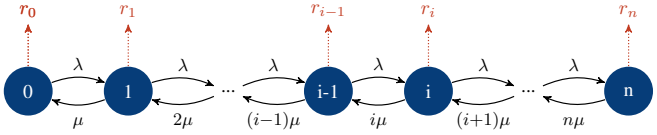

Fig. 4: State transition diagram of an M/M/n-0 loss system with reward rates r_i in state i.

C. Problem Formulation

The problem we are solving is the derivation of the expected user-centric reward $\overline{R}$ of the M/M/n loss system. The user-centric reward quantifies for an arbitrary user entering the system the time-average of the accumulated reward over time. The user enters the system at time t_0 and finds there k other users. Thus, immediately after the arrival of the tagged user, the system is in state $\Gamma(t_0) = k{+}1$. In that state, the user gets a reward $r_{\Gamma(t_0)}$. The user leaves the system at time t_m. However, the system state changes over time at $t_1, t_2, \ldots, t_{m-1}$ and accordingly the reward for that user. The reward of the tagged customer is the time-average of the accumulated reward over time.

$$\overline{r^*} = \frac{1}{t_m - t_0} \sum_{j=0}^{m-1} (t_{j+1} - t_j) \cdot r_{\Gamma(t_j)} \tag{3}$$

The system state $\Gamma(t_j)$ at time t_j is immediately after the event (arrival or departure) and the accumulated reward in that state is $(t_{j+1} - t_j) \cdot r_{\Gamma(t_j)}$. The research question we are answering is then: *What is the expected user-centric reward $\overline{R}$ of the M/M/n loss system for an arbitrary user?*

D. Non-stationary Analysis of Markov Model

For Markov state processes, the Kolmogorov forward equation for transition probabilities $\mathcal{P}(t)$ during the time interval t is provided in matrix notation [23].

$$\frac{d\mathcal{P}(t)}{dt} = \mathcal{P}(t) \cdot \mathcal{Q} \tag{4}$$

The solution of this system of differential equations requires the computation of the matrix exponential of the matrix $t\mathcal{Q}$ for which efficient implementations exist [24], [25].

$$\mathcal{P}(t) = e^{t\mathcal{Q}} = \sum_{k=0}^{\infty} \frac{(t\mathcal{Q}^k)}{k!} \tag{5}$$

This allows to compute the state probabilities at any time t for given initial state $\mathcal{X}(0)$.

$$\mathcal{X}(t) = \mathcal{X}(0) \cdot \mathcal{P}(t) = \mathcal{X}(0) \cdot e^{t\mathcal{Q}} \tag{6}$$

The steady state probabilities are obtained for $t \to \infty$

$$\mathcal{X} = \big(x(0), x(1), \ldots, x(n)\big) = \lim_{t\to\infty} \mathcal{X}(t) \tag{7}$$

by solving

$$\mathcal{X} \cdot Q = \mathbf{0} \text{ with } \sum_{i=0}^{n} x(i) = 1\,. \tag{8}$$

Due to the PASTA property an arriving customer sees the state as a random observer and we obtain the same probabilities $x_A(i) = x(i)$. Hence, we do not need to distinguish the state probabilities as seen by an arriving customer and simply use $x(i)$ for the probability that the system is in state i. The corresponding steady state probabilities are summarized in the vector $\mathcal{X} = \big(x(0), x(1), \ldots, x(n)\big)$.

V. System-Centric Reward

The *expected (instantaneous) system-centric reward* $\overline{S}_0$ (or system-centric reward in short) is defined as

$$\overline{S}_0 = \sum_{i=0}^{n} r_i \cdot x(i) \tag{9}$$

based on the reward rate r_i of the system in state i and the steady state probability $x(i)$ to be in state i.

The definition of the system-centric reward $\overline{S}_0$ evaluates the reward the system gives to its users. In particular, $\overline{S}_0$ also considers the idle system and assigns the reward rate per user in idle state, r_0. In our results, we use $r_0 = 0$ for the idle system to reflect that the system is idle and not serving any customers. Hence, $\overline{S}_0$ mixes the reward of an arbitrary user and the utilization of the system, i.e. not being idle.

We may also define the *accumulated system-centric reward* to quantify the reward of all users from a system-centric perspective. This leads to the following notion taking into account the number of users per state and their reward.

$$\overline{S}_\Sigma = \sum_{i=0}^{n} i \cdot r_i \cdot x(i) \tag{10}$$

However, we are more interested to quantify the expected reward of an arbitrary user in the system. To this end, the expected system-centric reward from the perspective of an arriving user is defined. We consider the evolution over time for an arriving user. The arriving user occupies one of the n places in the system. The remaining $n^* = n-1$ places may be occupied by other users. From the perspective of the arriving users (tagged user), the system behaves like an M/M/n* = M/M/(n-1) loss system during the service time of the tagged user. Then, in state i of the M/M/n* system, there are in total $(i+1)$ users in the system including the tagged user leading to the reward r_{i+1} in state i of the M/M/n* system. The steady state probabilities are denoted as $x^*(i)$ and follow from the Erlang formula in Eq.(1) for $n-1$ available servers, i.e. accepted other customers with $a = \lambda/\mu$.

$$x^*(i) = \frac{\frac{a^i}{i!}}{\sum_{k=0}^{n-1} \frac{a^k}{k!}} \qquad i = 0, 1, \ldots, n-1 \tag{11}$$

Together with the reward r_i^* of the M/M/n* loss system,

$$r_i^* = r_{i+1} \qquad i = 0, 1, \ldots, n-1, \tag{12}$$

we finally arrive at the *expected system-centric reward of an individual (tagged) user*

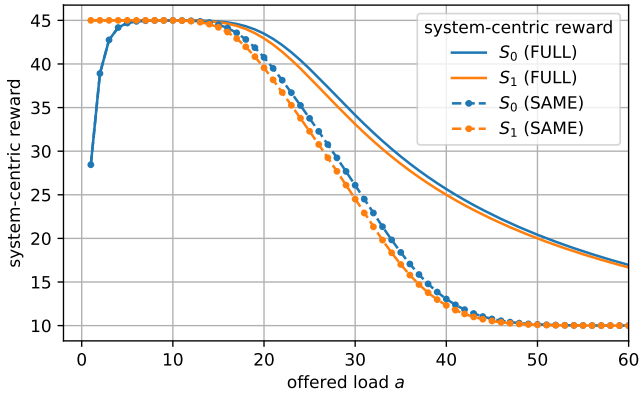

Fig. 5: System-centric reward for the M/M/100 loss system and varying offered load a.

$$\overline{S}_1 = \sum_{i=0}^{n-1} r_i^* \cdot x^*(i)\ . \tag{13}$$

Thereby, we observe the following relationship between the system-centric reward of an individual user $\overline{S}_1$ and the accumulated system-centric reward $\overline{S}_\Sigma$.

$$\overline{S}_1 = \frac{\overline{S}_\Sigma}{\mathrm{E}[\,X\,]} \tag{14}$$

The mean number of users in the system is

$$\mathrm{E}[\,X\,] = \sum_{i=0}^{n} i \cdot x(i)\ . \tag{15}$$

Then, we can algebraically transform the ratio $\frac{\overline{S}_\Sigma}{\mathrm{E}[\,X\,]}$ using $x(i) = \frac{a^i/i!}{\sum_{k=0}^{n} a^k/k!}$ and obtain the relationship.

$$\frac{\overline{S}_\Sigma}{\mathrm{E}[\,X\,]} = \frac{\sum_{i=0}^{n} i \cdot r_i \cdot x(i)}{\sum_{k=0}^{n} k \cdot x(k)} = \frac{\sum_{i=0}^{n} i \cdot r_i \cdot \frac{a^i}{i!}}{\sum_{k=0}^{n} k \cdot \frac{a^k}{k!}} \tag{16a}$$

$$= \frac{a \sum_{j=0}^{n-1} r_{j+1} \cdot \frac{a^j}{j!}}{a \sum_{k=0}^{n-1} \frac{a^k}{k!}} = \sum_{j=0}^{n-1} r_{j+1} \cdot \frac{\frac{a^j}{j!}}{\sum_{k=0}^{n-1} \frac{a^k}{k!}} \tag{16b}$$

$$= \sum_{j=0}^{n-1} r_{j+1} \cdot x^*(j) = \sum_{j=0}^{n-1} r_j^* \cdot x^*(j) = \overline{S}_1 \tag{16c}$$

Figure 5 illustrates the system-centric reward for an M/M/100 loss system with varying load. The reward rates are defined as in Figure 3 based on the video streaming bitrates assigned to users. We see that for high loads, there is no significant difference between $\overline{S}_0$ and $\overline{S}_1$. However, for small loads, an arriving accepted user will get the maximum video bitrate of 45 Mbps as indicated by $\overline{S}_1$. However, the system-centric reward $\overline{S}_0$ will consider the idle system and reward it with $r_0 = 0$ Mbps. The figure also shows that, as expected, the full utilization of the system capacity C (`FULL`) yields a higher expected rewards than the `SAME` video bitrate assignment.

VI. User-centric Reward

A. Reward of User Arriving in State i

The expected (instantaneous) reward of a user arriving in state i and staying in the system for time t is denoted as $\overline{R}_{t|i}$. Thereby, the system behaves like an M/M/n* loss system from the perspective of the arriving user with $n^* = n-1$. Hence, the M/M/n system without the arriving (tagged) user is considered. The steady state probabilities of the M/M/n* loss system are $x^*(i)$ (Eq.(11)) and the reward of the tagged user is r_i^* (Eq.(12)).

We define the conditional state probability vector that in the steady state an arriving customer finds the system in state $i = 0,1,\ldots,n-1$ with the vector $\mathcal{I}_i$. This vector has 1 in position i and 0 otherwise. For $i = 0, 1, \ldots, n-1$, it is

$$\mathcal{I}_i = \big(I_i(0), \cdots, I_i(j), \cdots I_i(n-1)\big) \tag{17}$$

where $I_i(j) = 1$ when $j = i$, and 0 otherwise.

In the steady state, the user arriving in state i at time t_0 stays in the system for time t. Analogously to Eq.(6), the corresponding state probabilities are summarized in the vector

$$\mathcal{X}^*_{t|i} = \mathcal{I}_i \cdot \mathcal{P}^*(t) = \mathcal{I}_i \cdot e^{t\mathcal{Q}^*} \qquad i = 0, 1, \ldots, n-1. \tag{18}$$

with $\mathcal{Q}^*$ and $\mathcal{P}^*(t)$ being the rate matrix and the state transition probability matrix in the interval $(t_0, t_0 + t)$ of the M/M/n* loss system, respectively.

Then, the expected user-centric reward is the (time-averaged) accumulated reward over the interval of length t

$$\overline{R}_{t|i} = \frac{1}{t}\int_{\tau=0}^{t} \mathcal{R}^*\mathcal{X}^*_{\tau|i}\, d\tau = \frac{1}{t}\int_{\tau=0}^{t}\sum_{k=0}^{n-1} r_k^* x^*_{\tau|i}(k)\, d\tau \tag{19}$$

with the conditional probability $x^*_{\tau|i}(k)$ that a user arriving at time t_0 in the M/M/n* system in state i will be in the state k after time τ. Note that i and k may take values $0, 1, \ldots, n-1$. The scalar product $\mathcal{R}^* \cdot \mathcal{X}^*_{\tau|i} = \sum_{k=0}^{n-1} r_k^* \cdot x^*_{\tau|i}(k)$ reflects the instantaneous reward for that user at time τ with reward vector

$$\mathcal{R}^* = \big(r_0^*, r_1^*, \ldots, r_{n-1}^*\big) = \big(r_1, r_2, \ldots, r_n\big)\,. \tag{20}$$

A user stays in the system for a randomly distributed time B with probability density function $b(t)$. Then, the expected user-centric reward for a user arriving in state i is

$$\begin{aligned}\overline{R}_i &= \int_{t=0}^{\infty} \overline{R}_{t|i} \cdot b(t)\, dt \\ &= \int_{t=0}^{\infty} \frac{1}{t}\int_{\tau=0}^{t}\sum_{k=0}^{n-1} r_k^* \cdot x^*_{\tau|i}(k)\, d\tau \cdot b(t)\, dt\end{aligned} \tag{21}$$

B. Expected User-Centric Reward

Finally, an arriving user that is not blocked finds the system in state $i = 0, 1, \ldots, n-1$ with probability $x^*(i)$ of the M/M/n* system without the tagged customer due to the PASTA property.

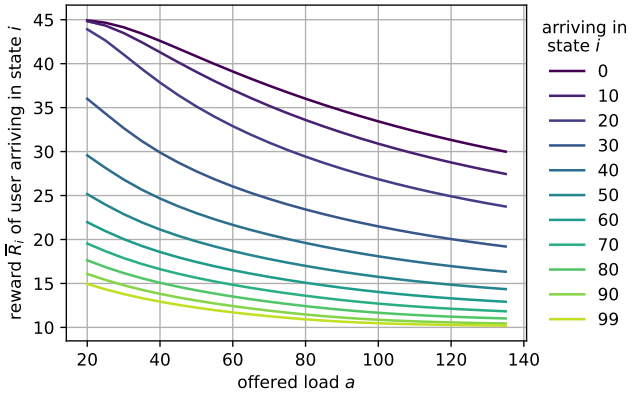

Fig. 6: Expected reward $\overline{R}_i$ of a user arriving in state i for M/M/100 and reward rates corresponding to `FULL` strategy.

$$\overline{R} = \sum_{i=0}^{n-1} x^*(i) \cdot \overline{R}_i \tag{22}$$

The expected user-centric reward $\overline{R}$ is then identical to $\overline{S}_1$, the expected system-centric reward of an individual user.

$$\overline{R} = \overline{S}_1 = \frac{\overline{S}_\Sigma}{\mathrm{E}[\,X\,]} \tag{23}$$

The formal proof of Eq.(23) is as follows.

$$\overline{R} = \sum_{i=0}^{n-1} x^*(i) \cdot \overline{R}_i \tag{24a}$$

$$= \sum_{i=0}^{n-1} x^*(i) \int_{t=0}^{\infty} \frac{1}{t}\int_{\tau=0}^{t}\sum_{k=0}^{n-1} r_k^* x^*_{\tau|i}(k)\, d\tau \cdot b(t)\, dt \tag{24b}$$

$$= \int_{t=0}^{\infty} \frac{1}{t} b(t) \int_{\tau=0}^{t}\sum_{k=0}^{n-1} r_{k+1} \underbrace{\sum_{i=0}^{n-1} x^*(i) x^*_{\tau|i}(k)}_{=x^*_\tau(k)=x^*(k)}\, d\tau\, dt \tag{24c}$$

$$= \sum_{k=0}^{n-1} r_{k+1} \cdot x^*(k) \int_{t=0}^{\infty} \frac{1}{t} b(t) \underbrace{\int_{\tau=0}^{t} 1\, d\tau}_{t}\, dt \tag{24d}$$

$$= \sum_{k=0}^{n-1} r_{k+1} \cdot x^*(k) \underbrace{\int_{t=0}^{\infty} b(t)\, dt}_{1} = \overline{S}_1 \tag{24e}$$

Please note $x^*_\tau(k) = x^*(k)$ in Eq.(24c), since $x^*_\tau(k)$ reflects the probability that the M/M/n* system is in the steady state at time t and at time $t + \tau$ in the state k. However, we are already in the steady state, and the system state probabilities are not changing anymore.

For the M/M/n loss system, we finally arrive at

$$\overline{R} = \sum_{i=0}^{n-1} r_{i+1} \cdot \frac{\frac{a^i}{i!}}{\sum_{k=0}^{n-1}\frac{a^k}{k!}} = \overline{S}_1 \tag{25}$$

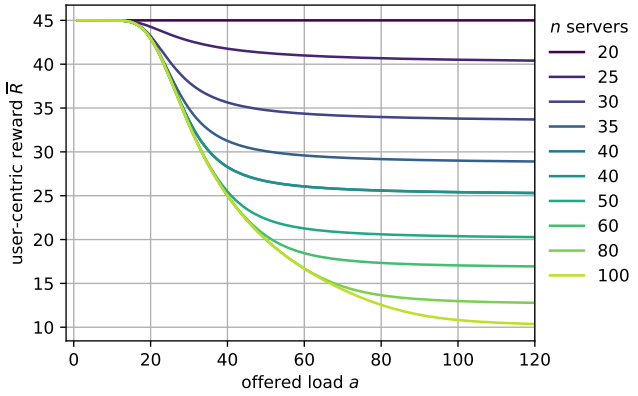

Fig. 7: User-centric reward depending on the offered load and the number n of users who is allowed to access the cloud gaming server at the same time.

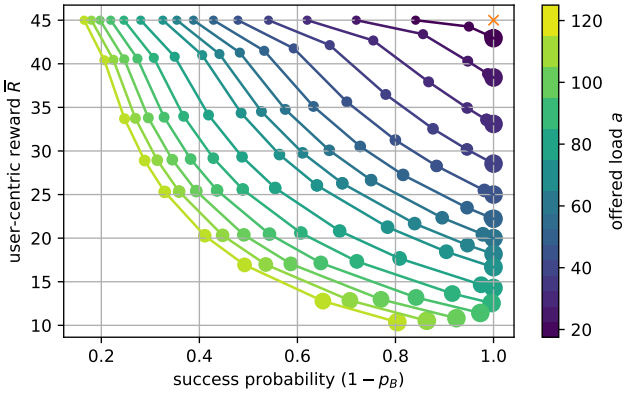

Fig. 8: Pareto front of the user-centric reward $\overline{R}$ vs. the success probability $(1-p_B)$. The marker size indicates the number n of allowed users; the color represents the offered load a.

with an offered load $a = \lambda/\mu$ and the well known Erlang formula in Eq.(1) for the steady state distribution $x^*(i)$ of the M/M/n* loss system.

VII. Numerical Results

A. Scenario Description and Reward Rates

An access link with a bottleneck capacity of $C = 1\,\text{Gbps}$ is considered. The video bitrates of the cloud gaming service are $10\,\text{Mbps}$, $30\,\text{Mbps}$ and $45\,\text{Mbps}$. Users with an access capacity below $10\,\text{Mbps}$ are not allowed to enter the game server. Hence, at most 100 users may access the server at the same time resulting in an M/M/100 loss system. We assume an average play time of one hour and $22\,\text{min}$ according to [26]. The reward rates are defined as in Section IV-B2 and illustrated in Figure 3 for the `FULL` and `SAME` video bitrate assignment strategies.

In the following, we investigate the expected user-centric reward $\overline{R}$. We will also adjust the number n of allowed users at the cloud gaming server, which corresponds to the number of servers in the M/M/n loss system. The idea behind this is that a smaller number n will increase the video bitrates per user and the user-centric reward. However, this will also increase the blocking probability p_B. This trade-off is quantified in Section VII-B and operational points are identified with Kleinrock's power metric in Section VII-C.

B. Quantifying User-centric Reward and Blocking Probability

Figure 7 visualizes the user-centric reward depending on the offered load for various values of n. First, the higher the load, the smaller the expected reward per user. Then, a smaller parameter n increases the reward, since the capacity C is shared among less users. In the extreme case $n = 20$, all users will get the maximum video bitrate of $45\,\text{Mbps}$.

However, a reduced number n of allowed users will increase the blocking probability p_B. Figure 8 illustrates the trade-off between user-centric reward and the success probability $(1-p_B)$. The color of the curve indicates the offered load a, the lighter the color, the higher the load. The size of the markers indicates the number n of allowed users at the gaming server. The cross visualizes the optimal value $\overline{R} = 45\,\text{Mbps}$ and $p_B = 0$.

C. Operational Point: Tradeoff Between Blocking and QoE

For the identification of the operational point for the number n of allowed users, we follow an approach by Kleinrock [27] who suggests the Power metric to identify the knee of the curve. The power metric is the ratio of 'goodness' divided by 'badness'. Then, the optimization of power leads to a trade-off between maximizing 'goodness' while minimizing 'badness'. The maximum value of the power metric is then the operational point. In our case, the user-centric reward $\overline{R}$ is the goodness and the blocking probability p_B is the badness. Then, the power metric is $f(n) = \overline{R}/p_B$ of the M/M/n loss system. The operational point is then the number n^* for which $f(n)$ is maximal: $n^* = \arg\max\,(f(n) : n \in \{1, 2, \ldots, 100\})$.

The result of Kleinrock's approach suggestes that the operational point is $n^* = 100$ if the offered load is below $a < 117$ for both strategies (`FULL`,`SAME`). Hence, blocking probabilities are minimized in that situation. However, for higher load $a > 117$, only $n^* = 22$ users are accepted, i.e., they all get the maximum video bitrate, but the blocking probability is huge.

VIII. Conclusions and Discussions

The key contribution of the paper is the development of a Markov reward model for the use case of cloud gaming which evaluates the video bitrate per user as key QoE influence factor. Experiments on Google Stadia as an example platform for cloud gaming revealed that a minimum network bandwidth of $10\,\text{Mbps}$ is required to get access at the gaming server. In a check up phase before entering the game server, the gaming platform tests the available network bandwidth and only provides access if sufficient network capacity is available. Depending on the available network capacity, the video bitrate

of the game rendered at a cloud server and then streamed to the end user's device is adapated. Our measurements indicated different resolutions leading to video bitrates of $10\,\mathrm{Mbps}$, $30\,\mathrm{Mbps}$ and $45\,\mathrm{Mbps}$. Those video bitrates are then considered in our cloud gaming model and are a crucial QoE influence factor. We assume a bottleneck access link of $1\,\mathrm{Gbps}$ which is equally shared by the users in the system. Then, we can describe the system as an M/M/n loss system with the parameter n indicating the maximum number of allowed servers across that bottleneck link.

For the analysis of the system, we define different system-centric and user-centric rewards. The latter aims at analyzing the QoE of an individual user of the cloud gaming scenario and is another key contribution of this work. In particular, we analyze then the relationships between the system-centric and user-centric rewards. The system-centric reward quantifies the reward from the perspective of the entire system. However, we show that the accumulated system-centric reward normalized by the mean number of users in the system is identical to the user-centric reward of an individual user. This is a strong result and leads to a much simpler computation of the user-centric reward for the M/M/n loss system. In the end, the computation requires only the well-known steady state probabilities of the M/M/n loss system and the reward function.

Finally, we provide some numerical results to quantify the trade-off between blocking probability and the expected user-centric reward. Using Kleinrock's power metric, we identify operational points of the system and how many users should be accepted by the system under different load conditions and video bitrate assignment strategies.

To the best of our knowledge, our works are the first that define user-centric rewards to apply Markov reward models for the analysis of QoE or QoE indicators. In particular, this is the very first Markov reward model for online cloud gaming from the user's perspective. The modeling framework of this paper can be also applied to the QoE analysis of other multimedia and Internet services. In future work, we will analyze the user-centric rewards for arbitrary continuous-time Markov chains, e.g. with state-dependent arrival rates without PASTA property, and their relation to system-centric rewards. Available QoE models for video streaming will be utilized to quantify QoE for such services based on the definition of the user-centric rewards and proper Markov reward models.

REFERENCES

[1] A. Reibman, R. Smith, and K. Trivedi, "Markov and markov reward model transient analysis: An overview of numerical approaches," *European Journal of Operational Research*, vol. 40, no. 2, 1989.

[2] K. S. Trivedi, *Probability & statistics with reliability, queuing and computer science applications*. John Wiley & Sons, 2008.

[3] K. S. Trivedi, M. Malhotra, and R. M. Fricks, "Markov reward approach to performability and reliability analysis," in *Proceedings of International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, IEEE, 1994.

[4] Y. Kirsal, Y. K. Ever, L. Mostarda, and O. Gemikonakli, "Analytical modelling and performability analysis for cloud computing using queuing system," in *2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC)*, IEEE, 2015.

[5] J. Hou and M. Radetzki, "Performability analysis of mesh-based nocs using markov reward model," in *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, IEEE, 2018.

[6] S. Ahamad *et al.*, "Some studies on performability analysis of safety critical systems," *Computer Science Review*, vol. 39, 2021.

[7] L. Jin, G. Zhang, H. Zhu, and W. Duan, "Sdn-based survivability analysis for v2i communications," *Sensors*, vol. 20, no. 17, 2020.

[8] P. E. Heegaard and K. S. Trivedi, "Network survivability modeling," *Computer Networks*, vol. 53, no. 8, 2009.

[9] L. Xie, P. E. Heegaard, and Y. Jiang, "Modeling and quantifying the survivability of telecommunication network systems under fault propagation," in *Meeting of the European Network of Universities and Companies in Information and Communication Engineering*, Springer, 2013.

[10] A. P. C. da Silva, D. Renga, M. Meo, and M. A. Marsan, "Small solar panels can drastically reduce the carbon footprint of radio access networks," in *31th International Teletraffic Congress (ITC 31)*, Budapest, Hungary, 2019.

[11] T. Hoßfeld, L. Atzori, P. E. Heegaard, L. Skorin-Kapov, and M. Varela, "The interplay between qoe, user behavior and system blocking in qoe management," in *2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, IEEE, 2019.

[12] F. Metzger, S. Geißler, A. Grigorjew, *et al.*, "An introduction to online video game qos and qoe influencing factors," *IEEE Communications Surveys & Tutorials*, 2022, "DOI: 10.1109/COMST.2022.3177251".

[13] A. A. Laghari, H. He, K. A. Memon, R. A. Laghari, I. A. Halepoto, and A. Khan, "Quality of experience (qoe) in cloud gaming models: A review," *multiagent and grid systems*, vol. 15, no. 3, 2019.

[14] K.-T. Chen, Y.-C. Chang, H.-J. Hsu, D.-Y. Chen, C.-Y. Huang, and C.-H. Hsu, "On the quality of service of cloud gaming systems," *IEEE Transactions on Multimedia*, vol. 16, no. 2, 2013.

[15] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hoßfeld, "Gaming in the clouds: Qoe and the users' perspective," *Mathematical and Computer Modelling*, vol. 57, no. 11-12, 2013.

[16] A. Sackl, R. Schatz, T. Hossfeld, F. Metzger, D. Lister, and R. Irmer, "Qoe management made uneasy: The case of cloud gaming," in *2016 IEEE International Conference on Communications Workshops (ICC)*, IEEE, 2016.

[17] J. Beyer, R. Varbelow, J.-N. Antons, and S. Möller, "Using electroencephalography and subjective self-assessment to measure the influence of quality variations in cloud gaming," in *2015 Seventh International Workshop on Quality of Multimedia Experience (QoMEX)*, IEEE, 2015.

[18] I. Slivar, L. Skorin-Kapov, and M. Suznjevic, "Cloud gaming qoe models for deriving video encoding adaptation strategies," in *Proceedings of the 7th international conference on multimedia systems*, 2016.

[19] S. Zadtootaghaj, S. Schmidt, and S. Möller, "Modeling gaming qoe: Towards the impact of frame rate and bit rate on cloud gaming," in *2018 Tenth international conference on quality of multimedia experience (QoMEX)*, IEEE, 2018.

[20] S. Schmidt, S. Zadtootaghaj, and S. Möller, "Qitu-t standardization activities targeting gaming quality of experience," *ACM SIGMultimedia Records*, 2021.

[21] R. W. Wolff, "Poisson arrivals see time averages," *Operations research*, vol. 30, no. 2, 1982.

[22] B. Melamed and W. Whitt, "On arrivals that see time averages," *Operations Research*, vol. 38, no. 1, 1990.

[23] P. Tran-Gia and T. Hoßfeld, *Performance Modeling and Analysis of Communication Networks, A Lecture Note*. Würzburg University Press, 2021, ISBN: 978-3-95826-153-2. DOI: 10.25972/WUP-978-3-95826-153-2.

[24] J. Sastre, J. Ibáñez, and E. Defez, "Boosting the computation of the matrix exponential," *Applied Mathematics and Computation*, vol. 340, 2019.

[25] M. Fasi and N. J. Higham, "An arbitrary precision scaling and squaring algorithm for the matrix exponential," *SIAM Journal on Matrix Analysis and Applications*, vol. 40, no. 4, 2019.

[26] Limelight Networks, "Market research: The state of online gaming – 2019," Tech. Rep., 2019.

[27] L. Kleinrock, "Internet congestion control using the power metric: Keep the pipe just full, but no fuller," *Ad Hoc Networks*, vol. 80, 2018, ISSN: 1570-8705. DOI: https://doi.org/10.1016/j.adhoc.2018.05.015.