Performance Evaluation 165 (2024) 102425

Contents lists available at ScienceDirect

Performance Evaluation

journal homepage: www.elsevier.com/locate/peva

User-centric Markov reward model for state-dependent Erlang loss systems

Tobias Hoßfeld [a,*], Poul E. Heegaard [b], Martín Varela [c], Michael Jarschel [d]

[a] University of Wurzburg, Chair of Communication Networks, Würzburg, Germany
[b] NTNU - Norwegian University of Science and Technology, Trondheim, Norway
[c] Universidad de Málaga, Spain
[d] Technische Hochschule Ingolstadt, Faculty of Computer Science, Ingolstadt, Germany

ARTICLE INFO

Keywords:
Markov reward model
Quality of Experience (QoE)
User-centric reward
Cloud gaming
User engagement

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 dynamics while the user consumes a service. The key contributions of this paper are proper definitions for (i) system-centric reward and (ii) user-centric reward of the Erlang loss model M/M/n-0 and M/M(x)/n with state-dependent service rates, as well as (iii) the analysis of the relationships between those metrics. Our key result allows a simple computation of the user-centric rewards. The differences between the system-centric and the user-centric rewards are demonstrated for a real-world cloud gaming use case. To the best of our knowledge, this is the first analysis showing the relationship between user-centric rewards and system-centric rewards. 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 while also considering user engagement.

1. 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 perspective, since the system state and the corresponding reward is considered: $\sum_i x(i)r_i$ is the expected (system-centric) reward. This is illustrated in Fig. 1. The state probabilities and the reward per state define the system-centric reward.

However, this system-centric perspective does not consider which states individual users observe. For example, a user may arrive in an empty system or in a crowded system. For that individual user, the future evolution therefore depends on the initial state, see Fig. 1. From a user-centric perspective, 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, since the individual user sessions need to be considered.

* Corresponding author.
E-mail addresses: tobias.hossfeld@uni-wuerzburg.de (T. Hoßfeld), poul.heegaard@ntnu.no (P.E. Heegaard), martin@varela.fi (M. Varela), Michael.Jarschel@thi.de (M. Jarschel).

https://doi.org/10.1016/j.peva.2024.102425
Received 25 May 2023; Received in revised form 11 April 2024; Accepted 8 June 2024
Available online 10 June 2024

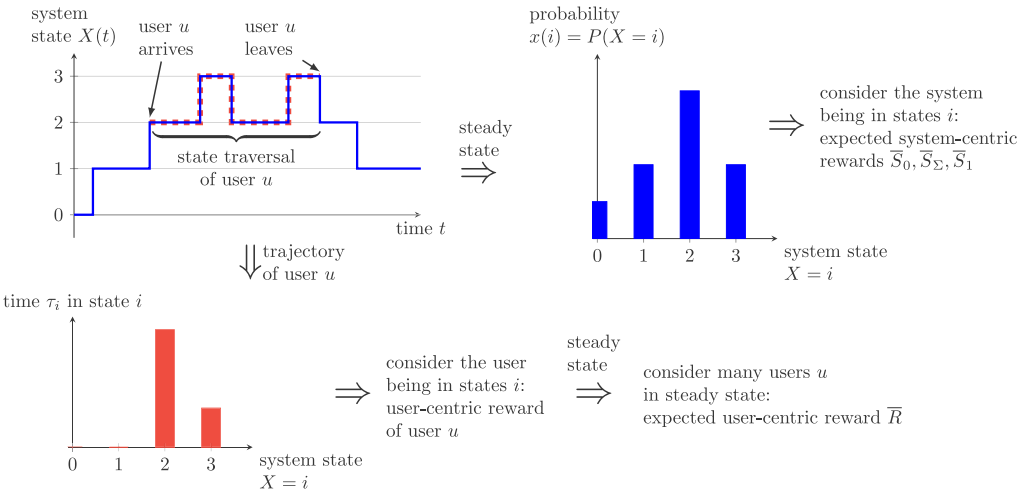


Fig. 1. Illustration of system-centric and user-centric rewards.

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), still just considering the state probabilities instead of the user session provides a system-centric view, as opposed to user-centric. Fig. 1 illustrates the difference between system-centric and the user-centric rewards.

The key contributions of this article are as follows.

- We differentiate and provide definitions of system-centric and user-centric rewards. The reward of a user is formulated as a novel user-centric reward, which allows integrating QoE of a user over the entire session of using as an appropriate reward.
- We are the first to analyze of relationships of system-centric and user-centric rewards for the Erlang loss system M/M/n-0 and the state-dependent Erlang loss system M/M(x)/n-0.
- We show that the user-centric reward ($\overline{R}$) corresponds to appropriate system-centric rewards ($\overline{S}_1, \overline{S}_\Sigma$), but differs to commonly used system reward ($\overline{S}_0$).
- Numerical results are provided for a real use case of cloud gaming, for which the main QoE influence factor is defined as reward, while realistic user behavior and user engagement is integrated in the Markov reward model. The cloud gaming use case is a blueprint for the user-centric analysis of other services beyond cloud gaming, considering QoE as well as user engagement.

In other words, we are answering the following research questions:

- How to analyze user-centric reward models to understand system-level performance?
- What is the user-centric reward for an M/M/n-0 loss system and an M/M(x)/n-0 loss system with state-dependent service rates?
- Is there a significant difference between system-centric and user-centric views in practice for realistic cases?

The remainder of this article is structured as follows. Section 2 provides a background on reward models and introduces the definition of a Markov reward model. The notation is introduced for the non-stationary analysis of Markov models, as well as the notion of rewards. The model of the cloud gaming example is briefly summarized to illustrate the application of the Markov reward model. Section 3 presents definitions for the expected system-centric reward. In addition, we show how the accumulated system-centric reward relates to the expected system-centric reward of an individual user. Note that still the perspective of the system is considered, but not how individual users experience the system. Section 4 defines the expected user-centric reward for Erlang loss systems. The key contribution of this paper is then the derivation of closed formulas for the expected user-centric rewards. Those rewards are derived for the M/M/n-0 Erlang loss system and the M/M(x)/n-0 Erlang loss system with state-dependent service rates. We will show the relation between system-centric and user-centric rewards for both models and analyze whether they are different.

As a use case in this paper, we consider cloud gaming, concretely, Google Stadia (we note that since this work was started, Google has proceeded to shut down Stadia, but it remains a state-of-the-art implementation of Cloud Gaming). The cloud servers implemented admission-control on the application layer; if the available bandwidth is below a certain threshold (10 Mbps), then a user is not allowed to enter the system. This is reasonable, as the QoE with lower bandwidth would be inadequate. Depending on the available bandwidth, Google Stadia delivers the video contents to the users in higher resolutions and higher video bitrates. The rewards reflect the video bitrate as a key QoE indicator. Section 6 provides numerical results showing the difference between

system-centric and user-centric perspective on the cloud gaming use case. Finally, Section 7 concludes this work and gives an outlook on relevant future work.

Note that this is an extended version of the paper [1] by T. Hoßfeld, P. E. Heegaard, M. Varela, M. Jarschel: "User-centric Markov Reward Model on the Example of Cloud Gaming" in *34th International Teletraffic Congress (ITC 34)*, 2022. In [1], we have analyzed the system-centric and user-centric rewards of the Erlang-loss model. Here, we extend [1] by considering state-dependent service rates and therefore provide a novel analysis on the system-centric and user-centric rewards of a state-dependent Erlang loss system M/M(x)/n-0 in Section 3.3 and Section 5, respectively. In addition, the impact of user engagement according to the QoE or the number of other players in the game is modeled and numerically investigated in Section 6.5. Recent related work on traffic characteristics (Section 6.2) and QoE of cloud gaming are revisited (Section 6.3).

2. Modeling framework: Markov reward model

2.1. Background on reward models

Markov reward models are commonly used in the literature to analyze communication networks and distributed systems. The authors of [2] 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 a 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 a time t and especially in the steady-state. Accumulated rewards [3] 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 [4]. 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 [4]. For example, Markov reward models are used for the analysis of cloud computing [5], networks-on-chips [6], safety critical systems such as smart grids [7], vehicle-to-infrastructure communications [8]. For the analysis of network survivability [9], reward models are also essential, *e.g.*, the survivability of telecommunication network systems under fault propagation [10]. 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 [11]. 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 to 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 [12], 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 a 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 customer 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 also be applied to the QoE analysis of other multimedia and Internet services.

2.2. Definition of Markov reward model

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 a 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 transition rates are summarized in the rate matrix 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 the state i. The reward rates are summarized in the reward vector $\mathcal{R} = (r_0, r_1, \ldots)$. A *Markov reward model* is therefore defined by $(Q, \mathcal{R})$.

2.3. 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 [13].

$$\frac{d\mathcal{P}(t)}{dt} = \mathcal{P}(t) \cdot Q \tag{1}$$

The solution of this system of differential equations requires the computation of the matrix exponential of the matrix tQ for which efficient implementations exist [14,15].

$$\mathcal{P}(t) = e^{tQ} = \sum_{k=0}^{\infty} \frac{(tQ)^k}{k!} \tag{2}$$

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

$$\mathcal{X}(t) = \mathcal{X}(0) \cdot \mathcal{P}(t) = \mathcal{X}(0) \cdot e^{tQ} \tag{3}$$

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

$$\mathcal{X} = \big(x(0), x(1), \dots, x(n)\big) = \lim_{t\to\infty} \mathcal{X}(t) \tag{4}$$

by solving

$$\mathcal{X} \cdot Q = \mathbf{0} \text{ with } \sum_{i=0}^{n} x(i) = 1\ . \tag{5}$$

2.4. Notion of rewards

The expected user-centric reward of a customer arriving in state i and staying in the system for time t is denoted as $\overline{R}_{t|i}$. 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_i. For the cloud gaming use case, the users stay in the system for time B (random variable) for any state i. Considering engagement, state-dependent times B_i may be required. Finally, the expected user-centric reward of an arbitrary customer is $\overline{R}$. Additionally, we quantify the expected system-centric reward $\overline{S}_0$ by considering the steady state probabilities of the system and the reward rate per state. The expected accumulated system-centric reward $\overline{S}_\Sigma$ is the sum of the user rewards over all users in the system. Taking the perspective of an individual (tagged) user, we also define the expected system-centric reward $\overline{S}_1$ of that individual (tagged) user. To this end, we consider the system from the tagged user's perspective, how the system evolves and which reward is received by the tagged user. The notatation of the variables used in this article is summarized in Table 2.

3. System-centric reward of Erlang loss systems

We develop appropriate definitions of system-centric rewards, which are based on the steady-state probabilities $x(i)$ that the system is in state i, *i.e.*, the probability that i users are in the system. With different rewards, we obtain the expected system-centric reward $\overline{S}_0$, the accumulated system-centric reward $\overline{S}_\Sigma$, and the expected system-centric reward of a tagged customer $\overline{S}_1$.

3.1. Definition of system-centric reward

Definition 1 (*System-Centric Reward* $\overline{S}_0$)**.** The *expected (instantaneous) system-centric reward* $\overline{S}_0$ (or system-centric reward in short) is defined as

$$\overline{S}_0 = \sum_{i=0}^{n} r_i \cdot x(i) \tag{6}$$

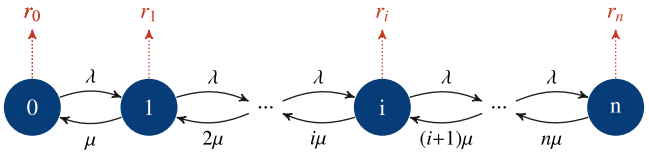
based on the reward rate r_i of the system in state i and the steady state probability $x(i)$ that the system is in state i.

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.

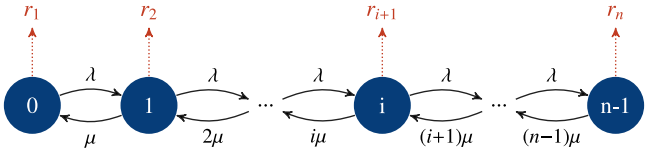
We may also define the *accumulated system-centric reward* to quantify the reward of all users from a system-centric perspective. This leads to the following notion, taking into account the number of users per state and their reward.

Definition 2 (*Accumulated System-Centric Reward* $\overline{S}_\Sigma$)**.** The *accumulated system-centric reward* aggregates the reward of all users and is defined as follows.

$$\overline{S}_\Sigma = \sum_{i=0}^{n} i \cdot r_i \cdot x(i) \tag{7}$$



(a) System perspective M/M/n-0.



(b) Perspective of a tagged user.

Fig. 2. State transition diagram of an M/M/n-0 loss system with reward rates r_i in state i.

However, we are more interested in quantifying 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. The corresponding steady state probabilities of the system from the perspective of a tagged customer are denoted as $x^*(i)$. Together with the reward r_i^* of the system from the perspective of the tagged customer,

$$r_i^* = r_{i+1} \qquad i = 0, 1, \dots, n-1, \tag{8}$$

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

Definition 3 (*System-Centric Reward $\overline{S}_1$ From a Tagged User's Perspective*)**.** The expected system-centric reward of an individual (tagged) user considers the system evolution from a tagged users's perspective with steady state probabilities $x^*(i)$, but appropriate rewards r_i^* for the tagged user.

$$\overline{S}_1 = \sum_{i=0}^{n-1} r_i^* \cdot x^*(i). \tag{9}$$

3.2. System-centric rewards of the M/M/n loss system

Users arrive 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 Fig. 2(a). The steady state probabilities $x(i)$ are computed according to the well-known Erlang formula for loss systems.

$$x(i) = \frac{\frac{a^i}{i!}}{\sum_{k=0}^{n} \frac{a^k}{k!}} \qquad i = 0, 1, \dots, n \tag{10}$$

The blocking probability is the Erlang-B formula

$$p_B = x(n) = \frac{\frac{a^n}{n!}}{\sum_{k=0}^{n} \frac{a^k}{k!}} \tag{11}$$

and follows from the PASTA property ("Poisson Arrival Sees Time Average" [16,17]) 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.

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 the 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, see Fig. 2(b). The steady state probabilities are denoted as $x^*(i)$ and follow from the Erlang formula in Eq. (10) for $n-1$ available servers, *i.e.*, accepted other customers with $a = \lambda/\mu$. Note that the offered load may be larger than $a > 1$, since we have a loss system.

$$x^*(i) = \frac{\frac{a^i}{i!}}{\sum_{k=0}^{n-1} \frac{a^k}{k!}} \qquad i = 0, 1, \dots, n-1 \tag{12}$$

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

Theorem 1 (*System-Centric Reward For M/M/n-0*). *For an M/M/n loss system, the system-centric reward of an individual user* $\overline{S}_1$ *and the accumulated system-centric reward* $\overline{S}_\Sigma$ *averaged over the number of users* $\mathrm{E}[X]$ *in the system are identical.*

$$\overline{S}_1 = \frac{\overline{S}_\Sigma}{\mathrm{E}[X]} \qquad M/M/n\text{-}0 \tag{13}$$

The mean number of users in the system is

$$\mathrm{E}[X] = \sum_{i=0}^{n} i \cdot x(i)\,. \tag{14}$$

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

Proof.

$$\begin{aligned}\frac{\overline{S}_\Sigma}{\mathrm{E}[X]} &= \frac{\sum_{i=0}^{n} i \cdot r_i \cdot x(i)}{\sum_{k=0}^{n} k \cdot x(k)} = \frac{\sum_{i=0}^{n} i \cdot r_i \cdot \frac{a^i/i!}{z}}{\sum_{k=0}^{n} k \cdot \frac{a^k/k!}{z}} = \frac{\sum_{i=1}^{n} r_i \cdot \frac{a^i}{(i-1)!}}{\sum_{k=1}^{n} \frac{a^k}{(k-1)!}} = \frac{\sum_{j=0}^{n-1} r_{j+1} \cdot \frac{a^{j+1}}{j!}}{\sum_{k=0}^{n-1} \frac{a^{k+1}}{k!}} \\ &= \sum_{j=0}^{n-1} r_{j+1} \cdot \frac{\frac{a^j}{j!}}{\sum_{k=0}^{n-1} \frac{a^k}{k!}} = \sum_{j=0}^{n-1} r_{j+1} \cdot x^*(j) = \sum_{j=0}^{n-1} r_j^* \cdot x^*(j) = \overline{S}_1\end{aligned} \tag{15}$$

Fig. 6 shows the difference between $\overline{S}_1$ and $\overline{S}_0$ for the cloud gaming use case in Section 6.4. Obviously, $\overline{S}_1 \neq \overline{S}_0$.

3.3. System-centric rewards of the M/M(x)/n loss system with state-dependent service rates

The Erlang loss system with state-dependent service rates μ_i for states $i = 1, \ldots, n$ is depicted in Fig. 3(a) and denoted as M/M(x)/n-0. The service rate per user (corresponding to the mean state-dependent service time, *i.e.*, mean play time of a user in cloud gaming) is denoted by μ_i for state i. Again, we observe the same relationship of system-centric rewards as for the M/M/n loss system, which we prove below.

Theorem 2 (*System-Centric Reward For M/M(x)/n-0*). *For an M/M(x)/n loss system, the system-centric reward of an individual user* $\overline{S}_1$ *and the accumulated system-centric reward* $\overline{S}_\Sigma$ *averaged over the number of users* $\mathrm{E}[X]$ *in the system are identical.*

$$\overline{S}_1 = \frac{\overline{S}_\Sigma}{\mathrm{E}[X]} \qquad M/M(x)/n\text{-}0 \tag{16}$$

The analysis of the M/M(x)/n-0 system is provided in the literature, *e.g.*, [18]. The steady state probabilities also follow from Eq. (5). However, we can provide the steady state probabilities $x(i)$ with a closed-form equation. Since we have a birth-and-death process, the steady state probabilities are as follows, see, *e.g.*, [13].

$$x(i) = x(0) \cdot \frac{\lambda^i}{i! \prod_{k=1}^{i} \mu_k}\,, \quad i = 1, 2, \ldots, n, \tag{17a}$$

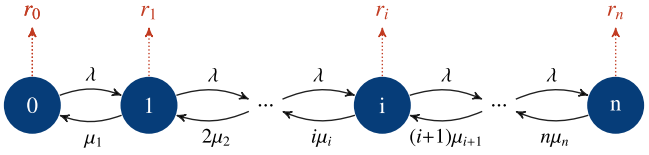
$$x(0) = \left(1 + \sum_{i=1}^{n} \frac{\lambda^i}{i! \prod_{k=1}^{i} \mu_k}\right)^{-1} \tag{17b}$$

From the perspective of a tagged user, we observe a system with $n-1$ other users. The service rates and the reward rates are depicted in Fig. 3(b). The steady state probabilities $x^*(i)$ of the corresponding M/M(x)/(n-1) loss system from the perspective of the tagged user are

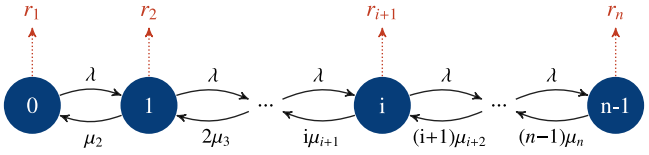
$$x^*(i) = x^*(0) \cdot \frac{\lambda^i}{i! \prod_{k=1}^{i} \mu_{k+1}}\,, \quad i = 1, 2, \ldots, n-1, \tag{18a}$$

$$x^*(0) = \left(1 + \sum_{i=1}^{n-1} \frac{\lambda^i}{i! \prod_{k=1}^{i} \mu_{k+1}}\right)^{-1}\,. \tag{18b}$$

Now we consider the accumulated system-centric reward $\overline{S}_\Sigma$ normalized by the mean number $\mathrm{E}[X]$ of users in the system. This will be equivalent to the system-centric reward $\overline{S}_1 = \sum_{i=0}^{n-1} r_i^* \cdot x^*(i)$ of a tagged user.



(a) System perspective M/M(x)/n-0.



(b) Perspective of a tagged user.

Fig. 3. State transition diagram of an M/M(x)/n-0 loss system with reward rates r_i and state-dependent service rates μ_i in state i.

Proof.

$$\frac{\overline{S}_\Sigma}{\mathrm{E}[X]} = \frac{\sum_{i=0}^{n} i \cdot r_i \cdot x(i)}{\sum_{i=0}^{n} i \cdot x(i)} = \frac{\sum_{i=0}^{n} i \cdot r_i \cdot x(0) \cdot \frac{\lambda^i}{i! \prod_{k=1}^{i} \mu_k}}{\sum_{i=0}^{n} i \cdot x(0) \cdot \frac{\lambda^i}{i! \prod_{k=1}^{i} \mu_k}} = \frac{\sum_{i=0}^{n-1} r_{i+1} \cdot \frac{\lambda^i}{i! \prod_{k=1}^{i} \mu_{k+1}}}{\sum_{i=0}^{n-1} \frac{\lambda^i}{i! \prod_{k=1}^{i} \mu_{k+1}}} = \frac{\sum_{i=0}^{n-1} r_{i+1} \cdot \frac{\lambda^i}{i! \prod_{k=1}^{i} \mu_{k+1}}}{x^*(0)^{-1}}$$

$$= \sum_{i=0}^{n-1} r_{i+1} \cdot x^*(0) \cdot \frac{\lambda^i}{i! \prod_{k=1}^{i} \mu_{k+1}} = \sum_{i=0}^{n-1} r_{i+1} \cdot x^*(i) = \sum_{i=0}^{n-1} r_i^* \cdot x^*(i) = \overline{S}_1 \tag{19}$$

4. User-centric reward for M/M/n-0 loss system

4.1. Reward of user arriving in state i

The expected (instantaneous) reward of a user arriving in a 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. (12)) and the reward of the tagged user is r_i^* (Eq. (8)).

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

$$\mathcal{I}_i = \big(I_i(0), \ldots, I_i(j), \ldots, I_i(n-1)\big) \tag{20}$$

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

In the steady state, the user arriving in state i at a time t_0 stays in the system for time t. Analogously to Eq. (3), the corresponding state probabilities are summarized in the vector

$$\mathcal{X}^*_{t|i} = \mathcal{I}_i \cdot \mathcal{P}^*(t) = \mathcal{I}_i \cdot e^{tQ^*} \qquad i = 0, 1, \ldots, n-1. \tag{21}$$

with Q^* and $\mathcal{P}^*(t)$ being the transition 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

$$\overline{R}_{t|i} = \frac{1}{t} \int_{\tau=0}^{t} \mathcal{R}^* \mathcal{X}^*_{\tau|i} \, d\tau = \frac{1}{t} \int_{\tau=0}^{t} \sum_{k=0}^{n-1} r_k^* x^*_{\tau|i}(k) \, d\tau \tag{22}$$

with the conditional probability $x^*_{\tau|i}(k)$ that a user arriving at a 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 a time τ with reward vector

$$\mathcal{R}^* = \big(r_0^*, r_1^*, \ldots, r_{n-1}^*\big) = \big(r_1, r_2, \ldots, r_n\big) \, . \tag{23}$$

A user stays in the system for a randomly distributed time B with a probability density function $b(t)$.

Theorem 3 (*User-Centric Reward $\overline{R}_i$ Of a User Arriving in State i for M/M/n-0*)**.** *The expected user-centric reward for a user arriving in state i is as follows.*

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

4.2. 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.

Definition 4 (*User-Centric Reward* $\overline{R}$)**.** The expected user-centric reward $\overline{R}$ of an arriving user that is not blocked is defined by the probability $x^*(i)$ that the user arrives in state i and the expected user-centric reward $\overline{R}_i$ when arriving in state i.

$$\overline{R} = \sum_{i=0}^{n-1} x^*(i) \cdot \overline{R}_i \tag{25}$$

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

Theorem 4 (*User-Centric and System-Centric Reward For M/M/n-0*)**.** *For an M/M/n loss system, the expected user-centric reward* $\overline{R}$ *is identical to the system-centric reward* $\overline{S}_1$ *from the perspective of the tagged user.*

$$\overline{R} = \overline{S}_1 = \frac{\overline{S}_\Sigma}{\mathrm{E}[X]} \qquad M/M/n\text{-}0 \tag{26}$$

The formal proof of Eq. (26) is as follows.

$$\overline{R} = \sum_{i=0}^{n-1} x^*(i) \cdot \overline{R}_i = \sum_{i=0}^{n-1} x^*(i) \int_{t=0}^{\infty} \frac{1}{t} \int_{\tau=0}^{t} \sum_{k=0}^{n-1} r_k^* x_{\tau|i}^*(k)\, d\tau \cdot b(t)\, dt \tag{27a}$$

$$= \int_{t=0}^{\infty} \frac{1}{t} b(t) \int_{\tau=0}^{t} \sum_{k=0}^{n-1} r_{k+1} \underbrace{\sum_{i=0}^{n-1} x^*(i) x_{\tau|i}^*(k)}_{=x_\tau^*(k)=x^*(k)}\, d\tau\, dt \tag{27b}$$

$$= \sum_{k=0}^{n-1} r_{k+1} \cdot x^*(k) \int_{t=0}^{\infty} \frac{1}{t} b(t) \underbrace{\int_{\tau=0}^{t} 1\, d\tau}_{t}\, dt = \sum_{k=0}^{n-1} r_{k+1} \cdot x^*(k) \underbrace{\int_{t=0}^{\infty} b(t)\, dt}_{1} = \overline{S}_1 \tag{27c}$$

Please note $x_\tau^*(k) = x^*(k)$ in Eq. (27b), since $x_\tau^*(k)$ reflects the probability that the M/M/n* system is in the steady state at the time t and at the 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

$$\overline{R} = \sum_{i=0}^{n-1} r_{i+1} \cdot \frac{\frac{a^i}{i!}}{\sum_{k=0}^{n-1} \frac{a^k}{k!}} = \overline{S}_1 \tag{28}$$

with an offered load $a = \lambda/\mu$ and the well-known Erlang formula in Eq. (10) for the steady state distribution $x^*(i)$ of the M/M/n* loss system.

5. User-centric reward: M/M(x)/n loss system with state-dependent service rates

Now, we consider state-dependent service rates μ_i for $i = 1, \ldots, n$. For deriving the expected user-centric reward $\overline{R}$, we follow a similar approach as for the Erlang loss system in Section 4. However, for M/M(x)/n-0, we have to modify the expected reward $\overline{R}_i$ of a user arriving in state i and need to modify Eq. (24). The time B a user stays in the system depends now on the arriving state i. Therefore, we need to consider the state-dependent response time B_i when a user arrives in a state with i customers. B_i reflects the total play time of a user in the cloud gaming scenario.

The response time B_i of a customer arriving in state i is a random variable. The response time B for an arbitrary user is then the mixture distribution of the conditional response times B_i which are observed with probability $x^*(i) = p_i$. Due to PASTA, the probability $x_a(i)$ that the system is in state i for an arriving customer is the same as the steady-state system probability $x(i)$. It is $x_a(i) = x(i) = p_i$ for $i = 0, 1, \ldots, n$. We will utilize the mixture distribution in Eq. (29) and the corresponding probability density function (PDF) to show that the system-centric reward of a tagged customer is identical to the user-centric reward.

$$B = \mathrm{MIX}\left((B_0, B_1, \ldots, B_{n-1}), (p_0, p_1, \ldots, p_{n-1})\right)$$
$$\text{with PDF } b(t) = \sum_{i=0}^{n-1} b_i(t) \cdot p_i \tag{29}$$

As a key result, we obtain that the system-centric reward $\overline{S}_1$ of a tagged customer is identical to the user-centric reward $\overline{R}$.

Theorem 5 (*User-Centric and System-Centric Reward For M/M(x)/n-0*)**.** *For an M/M(x)/n loss system with state-dependent service rates, the expected user-centric reward $\overline{R}$ is identical to the system-centric reward $\overline{S}_1$ from the perspective of tagged user.*

$$\overline{R} = \overline{S}_1 = \frac{\overline{S}_\Sigma}{\mathrm{E}[X]} \qquad M/M(x)/n\text{-}0 \tag{30}$$

To prove Eq. (30), we consider the system from the perspective of a tagged customer. The system state is described by the random variable X^* with steady-state probabilities $x^*(i)$ for $i = 0, 1, \ldots, n-1$. In case of $X = n$, an arriving customer is blocked. Hence, a tagged customer who is accepted will only see the system in the states $i = 0, 1, \ldots, n-1$. The steady-state system probabilities are provided in Eq. (18) for the tagged customer perspective M/M(x)/n* with n*=n-1 and $\overline{R}_i$ from Eq. (24), see Fig. 3(b).

Proof.

$$\overline{R} = \sum_{i=0}^{n-1} \underbrace{x_A^*(i)}_{x^*(i)} \cdot \overline{R}_i = \sum_{i=0}^{n-1} x^*(i) \int_{t=0}^{\infty} b_i(t) \frac{1}{t} \int_{\tau=0}^{t} \sum_{k=0}^{n-1} \underbrace{r_k^*}_{r_{k+1}} \cdot x_{\tau|i}^*(k)\, d\tau\, dt \tag{31a}$$

$$= \int_{t=0}^{\infty} \frac{1}{t} \int_{\tau=0}^{t} \sum_{k=0}^{n-1} r_{k+1} \underbrace{\sum_{i=0}^{n-1} x^*(i) x_{\tau|i}^*(k) b_i(t)}_{\text{see Eq. (32)}}\, d\tau\, dt \tag{31b}$$

$$= \int_{t=0}^{\infty} \frac{1}{t} \int_{\tau=0}^{t} \sum_{k=0}^{n-1} r_{k+1} \cdot x^*(k) \cdot b(t)\, d\tau\, dt \tag{31c}$$

$$= \sum_{k=0}^{n-1} r_{k+1} \cdot x^*(k) \int_{t=0}^{\infty} \frac{1}{t} \cdot b(t) \underbrace{\int_{\tau=0}^{t} 1\, d\tau}_{t}\, dt = \sum_{k=0}^{n-1} r_k^* \cdot x^*(k) = \overline{S}_1 \tag{31d}$$

Our proof utilizes the following relationship in Eq. (32) between the response time B of arbitrary users and the response time B_i of users arriving in state i. With the law of total probability, we are able to remove the condition of users arriving in state i. The random variable X_A indicates the state as seen by the arriving user. The random variable X_τ^* captures that an arriving user (tagged customer) who finds the system in state i will find the system in state k after time τ. This is a conditional random variable and the conditional probability $x_{\tau|i}^*(k)$ is therefore $P(X^* = k | X_A = i)$. The random variable of the response time of a user arriving in state i is $B_i = B|X_A = i$. The joint probability $P(X^* = k, B = t)$ means the probability that the M/M(x)/n* system is in state $X^* = k$ *and* that the response time is $B = t$.

$$\sum_{i=0}^{n-1} x_A(i) \cdot x_{\tau|i}^*(k) \cdot b_i(t) = \sum_{i=0}^{n-1} P(X_A = i) \cdot P(X_\tau^* = k | X_A = i) \cdot b_i(t) \tag{32a}$$

$$= P(X^* = k) \cdot \sum_{i=0}^{n-1} P(X_A = i) \cdot b_i(t) = x^*(k) \cdot b(t) \tag{32b}$$

With the relationship in Eq. (30), we can now provide numerical results by just deriving the system-centric reward $\overline{S}_1$ of a tagged user.

6. Numerical results for the example of cloud gaming

As an example use case, we consider a cloud gaming service and develop a Markov reward model to evaluate the QoE of the users while considering user engagement. In a recent study [19], the traffic characteristics of Google Stadia, Google's now defunct cloud-gaming solution, are investigated in detail. The different protocols involved for both signaling and video/audio contents are analyzed as well as the traffic generation patterns, *e.g.*, the packet size and inter-packet time probability distributions. The study evaluated the ability of Stadia to adapt to different link capacity conditions, including cases where the capacity drops suddenly, as well as sudden increases in the network latency. Furthermore, Google Stadia traffic is compared to other video streaming applications, showcasing the main differences between them. [20] analyzes the network characteristics of different cloud gaming services, Google Stadia, GeForce Now, and PSNow under different application settings and network conditions. As a result, Stadia used the RTP protocol and consumes up to 45 Mbps. A measurement tool called Decaf is proposed by [21]. Decaf is designed to capture and dissect the network traffic of gaming services to understand the factors affecting their performance. An open-source tool called Retina [22] enables real-time communication (RTC) traffic analysis, facilitating Stadia traffic examination [19]. A dataset of gaming sessions including the network traces is provided by [23].

A dedicated measurement study on cloud gaming is conducted (Section 6.1), which provides the parameters for the Erlang loss model for cloud gaming (Section 6.2). Furthermore, the cloud gaming QoE and the rewards are to be defined (Section 6.3). This allows to quantify the difference between user-centric and system-centric rewards (Section 6.4) as well as the impact of user engagement (Section 6.5).

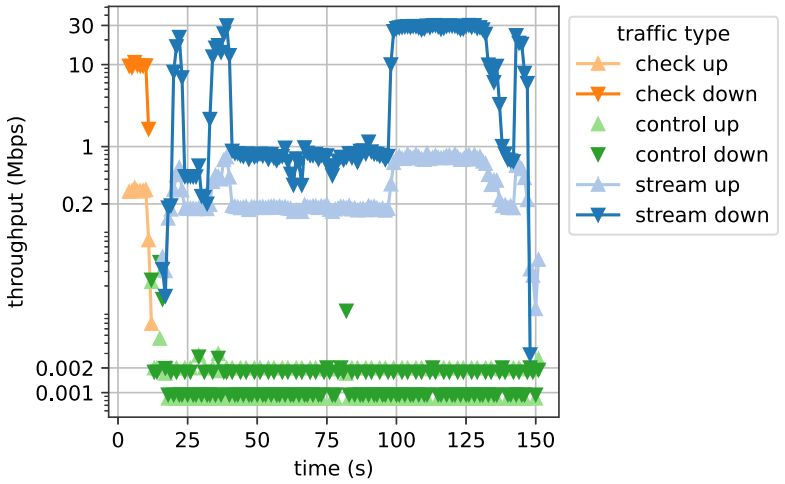


Fig. 4. Example of throughput measurements of Google Stadia without any impairments and a resolution of 1920 × 1080.

6.1. Measurement study on cloud gaming: Google Stadia

Google Stadia used 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 was carried by a QUIC connection, consistently using UDP port 44700 on the server side during our measurement campaign. A second QUIC connection was established using UDP port 44732, that carried the control inputs from the client to the server. Video encoding was selected based on the capabilities of the client, its screen resolution and the quality of the connection between client and server. Stadia had a preference for Google's VP9 codec, if a VP9 decoder was present in hardware at the client; otherwise it defaulted to the H.264 codec. The highest supported screen resolution was 4K, or 3840 × 2160 pixels. A frame rate of 60 frames per second was maintained in all cases to ensure responsiveness.

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 a 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 10 ms round trip time (RTT) was observed by constantly pinging the game server during an active session without impairments.

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 preempted 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 ms at most is tolerated, while a minimum bandwidth of 10 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.

Fig. 4 shows the throughput measurements over time of an example 150 s long run using a 1920 × 1080 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 s until about 15 s the connection test is performed. 10 Mbps probes are sent from server to client (labeled 'check down'). Only about 0.2 Mbps are sent in the opposite direction ('check up'). Afterward, 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 kbps and 2 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 adapted the video encoding to the content. We observe an initial ramp-up of downlink traffic to just below 30 Mbps after the connection check is passed. However, it quickly drops again to less than 1 Mbps at 25 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 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 Mbps during the loading of the gameplay session 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 s. What follows is the interactive gameplay phase, where the throughput jumps back up to just below 30 Mbps. Gameplay is stopped by the user after 30 s. Once again this is followed by a brief loading screen and corresponding drop below 1 Mbps throughput as well as subsequent ramp up when entering the menu where the user ends the session at 150 s.

In Fig. 5, 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 Mbps and just below 1 Mbps throughout the session depending on the content transmitted. This is not surprising, as the

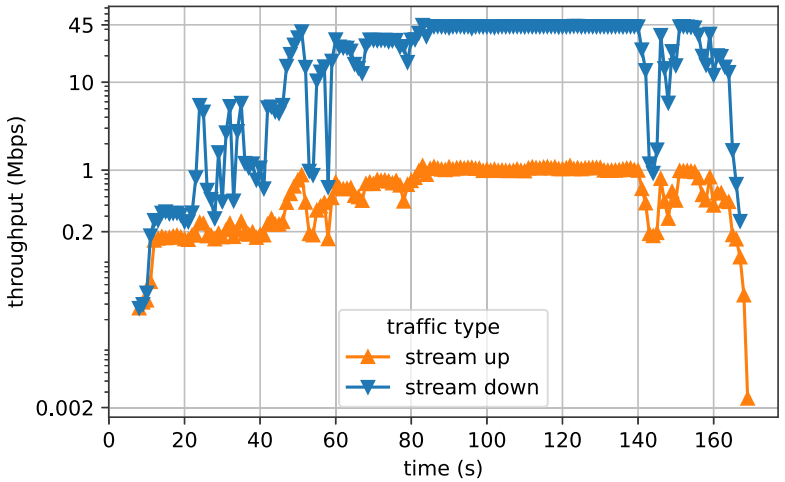


Fig. 5. Example of throughput measurements of Google Stadia without any impairments and a 4k resolution.

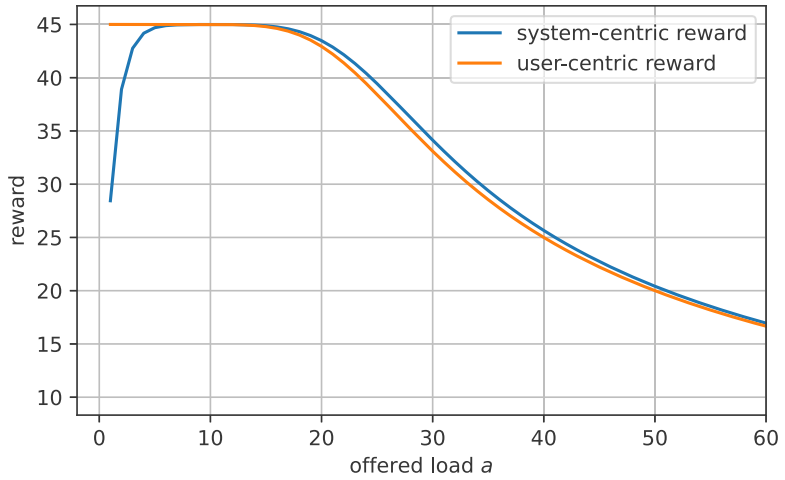


Fig. 6. Comparison of the system-centric reward $\overline{S}_0$ and the user-centric reward $\overline{R}$ for the M/M/100 loss system and varying offered load a.

uplink of the WebRTC connection mainly carries acknowledgments. 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 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 1, 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 Figs. 4 and 5. The third column represents a scenario with added delay and an 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 was reduced by Stadia to 720p to facilitate this reduction in throughput.

6.2. Erlang loss model for cloud gaming

The measurement study shows that the cloud gaming server implements admission control. If the available network capacity of a user is less than $C_0 = 10$ Mbps, the user is blocked and not allowed to enter the cloud gaming server. In our numerical results, we consider a dedicated bottleneck in the access network, where several gaming users are located. We assume a bottleneck access link with capacity C. Then, the system can be modeled as an Erlang loss system which accepts $n = \lfloor \frac{C}{C_0} \rfloor$ users. The system state X reflects the number of users in the system which independently of each other stay in the system for some time to play a game on the cloud gaming server. Thus, we arrive at the Erlang loss system M/M/n-0. However, the play time of users may depend on the actual

Table 1
Measurement results summary on google stadia.

Variable	Description	No imp. 4K	No imp.	75 ms RTT	15% loss
Codec	Video codec used for downlink stream	VP9	H.264	H.264	H.264
Resolution	Video resolution of streamed video contents	3840 × 2160	1920 × 1080	1280 × 720	1280 × 720
Throughput Ctrl. Up	Median Throughput Control Channel Uplink	0.85 kbps	0.85 kbps	0.85 kbps	1.14 kbps
Throughput Ctrl. Down	Median Throughput Control Channel Downlink	0.94 kbps	0.94 kbps	0.94 kbps	0.94 kbps
Throughput Stream	Mean Interactive Throughput Stream Channel Downlink	**43.05 Mbps**	**28.6 Mbps**	**10.23 Mbps**	**11.36 Mbps**
Packet Size Ctrl. Up	Median Packet Size Control Channel Uplink	106 byte	106 byte	106 byte	106 byte
Packet Size Ctrl. Down	Median Packet Size Control Channel Downlink	118 byte	118 byte	118 byte	118 byte
Packet Size Stream Up	Median Packet Size Stream Channel Uplink	100 byte	100 byte	100 byte	100 byte
Packet Size Stream Down	Median Packet Size Stream Channel Downlink	1228 byte	1214 byte	1179 byte	1193 byte
IAT Stream Up	Median Inter-Arrival Time Stream Channel Uplink	0.315 ms	1.25 ms	2.65 ms	1.8 ms
IAT Stream Down	Median Inter-Arrival Time Stream Channel Downlink	0.001 ms	0.002 ms	0.002 ms	0.002 ms

system state. On the one hand, if there are more users on the server, the game is more fun, users are more engaged and stay longer on the system. On the other hand, more users result in lower video bitrates degrading the QoE, which may cause users to leave the system earlier in case of lower QoE. Both cases can be modeled with state-dependent service times (play times) of the users, which is a state-dependent Erlang loss system M/M(x)/n-0. The reward of a user is the shared capacity corresponding to the video bitrate the user experiences. The literature shows that the video bitrate a user experiences during a cloud gaming session is the most crucial QoE influence factor [24], cf. Section 6.3. Thus, our reward model needs to consider the video bitrate of an individual user.

6.3. Cloud gaming QoE and rewards

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. There are various ongoing ITU-T standardization activities targeting gaming QoE, which are summarized in [25]. However, several works investigate QoE and QoE influence factors for online cloud games. [24] 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. [26] reviews QoE in cloud gaming models and also identifies bitrate as a crucial parameter of QoE.

The first works in that area [27,28] evaluated the impact of delays and packet loss in upstream and downstream on the QoE. Reducing latency is a critical requirement for cloud gaming to ensure a responsive gaming experience [28,29]. The authors in [30] investigate the video quality of commercial cloud gaming services of the first wave in 2013. [31] found through subjective studies that their test subjects experienced higher flow and immersion for high video quality. However, [32] revealed that cloud gaming 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. Other subjective studies quantified the impact of frame rate and bit rate of cloud gaming QoE [33,34] and how error correction coding may improve the quality [35]. It is essential to maintain video quality in terms of resolution and frame rate to ensure smooth gameplay [35]. A measurement study [29] showed that Google Stadia had bitrates near capacity for 15 Mbps, 25 Mbps, and 35 Mbps. [36] showed that UHD-quality cloud gaming service need up to 45 Mbps. Similar bitrates were found by [37] as well as for Stadia [1]. Further, [19] analyzes how Stadia switched between the video bitrates depending on the available network bandwidth.

From the literature above, 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 and we will use the video bitrates found in literature, as discussed above.

6.4. Difference between user-centric and system-centric reward

First, we show the difference between user-centric and system-centric rewards for the cloud gaming use case. An access link with a bottleneck capacity of $C = 1$ Gbps is considered. The video bitrates of the cloud gaming service are 10 Mbps, 30 Mbps and 45 Mbps. Users with an access capacity below 10 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 min according to [38]. Recent studies showed that around 45% of users play between one and two hours [39]. Hence, we model the system as M/M/n-0 Erlang loss system.

The reward rates are defined as follows. In state i, there are i users in the system sharing the capacity C of the bottleneck link. For our numerical results, we consider $C = 1$ Gbps. Up to 22 users can be served with a maximum bitrate of 45 Mbps for $C = 1$ Gbps. At most, 100 users can join the system with a minimum bitrate of 10 Mbps. Due to adaptive streaming being used in cloud gaming [19,24] we assume that the bottleneck's capacity is fully utilized. We model this strategy as processor-sharing, and every user gets then a fraction of the capacity. The video bitrate as main QoE influence factor reflects the reward of the user.

Fig. 6 illustrates the system-centric and user-centric reward for an M/M/100 loss system with varying load. We see that for high loads, there is no significant difference between $\overline{S}_0$ and $\overline{R}$. However, for small loads, an arriving accepted user will get the maximum

video bitrate of 45 Mbps as indicated by the user-centric reward curve. However, the system-centric reward $\overline{S}_0$ will consider the idle system and reward it with $r_0 = 0$ Mbps. Therefore, strong differences are observed between the two curves in Fig. 6. Thereby, both rewards reflect the desired perspectives. On one hand, the system-centric reward evaluates the entire system and considers a reward for the entire system. If the system is empty, the system is not used and does not generate a reward. In that sense, the system is overprovisioned and not efficiently operated. The system could be improved by switching the system off if the system is idle, *e.g.*, to save costs like energy consumption, since no reward is generated. On the other hand, the user-centric reward reflects the expected reward for a user over the entire cloud gaming session, *i.e.*, over the service consumption phase. This expected user-centric reward is then the average video bitrate as the main QoE influence factor over the cloud gaming session. The capacity of the bottleneck is shared among all active users, determining the video bitrate of the individual user.

6.5. Impact of user engagement

Next, we show how to integrate user engagement in the Markov reward model. The QoE experienced by the user as well as other factors like the number of gamers may trigger the behavior and engagement of users. This analysis serves as a blueprint on how to integrate QoE or QoE influence factors as well as user engagement in a user-centric analysis through a Markov reward model. The user engagement is modeled with state-dependent service rates. We differentiate two scenarios. First, we consider that the actual video bitrate and the resulting QoE influences the play time of a user. To this end, we assume that the mean play time T_i^{QoE} in a certain state i is linearly correlated with the obtained video bitrate, *i.e.*, the reward r_i.

$$T_i^{QoE} = c_1 \cdot r_i \quad \text{with } c_1 = \frac{n \cdot \mathrm{E}[\,B\,]}{\sum_{i=1}^{n} r_i} \qquad \text{(QoE engagement)}$$

The constant c_1 is determined in such a way that the average time per state corresponds to $\mathrm{E}[\,B\,] = 22$ min according to [38]. Thus, $\frac{1}{n}\sum_{i=1}^{n} T_i^{QoE} = \mathrm{E}[\,B\,]$.

On the other hand, we may consider a scenario where the number of users increases the game play fun and triggers longer play times of users. The more users are at the gaming server, the longer the users will play. We assume a linear relationship between the mean play time T_i^{play} per state i and the number i of users at the game server. The constant is determined such that $\frac{1}{n}\sum_{i=1}^{n} T_i^{play} = \mathrm{E}[\,B\,]$.

$$T_i^{play} = c_2 \cdot i \quad \text{with } c_2 = \frac{2\mathrm{E}[\,B\,]}{n+1} \qquad \text{(other player engagement)}$$

Finally, we consider a mixed scenario. The play time T_i^{mix} is a weighted sum of T_i^{QoE} and T_i^{play}.

$$T_i^{mix} = c_3 \cdot \left(w^{QoE} \cdot T_i^{QoE} + w^{play} \cdot T_i^{play} \right)$$
$$\text{with } c_3 = \frac{n\mathrm{E}[\,B\,]}{\sum_{i=1}^{n} T_i^{mix}} \qquad \text{(mixed scenario)}$$

For the numerical results, we assume $w^{QoE} = 1.5$ and $w^{play} = 1.0$. We will compare the results to an Erlang loss system with $\mu = 1/\mathrm{E}[\,B\,]$.

Fig. 7 shows that the user-centric rewards for the mixed scenario and the M/M/n system with the same service rate for every state are very close for any arrival rate of users. However, the engagement of users if there are more other players (labeled as 'other player eng.' in Fig. 7) leads to very different results. The more users are there, the longer the player wants to stay. Thus, the system tends towards a fully loaded system, which means lower video bitrates and thus user-centric reward. On the other hand, if there are only a few people, they will not stay long, yielding high rewards for low arrival rates. Considering the QoE engagement model, the user-centric reward is higher than for the M/M/n loss system without state-dependent service rates. Users experiencing high QoE are more engaged and stay longer. The user-centric reward increases. However, if users are staying longer, then the system load may increase, *i.e.*, more users may be online in parallel. Then, users get lower bitrates, resulting in lower QoE and shorter gaming times. Therefore, the user-centric reward is higher for the QoE engagement scenario than for the M/M/n loss system.

7. Conclusions and discussions

The key contribution of the paper are the definitions of system-centric and user-centric rewards. The latter aims at analyzing the QoE of an individual user. To the best of our knowledge, we are the first to analyze of relationships of system-centric and user-centric rewards for the Erlang loss system M/M/n-0 and the state-dependent Erlang loss system M/M(x)/n-0. 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 that 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. We also show that the same relationships are valid when considering state-dependent service rates. In the case of cloud gaming, this may take into account the user's engagement due to the QoE and the video bitrate obtained or the number of players in a game. We provide some numerical results to highlight the difference of user-centric and system-centric rewards for the cloud gaming use case. In addition, we take a look at the impact of engagement on user-centric rewards.

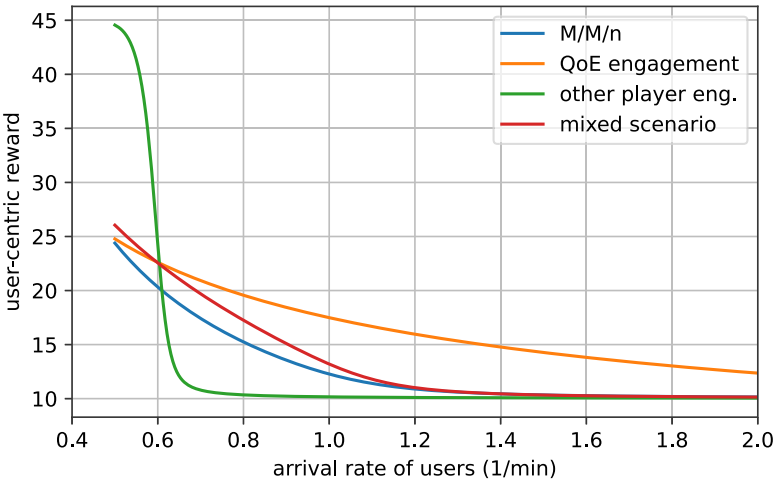


Fig. 7. M/M(x)/n: The user-centric reward of the state-dependent M/M(x)/n system depending on the arrival rate of users.

To the best of our knowledge, our works are the first (1) that define user-centric rewards to apply Markov reward models for the analysis of QoE or QoE indicators. Our analysis is the first that shows (2) that the session-based user-centric rewards can be derived using appropriate system-centric rewards from the perspective of tagged customers. This work gives relevant and important insights in how to integrate the user's perspective in the analysis of Markov reward models. It is a blueprint for the user-centric analysis of other services beyond cloud gaming and takes into account QoE as well as user engagement.

For using the relationships developed in this manuscript ($S_1 = \overline{R}$), the following assumptions must be met. The system must be modeled as Erlang loss model. The users stay in the system and consume the service for the service usage time, *i.e.*, the service time of the Erlang model. The application or the service implement admission control, which is reasonable to guarantee a minimum QoE for its users. The most critical part is the user-centric reward. For video services, the video bitrate is one important influence factor, which can be analyzed with our Markov reward model. To be more precise, the average (or aggregated) video bitrate over the session is captured as reward. In general, if the reward of a user can be aggregated or averaged over time, the equations can be applied. The power of our model is, however, the definition of the user-centric reward, which allows deriving more complex relationships beyond average video bitrates towards quantification of QoE. For example, there may be QoE influence factors like the minimum video bitrate or memory and recency effects. In particular, the peak-end rule may be important for the QoE of some applications in addition to the average quality [40]. It means that the users' experience is dominated largely by its peak (*i.e.*, most severe quality degradation) and at its end (*i.e.*, recency effect). With the definition of the user-centric reward, we can capture such relationships towards QoE. The recency term may be derived by considering the last seconds Δt in the integral $\int_{\tau=t-\Delta t}^{t}$ of the definition of R. The (negative) peak quality may be quantified using quantiles, *e.g.* 10%-percent quantile $Q_{10\%}[X^*]$ – however, this requires further subjective studies, which are appropriate quantiles. Nevertheless, the overall user-centric rewards towards a more sophisticated QoE model may be the weighted sum of the terms, as developed in [40] through subjective user studies. Note for the cloud gaming use case in the article, we were focusing on the average video bitrate as main QoE influence factor, *i.e.*, $\overline{R} = R_{\text{avg}}$. For more advanced QoE models, we may consider the weighted sum of average, peak, and recent (end) video bitrate experienced by a user as the user-centric reward.

$$\overline{R} = w_1 \cdot R_{\text{avg}} + w_2 \cdot R_{\text{end}} + w_3 \cdot R_{\text{peak}} \qquad \text{(QoE model)}$$

$$R_{\text{avg}} = \sum_{i=0}^{n-1} x^*(i) \int_{t=0}^{\infty} b_i(t) \frac{1}{t} \int_{\tau=0}^{t} \mathrm{E}[\, r^*(X^*_{\tau|i}(k))\,]\, d\tau\, dt \qquad \text{(average)}$$

$$R_{\text{peak}} = \sum_{i=0}^{n-1} x^*(i) \int_{t=0}^{\infty} b_i(t) \frac{1}{t} \int_{\tau=0}^{t} Q_{10\%}[r^*(X^*_{\tau|i}(k))]\, d\tau\, dt \qquad \text{(peak)}$$

$$R_{\text{end}} = \sum_{i=0}^{n-1} x^*(i) \int_{t=0}^{\infty} b_i(t) \frac{1}{t} \int_{\tau=t-\Delta t}^{t} \mathrm{E}[\, r^*(X^*_{\tau|i}(k))\,]\, d\tau\, dt \qquad \text{(end)}$$

From the example, it gets obvious that the system-centric perspective does not allow deriving some of those features. As we have shown in the manuscript, we may compute $R_{\text{avg}} = \overline{S}_1$ by utilizing the system-centric reward of a tagged customer for the Erlang loss system. However, the recency (end) term R_{end} cannot be derived from a steady-state system distribution X, but requires considering the system evolution as done in the definition of R_{end}. Alternatively, this memory effect may be integrated by adapting the system state capturing previous states, as done in [41]. In future work, terms like R_{peak} need to be analyzed and evaluated with subjective studies how to express them with system state distributions. Furthermore, correlations between the terms $R_{\text{avg}}, R_{\text{peak}}, R_{\text{end}}$ need to be considered. The relation between the user-centric reward $\overline{R}$ and system-centric reward $\overline{S}_1$ need to be analyzed in detail, as a simplification of the computational intensive derivation of $\overline{R}$ is desired.

Table 2
Notation and variables for the Markov reward model of the M/M(x)/n loss system with state dependent service rates μ_i.

Var.	Description
$\mathbb{S}$	State space of the system M/M(x)/n-0: $\mathbb{S} = \{0, 1, \dots, n\}$
q_{ij}	Transition rate from state i to state j for $j \neq i$
$\mu_i = q_{i+1,i}$	Service rate from state i to state $i-1$ for $i = 1, \dots, n$
$\lambda = q_{i,i+1}$	Arrival rate for $i = 0, \dots, n$
q_i	Transition rate for leaving state i: $q_i = \sum_{i \neq j} q_{ij}$
q_{ii}	$q_{ii} = -q_i$
$\mathcal{Q}$	Rate matrix $\mathcal{Q} = \{q_{ij}\}$ of size $\lvert\mathbb{S}\rvert \times \lvert\mathbb{S}\rvert$
$\mathcal{P}(t)$	Transition probability matrix for the time interval t
$x(i,t)$	Probability that the system is in state i at time t
$x(i)$	Steady state probability that the system is in state i
$\mathcal{X}(t)$	State probability vector at time t $\mathcal{X}(t) = \big(x(0,t), x(1,t), \dots\big)$
$\mathcal{X}(0)$	Initial state of the system
$\mathcal{X}$	Steady state probability vector $\mathcal{X} = \lim_{t\to\infty} \mathcal{X}(t) = \big(x(0), x(1), \dots, x(n)\big)$
$\mathbb{S}^*$	State space of the system from the perspective of a tagged customer: $\mathbb{S}^* = \{0, 1, \dots, n-1\}$
$\mathcal{Q}^*$	Rate matrix $\mathcal{Q} = \{q_{ij}\}$ of the system for a tagged customer with size $\lvert\mathbb{S}^*\rvert \times \lvert\mathbb{S}^*\rvert$
$x^*_{\tau\|i}(k)$	Probability that an arriving user (tagged customer) who finds the system in state i will find the system in state k after time τ $(i, k = 0, 1, \dots, n-1)$
$\mathcal{X}^*_{\tau\|i}$	State probability vector that a user arriving in state i stays in the system for time τ and will then find the system in one of the system states: $\mathcal{X}_{\tau\|i} = \big(x_{\tau\|i}(0), x_{\tau\|i}(1), \dots, x_{\tau\|i}(n-1),\big)$
$x^*(i)$	Steady state probability of the M/M/n* to be in state i without arriving user, *i.e.* $n^* = n-1$
$x^*_A(i)$	Steady state probability of the M/M/n* from the perspective of an arriving customer: $x^*_A(i) = x(i)$ due to PASTA property
$\mathcal{X}^*$	Steady state probability vector of the M/M(x)/n* with $n^* = n-1$: $\mathcal{X}^* = \big(x^*(0), x^*(1), \dots, x^*(n-1)\big)$
B_i	(Conditional) response time (r.v.) of a customer arriving in state i for $i = 0, 1, \dots, n-1$; note that a customer arriving in state n is blocked
B	Response time (r.v.) of an arbitrary customer
p_B	Blocking probability of the system: $p_B = x(n)$
p_i	Probability that an arriving customer find the system in state $i = 0, 1, \dots, n$: $p_i = x(i)$ and $p_B = p_n$
r	Reward function: $\mathbb{S} \to \mathbb{R}$ and $i \mapsto r(i)$
r_i	Reward of a user in state i: $r_i = r(i)$
$\mathcal{R}$	Reward vector: $\mathcal{R} = \big(r_0, r_1, \dots, r_n\big)$
$\mathcal{R}^*$	Reward vector of arriving user: $\mathcal{R}^* = \big(r_1, r_2, \dots, r_n\big)$
$\mathcal{I}_i$	Conditional state probability vector that an arriving customer finds the system in state i; the vector has a 1 at position i and 0 otherwise: $\mathcal{I}_i = \big(0, 0, \dots, 1, 0, \dots\big)$
$\overline{R}_{t\|i}$	Expected user-centric reward of a customer arriving in state i and staying in the system for time t
$\overline{R}_i$	Expected user-centric reward of a customer arriving in state i and staying in the system according to response time B_i
$\overline{R}$	Expected user-centric reward of an arbitrary customer
$\overline{S_0}$	Expected system-centric reward
$\overline{S_1}$	Expected system-centric reward from a tagged customer's perspective

CRediT authorship contribution statement

Tobias Hoßfeld: Conceptualization, Formal analysis, Investigation, Methodology, Project administration, Resources, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Poul E. Heegaard:** Conceptualization, Formal analysis, Investigation, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing. **Martín Varela:** Conceptualization, Formal analysis, Investigation, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing. **Michael Jarschel:** Conceptualization, Investigation, Methodology, Resources, Software, Validation, Visualization, Writing – original draft, Writing – review & editing.

Data availability

No data was used for the research described in the article.

Acknowledgment

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.

References

[1] T. Hoßfeld, P.E. Heegaard, M. Varela, M. Jarschel, User-centric Markov reward model on the example of cloud gaming, in: Proceedings of the 34th International Teletraffic Congress, ITC 34, Shenzhen, China, 2022, pp. 1–9.

[2] A. Reibman, R. Smith, K. Trivedi, Markov and Markov reward model transient analysis: An overview of numerical approaches, European J. Oper. Res. 40 (2) (1989) 257–267.

[3] K.S. Trivedi, Probability & Statistics with Reliability, Queuing and Computer Science Applications, John Wiley & Sons, 2008.

[4] K.S. Trivedi, M. Malhotra, R.M. Fricks, Markov reward approach to performability and reliability analysis, in: Proceedings of the International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, IEEE, 1994, pp. 7–11.

[5] Y. Kirsal, Y.K. Ever, L. Mostarda, O. Gemikonakli, Analytical modelling and performability analysis for cloud computing using queuing system, in: Proceedings of the IEEE/ACM 8th International Conference on Utility and Cloud Computing, UCC, IEEE, 2015, pp. 643–647.

[6] J. Hou, M. Radetzki, Performability analysis of mesh-based NoCs using markov reward model, in: Proceedings of the 26th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, PDP, IEEE, 2018, pp. 609–616.

[7] S. Ahamad, et al., Some studies on performability analysis of safety critical systems, Comp. Sci. Rev. 39 (2021) 100319.

[8] L. Jin, G. Zhang, H. Zhu, W. Duan, SDN-based survivability analysis for V2I communications, Sensors 20 (17) (2020) 4678.

[9] P.E. Heegaard, K.S. Trivedi, Network survivability modeling, Comput. Netw. 53 (8) (2009) 1215–1234.

[10] L. Xie, P.E. Heegaard, 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, pp. 25–36.

[11] A.P.C. da Silva, D. Renga, M. Meo, M.A. Marsan, Small solar panels can drastically reduce the carbon footprint of radio access networks, in: Proceedings of the 31th International Teletraffic Congress, ITC 31, Budapest, Hungary, 2019, pp. 64–65.

[12] T. Hoßfeld, L. Atzori, P.E. Heegaard, L. Skorin-Kapov, M. Varela, The interplay between QoE, user behavior and system blocking in qoe management, in: Proceedings of the 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops, ICIN, IEEE, 2019, pp. 112–117.

[13] P. Tran-Gia, T. Hoßfeld, Performance Modeling and Analysis of Communication Networks, Würzburg University Press, ISBN: 978-3-95826-153-2, 2021, http://dx.doi.org/10.25972/WUP-978-3-95826-153-2, URL https://modeling.systems.

[14] J. Sastre, J. Ibáñez, E. Defez, Boosting the computation of the matrix exponential, Appl. Math. Comput. 340 (2019) 206–220.

[15] M. Fasi, N.J. Higham, An arbitrary precision scaling and squaring algorithm for the matrix exponential, SIAM J. Matrix Anal. Appl. 40 (4) (2019) 1233–1256.

[16] R.W. Wolff, Poisson arrivals see time averages, Oper. Res. 30 (2) (1982) 223–231.

[17] B. Melamed, W. Whitt, On arrivals that see time averages, Oper. Res. 38 (1) (1990) 156–172.

[18] S.L. Brumelle, A generalization of Erlang's loss system to state dependent arrival and service rates, Math. Oper. Res. 3 (1) (1978) 10–16.

[19] M. Carrascosa, B. Bellalta, Cloud-gaming: Analysis of Google Stadia traffic, Comput. Commun. 188 (2022) 99–116.

[20] A. Di Domenico, G. Perna, M. Trevisan, L. Vassio, D. Giordano, A network analysis on cloud gaming: Stadia, GeForce Now and PSNow, Network 1 (3) (2021) 247–260.

[21] H. Iqbal, A. Khalid, M. Shahzad, Dissecting cloud gaming performance with DECAF, Proc. ACM Meas. Anal. Comput. Syst. 5 (3) (2021) 1–27.

[22] G. Perna, D. Markudova, M. Trevisan, P. Garza, M. Meo, M.M. Munafò, Retina: An open-source tool for flexible analysis of RTC traffic, Comput. Netw. 202 (2022) 108637.

[23] I. Slivar, K. Bacic, I. Orsolic, L. Skorin-Kapov, M. Suznjevic, CGD: a cloud gaming dataset with gameplay video and network recordings, in: Proceedings of the 13th ACM Multimedia Systems Conference, 2022, pp. 272–278.

[24] F. Metzger, et al., An introduction to online video game QoS and QoE influencing factors, IEEE Commun. Surv. Tutor. (2022) http://dx.doi.org/10.1109/COMST.2022.3177251.

[25] S. Schmidt, S. Zadtootaghaj, S. Möller, ITU-T standardization activities targeting gaming quality of experience, ACM SIGMultimed. Rec. 13 (1) (2022) 1.

[26] A.A. Laghari, H. He, K.A. Memon, R.A. Laghari, I.A. Halepoto, A. Khan, Quality of experience (QoE) in cloud gaming models: A review, Multiagent Grid Syst. 15 (3) (2019) 289–304.

[27] M. Jarschel, D. Schlosser, S. Scheuring, T. Hoßfeld, An evaluation of QoE in cloud gaming based on subjective tests, in: Proceedings of the Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IEEE, 2011, pp. 330–335.

[28] M. Jarschel, D. Schlosser, S. Scheuring, T. Hoßfeld, Gaming in the clouds: QoE and the users' perspective, Math. Comput. Modelling 57 (11–12) (2013) 2883–2894.

[29] X. Xu, M. Claypool, Measurement of the responses of cloud-based game streaming to network congestion, in: Proceedings of the 32nd Workshop on Network and Operating Systems Support for Digital Audio and Video, 2022, pp. 22–28.

[30] K.-T. Chen, Y.-C. Chang, H.-J. Hsu, D.-Y. Chen, C.-Y. Huang, C.-H. Hsu, On the quality of service of cloud gaming systems, IEEE Trans. Multimed. 16 (2) (2013) 480–495.

[31] J. Beyer, R. Varbelow, J.-N. Antons, S. Möller, Using electroencephalography and subjective self-assessment to measure the influence of quality variations in cloud gaming, in: Proceedings of the 2015 Seventh International Workshop on Quality of Multimedia Experience, QoMEX, IEEE, 2015, pp. 1–6.

[32] A. Sackl, R. Schatz, T. Hossfeld, F. Metzger, D. Lister, R. Irmer, QoE management made uneasy: The case of cloud gaming, in: Proceedings of the 2016 IEEE International Conference on Communications Workshops, ICC, IEEE, 2016, pp. 492–497.

[33] I. Slivar, L. Skorin-Kapov, M. Suznjevic, Cloud gaming QoE models for deriving video encoding adaptation strategies, in: Proceedings of the 7th International Conference on Multimedia Systems, 2016, pp. 1–12.

[34] S. Zadtootaghaj, S. Schmidt, S. Möller, Modeling gaming QoE: Towards the impact of frame rate and bit rate on cloud gaming, in: Proceedings of the Tenth International Conference on Quality of Multimedia Experience, QoMEX, IEEE, 2018, pp. 1–6.

[35] J. Wu, C. Yuen, N.-M. Cheung, J. Chen, C.W. Chen, Enabling adaptive high-frame-rate video streaming in mobile cloud gaming applications, IEEE Trans. Circuits Syst. Video Technol. 25 (12) (2015) 1988–2001.

[36] Y. Kim, Y. Choi, Y.C. Lee, H. Han, S. Kang, E-Render: Enabling UHD-quality cloud gaming through edge rendering, IEEE Access 10 (2022) 72107–72119.

[37] X. Marchal, P. Graff, J.R. Ky, T. Cholez, S. Tuffin, B. Mathieu, O. Festor, An analysis of cloud gaming platforms behaviour under synthetic network constraints and real cellular networks conditions, J. Netw. Syst. Manage. 31 (2) (2023) 1–31.

[38] Limelight Networks, Market Research: The State of Online Gaming – 2019, Tech. Rep., Limelight Networks, 2019, URL https://www.limelight.com/resources/white-paper/state-of-online-gaming-2019. (Last accessed 24 May 2022).

[39] B. Rathakrishnan, S.S. Bikar Singh, A. Yahaya, Gaming preferences and personality among school students, Children 10 (3) (2023) 428.

[40] H.T. Tran, N.P. Ngoc, T. Hoßfeld, T.C. Thang, A cumulative quality model for HTTP adaptive streaming, in: Proceedings of the 2018 Tenth International Conference on Quality of Multimedia Experience, QoMEX, IEEE, 2018, pp. 1–6.

[41] T. Hoßfeld, S. Biedermann, R. Schatz, A. Platzer, S. Egger, M. Fiedler, The memory effect and its implications on web QoE modeling, in: Proceedings of the 2011 23rd International Teletraffic Congress, ITC, IEEE, 2011, pp. 103–110.

Tobias Hoßfeld is Full Professor and head of the Chair of Communication Networks at the University of Wurzburg, Germany, since 2018. From 2014 to 2018, he was head of the Chair "Modeling of Adaptive Systems" at the University of Duisburg–Essen, Germany. Among others, he received the Fred W. Ellersick Prize of the IEEE Communications Society. He is a member of the editorial board of IEEE Communications Surveys & Tutorials, ACM SIGMM Records, Springer Quality and User Experience, and a senior member of the IEEE.

Poul E. Heegaard is Full Professor at the Department of Information Security and Communication Technology, Norwegian University of Science and Technology (NTNU), where he also has acted both as head of department and head of the research group in Networking. He was previously Senior Scientist with SINTEF Digital (1989–1999) and then Telenor R&I (1999–2009). He is a senior member of the IEEE.

Martin Varela has been involved in QoE research since 2001, first as an M.Sc. and Doctoral student at Universite de Rennes 1, and later mostly at VTT Technical Research Centre of Finland, where he was a Principal Scientist. Since 2017, he works in the industry, but has remained active in the QoE community, working mostly on topics related to QoE Management, and QoE for WebRTC services.

Michael Jarschel received his Ph.D. degree from the University of Wurzburg in 2014. Subsequently, he worked as a (Senior) Research Engineer in the area of network softwarization at Nokia (Bell Labs) in Munich until 2021. Currently, he is working as a professor of communication networks and cybersecurity at Technische Hochschule Ingolstadt. His main research interests are in the ongoing transformation of telecommunication systems towards software and the cloud, network security & early 6G concepts.