QoE aware placement of content in edge networks on the example of a photo album cloud service

Lam Dinh-Xuan, Michael Seufert, Florian Wamser, Phuoc Tran-Gia

QoE Aware Placement of Content in Edge Networks on the Example of a Photo Album Cloud Service

Lam Dinh-Xuan, Michael Seufert, Florian Wamser, Phuoc Tran-Gia
Chair of Communication Networks, Institute of Computer Science, University of Würzburg
Email: lam.dinh-xuan|seufert|florian.wamser|trangia@informatik.uni-wuerzburg.de

Abstract—The paradigm of Software as a Service has gained great achievements in the last decade. By transferring computation and storage to the cloud and migrating services to the edge network, users benefit from using demanding services on lightweight devices. However, the user perceived quality of experience (QoE) for these services is facing the challenges of network impairments and the accessibility of users. Unlike a typical PC-based software, the cloud provides users a location-aware, flexible placement of resource for a cost effective service. The geographical placement of content is therefore one of the key factors that affects the user's satisfaction. The closer the content to the user geographically is, the faster it will be delivered to the user that will also increase the user perceived QoE.

In this work, we estimate more precisely the QoE for photo loading time in a particular usage of a photo album cloud service with regard to the influence of various parameters. Firstly, we validate a TCP throughput model and use it to calculate the photo loading time from a given photo size and network QoS. Thereafter, we formulate a mapping function to calculate the MOS value from a QoE model adding the output of the TCP model. From this mapping function, we can estimate QoE for photo loading time from a given photo size, its placement and network QoS. Our main contribution is to determine the trade-off between the size of photo and its placement to acquire a high QoE for photo loading time, which is important for the development of a photo album cloud service.

Index Terms—Cloud Services; Quality of Experience; Content Placement

I. INTRODUCTION

In recent years, the rapid growth of personal smart devices has generated a huge amount of data uploaded to the cloud. Along with the increasing diversity of Software as a Service (SaaS), a trend is the replacement of entertainment applications running on PC by a SaaS (e.g., online cloud gaming, YouTube video streaming). An edge network photo album cloud service (EPC) is another example. While a desktop-based album application manages and stores photos predominantly on a PC, an cloud-based album provides almost unlimited space to store the user's photos, accessible everywhere. Furthermore, an edge network cloud service refers to a location-aware, flexible placement of the service and the content among multiple resources in the cloud and in the edge network. This means, the service providers can decide to place the EPC in a resource-efficient manner, such that the user perceived QoE for the photo album service is high.

An EPC uses HTTP or HTTPs over TCP to deliver stored photos. Thus, the photo loading time is influenced primarily by the file size, the distance between server and client, and the network QoS. If the user has to wait too long to view or upload a photo, the user may reject using the service. In order to achieve high satisfaction with the photo album services, the challenge is to efficiently place the photo content to appropriate geographical location(s) in order to achieve a high QoE perceived by the users.

In this paper, we deploy a mapping function from content size, distance and different network QoS parameters (i.e., link capacity, delay, packet loss) to the QoE of photo loading time. This can be used to decide the placement of content. To derive this mapping function, we conduct a study with several steps which are depicted in Fig. 1.
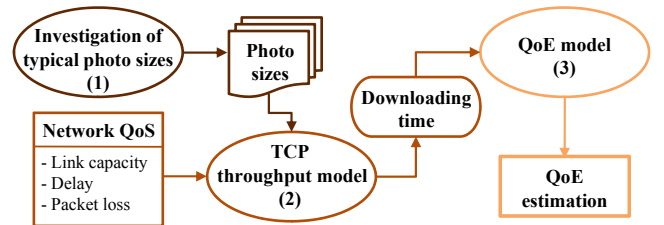

Fig. 1. Measurement Workflows

First, we investigate the properties of cloud-based photo content through a well-known photo album service (i.e., Google Photos). In this step (1), we measure the sizes of downloaded photos at different screen resolutions. The range of average photo sizes is then selected for the QoS measurements of file downloading to validate the TCP model in the next step. In the second step (2), we use a TCP throughput model proposed in [1] to calculate the downloading time of different photo sizes at various QoS parameters. We validate this TCP model in a local testbed, where the different network parameters can be configured. In the last step (3), we formulate a mapping function to calculate the MOS value from a QoE model adding the output of the TCP model. Our mapping function allows to determine the QoE for photo loading time depending on photo size, location, and the network parameters. This helps to investigate the trade-off between the size of photo and its placement in the cloud or edge network to achieve a high QoE for photo loading time, which is our main contribution of this paper.

The remainder of this paper is structured as follows. Section II highlights the background of the study and related work. Section III presents the QoS model and the measurement

setup for the model validation. Then, the QoE model and the discussion of the placement of content are described in Section IV. Finally, Section V concludes this work.

II. BACKGROUND AND RELATED WORK

In this section, we introduce the cloud service photo album (i.e., EPC) and the technology behind. Thereafter, we highlight an overview of related works.

A. Background

Image is one of the most popular content delivered over the Internet. To store and share this type of content, the common way is using a web-based photo album. An EPC is a SaaS, which allows users to upload and manage photos created by any digital device (e.g., digital camera, smart phone, etc). As a web-based service, an EPC typically uses HTTP or HTTPs to deliver stored photos over Internet. Users can access and manage an EPC using any modern web browser.

As an edge network cloud service, the EPC content may be located at the centric server or at the logical point of the network. Such a service has advantages of removing a major bottleneck at the centric server, reducing latency, utilizing efficiently computing power and storage of the edge server. Regarding to the content delivery, there is another technology introduced in [2] and [3], namely Content Delivery Networks (CDN). This network technology allows the service providers to cache a part of content on the edge-server nearby the users. The undelivered content is regularly replicated from the original server to the CDN. By doing this, users always have a high availability and high performance service. However, replicating the same content over Internet is not always a best solution for an EPC service, when each user has his own individual photo album. Instead, the EPC content is only migrated to the edge server when the users experience a low quality of EPC service due to high delay (e.g., long distance access) in the network. In other words, the trade-off between user perceived QoE, the network QoS and the placement of content is an important factor for developing such an EPC service.

B. Related Work

The QoE evaluation of cloud service as well as the relationship between QoE and QoS is widely studied. However, we observe a rare research about QoE-aware placement of content for cloud services. The concept of QoE refers to the overall level of customer's enjoyability with a service [4]. The QoE is evaluated using Mean Opinion Scores (MOS) [5]. Regarding to the web-based services, QoE has a strong relationship with the network QoS. The IQX hypothesis is proposed in [6], which described a natural and generic exponential relationship between QoE and network QoS. Meanwhile, [7] reported a logarithmic relationship between QoE and network QoS. However, the relation between the QoE of a specific application and the network QoS highly depends on the application.

In [8], Mok et al. investigated the relationship between the QoE of HTTP video streaming and network QoS using analytical models and empirical evaluation. This study is similar to our study in the method of research but different in the objective of study. In [9], Casas et al. provided the results of concrete cloud QoE studies, in which Cloud Storage and File Synchronization (CSFS), Remote Virtual Desktop (RVD) and telepresence system such as Microsoft Lync Online were conducted by subjective lab experiments. Meanwhile, HTTP Video Streaming like Youtube was evaluated by field trials approach. However, it is different from our study where we consider the placement of a photo album cloud service. The relationship between the waiting times of interactive data services and QoE is discussed in [10] and [11]. The authors focus on the time perception and its relation with the user's satisfaction rather than the trade-off between the placement of content and QoE as our main consideration. The authors explained the logarithmic relationship between the user perceived QoE and the photo loading time which benefits us as a QoE reference model.

III. QoS MODEL AND FILE DOWNLOADING MEASUREMENTS

In this section, we first describe the TCP throughput model used for our evaluation with the input parameters in order to figure out the relationship between network QoS and photo QoE. Thereafter, we describe the testbed setup for the measurements which are used to validate the accuracy of the TCP throughput model.

A. TCP Throughput Estimation Model

Despite the fact that the TCP CUBIC is currently implemented in Linux operating systems, most of TCP CUBIC throughput models are complex analytical models for special purposes, e.g., in the context of wireless environments [12], or for multiple TCP connections [13]. As the focus of this paper is not to provide accurate results but rather to present the methodology and to conduct a qualitative study, we employ a simpler TCP Reno throughput model proposed by Padhye et al. [1]. This model has an intuitive throughput calculation and fits well to the available parameters in our measurement scenario. Note that the methodology presented here can nevertheless be applied to the recent, more accurate TCP CUBIC models.

In [1], TCP throughput is computed as follows

$$T_p \approx \min\left(\frac{W_{max}}{RTT}, \frac{1}{RTT\sqrt{\frac{2bp}{8}} + T_0 \min\left(1, 3\sqrt{\frac{3bp}{8}}\right)p\left(1 + 32p^2\right)}\right), \quad (1)$$

T_p is the estimated TCP throughput, W_{max} is maximum TCP window size, $W_{max} = 64\,\text{KBytes}$, p is packet loss rate, b is the number of packets that are acknowledged by an received ACK, typically b is 2. RTT is the round trip time, T_0 is the retransmission time out. To achieve the objective of the study, we calculate the RTT parameter in more detail. In fact, RTT is affected by link capacity and additional delay in network. It is the sum of transmission, propagation, and additional delay. Thus, RTT is calculated as

$$RTT = \frac{bL}{C} + d + D_{pg}, \quad (2)$$

where L is average packet length, C is available bandwidth of the link, d is additional delay and D_{pg} is propagation delay. In [14], Balej et al. proposed a geographic distance estimation based on round trip time, where the propagation delay is calculated as

$$D_{pg} = \frac{2s}{c \cdot r}. \qquad (3)$$

where s is geographic distance between server and client, r is parameter of the velocity of signal propagation, $r = 0.335$. c denotes speed of light in vacuum. The propagation delay calculated by Eq. (3) can give us the hint about the placement of content in the cloud.

B. Testbed Setup and Methodology

To validate the TCP throughput model, we measure the TCP throughput of file downloads in a testbed. The results show the behavior of TCP throughput under the impact of different network parameters. First, we specify the range of file sizes for the download measurements by investigating a real web-based photo album. We choose Google Photos as an example of a well-known cloud photo album. By manually uploading and browsing a photo at the different screen resolutions, we summarize the properties of rescaled photos in our tests in Tab. I. The photo uploaded to Google Photos is taken from a typical digital camera. It has 5184×3456 pixels in resolution and 5711 KBytes in size. Table I shows that the resolution as well as the size of original photo is rescaled at the different screen resolutions. This adaptation is also explained in [15] and [16]. From this result, we select the range of file sizes corresponded with the rescaled photo sizes, which are 128, 256, 512 and 1024 KBytes.

TABLE I
RESCALED PHOTOS AT DIFFERENT SCREEN RESOLUTIONS

Screen Resolution	Photo Resolution	Size (KByte)
1920 x 1200	1658 x 1105	472
1680 x 1050	1433 x 956	372
1440 x 900	1208 x 805	276
1280 x 800	1058 x 705	218

To measure the TCP throughput of file downloads, we setup a testbed which is schematically depicted in Fig. 2. It consists of three PCs and one server running Ubuntu 12.04 LTS. The given files are transferred from Server to Client via the server running NetEm [17]. This network emulator server can adjust available bandwidth, delay and packet loss of the connection. We use a separated Control PC to manage the testbed via SSH protocol. To transfer files from Server to Client, we use the Linux `netcat` command. We use `tcpdump` to capture the packets. The TCP throughput is then calculated by the total length and duration of packets. For the later evaluation, we emulate the different network QoS on NetEm. These parameters are the typical network characteristics of the Internet that are documented in [18] and [19] as well. The link capacity is also limited to evaluate the impact of available bandwidth on the TCP throughput.

Table II specifies the different network parameters we emulate on NetEm. The *Baseline* round trip time is measured in the testbed without any configuration on the NetEm server and we observe an average round trip time of 0.4 ms over 1000 packets.

TABLE II
TYPICAL NETWORK PARAMETERS EMULATED BY NETEM

Network QoS	Parameters
Available Bandwidth (KByte per second)	128; 256; 512; 1024
Round Trip Time (millisecond)	Baseline; 250; 500; 750; 1000
Packet Loss (%)	0; 2.5; 5; 7.5; 10

C. Validation

In this section we present the comparison of the values generated by the model (1) and the results obtained from the measurements. To calculate the TCP throughput from the model, we execute all available network parameters presented in Tab. II. Besides, RTT is calculated in Eq. (2) with $D_{pg} \approx 0$ due to the short distance between Server and Client in the testbed. T_0 is the TCP retransmission timeout defined in RFC document [20] and it is usually estimated by RTT and its variation. However, we observe a negligible round trip time variation in the testbed, therefore $T_0 \approx RTT$. The packet size is averaged through the `tcpdump` trace, given by $L = 2557$ KBytes. We observe that the behavior of TCP throughput is mostly similar for different file sizes. Hence, we only show the measured TCP throughput of the file 512 KBytes as an example in the following graphical results.

In all figures, the TCP throughput in KBytes is depicted on the y-axis. The solid lines and the pluses represent the TCP throughput obtained from the model (1) and the measurements, respectively. The bars on the pluses show 95% confidence intervals over 30 runs. To validate the discrepancy between the model and the measurements, we calculate the relative error as

$$(|x_m - x|)/x_m * 100 \quad (\%) \qquad (4)$$

where x_m and x are the throughput values calculated from the TCP model and obtained from the measurements, respectively.

Figure 3 shows the impact of delay and packet loss on the TCP throughput. The file is transferred at link capacity $C = 3750$ KByte/s to avoid bottleneck at both sender and receiver.
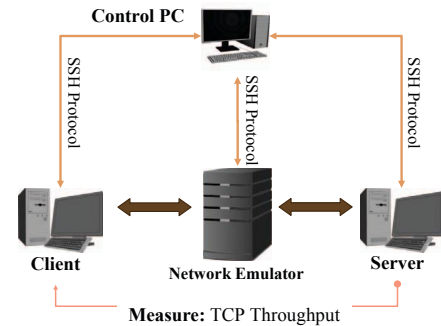

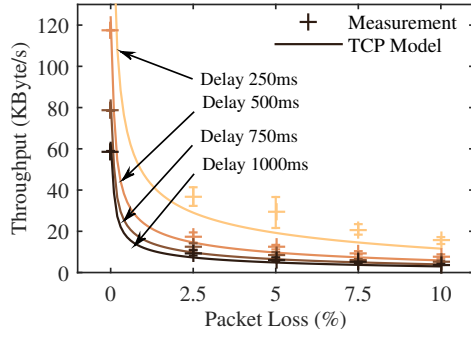Fig. 2. Overview of Testbed Setup for File Download Measurements

Fig. 3. Impact of Delay and Packet Loss on TCP Throughput

In the figure, the x-axis indicates different packet loss rates ranged from 0 to 10 %. The different colors of the solid lines and the pluses represent the TCP throughput under the impact of specific network delay combined with packet loss. For sake of readabilities the y-axis is cropped to 130 KByte/s, but the maximum actual value is 233.90 KByte/s. As displayed on the figure, we observe that the results from the model and the measurements agree with each other.

Figure 4 shows the impact of available link capacity C and packet loss on TCP throughput. The x-axis indicates the different packet loss rates. The darker lines and pluses depict the TCP throughput calculated and measured at lower link capacities, respectively. The graph shows that there are small errors between the results calculated from the model and measured from the tests.
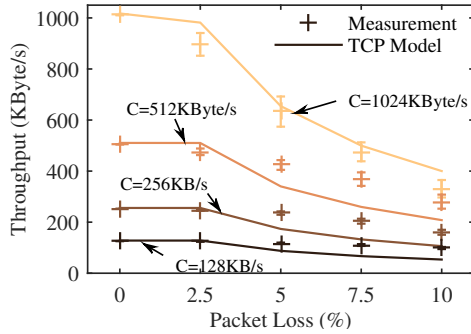

Fig. 4. Impact of Available Bandwidth and Packet Loss on TCP Throughput

Next, we investigate the impact of the path Bandwidth-Delay Products (BDP) described in [21] on the TCP throughput without the presence of packet loss. The RTT is therefore recalculated as

$$RTT = \frac{W_{max}}{C} + d + D_{pg} \qquad (5)$$

The results from the measurements and the model (1) are presented in Fig. 5. The x-axis shows the different delay values ranged from *Baseline* to 1000 ms. The lines and the pluses with different colors represent TCP throughput at various link capacities C. From the figure, the TCP throughput calculated from the model and obtained from the measurements are proximately close to each other. We observe a majority of the results have errors less than 40 % calculated by Eq. (4).
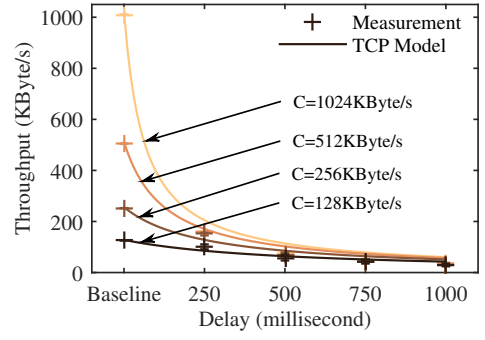

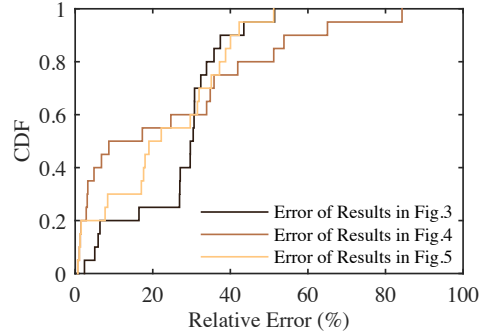Fig. 5. Impact of BDP path on TCP Throughput

Fig. 6. The CDF of Relative Error Between TCP Model and Measurements

Figure 6 shows the cumulative distribution function (CDF) of relative errors between the results from the TCP model and the measurements, where the x-axis indicates the error rates, the different lines shows the relative errors of different experiments. From this figure, we observe that there are approximately 60 % of the measurements values have errors less than 30 % compared to the TCP model. To close this section, we conclude that the TCP throughput model (1) with the RTT calculated in Eq. (2) and Eq. (5) has sufficient reliability to be deployed in general measurements.

IV. THE QOE MODEL AND THE PLACEMENT OF CONTENT

In this section, we describe a QoE study for photo loading time. The QoE is estimated as a function of MOS given by the duration of loading a photo. Meanwhile, the downloading time of a photo with a given photo size and network QoS can be calculated by the TCP model described in Section III. Therefore, this time factor plays a role as a bridge in order to connect the TCP throughput model with the QoE model. The remaining of this section presents our discussion about the placement of content with regard to the user's satisfaction.

A. The QoE Estimation Model

In [10], Egger et al. contributed a study of waiting times in the context of interactive applications. They examined the QoE for several web applications including web browsing, email processing, VoIP, as well as video streaming. The authors concluded that the user perceived QoE for web-based services has a logarithmic decrease along with the increase in waiting

time. In addition, the logarithmic behaviour of QoE regarding to the time factor is also reported in [7] and [22]. Regarding to the waiting times in the context of browsing photos, in another paper [11], Egger et al. proposed a logarithmic fitting function to describe specifically the relationship between picture loading time and the user perceived QoE as follows

$$QoE(t) = -0.80\ln(t) + 3.77, \qquad (6)$$

where $QoE(t)$ is the function of MOS given by the picture loading time t. The authors measured the goodness of fit by calculating the coefficient of determination r^2 which has value of 1.00 in this case. The verification of the model can be found in [11]. Despite the fact that QoE model has been widely studied (e.g. in [6] [7] [22]), we choose model (6) as the mapping function due to its high reliability and it fits well to our measurements where photo loading time t can be calculated by the TCP model.

B. The Placement of Content

We present in this subsection a trade-off between the photo properties, its geographical placement and network QoS in which the user perceived QoE can be estimated from these parameters. Indeed, from model (1), we calculate the duration of loading a photo as

$$t = \frac{size}{T_p(C, rtt_s, p)}, \qquad (7)$$

where T_p is the TCP throughput estimated by monitoring the network QoS with C, rtt_s, p are link capacity, round trip time and packet loss, respectively. rtt_s is estimated according to Eq. (2) and Eq. (3) with the distance s between server and client. $size$ is given size of a photo. From Eq. (6) and Eq. (7), the estimated QoE model based on network parameters, photo size and distance is formulated as

$$QoE(size, s) = -0.80\ln\left(\frac{size}{T_p(C, rtt_s, p)}\right) + 3.77. \quad (8)$$

Equation (8) can completely compute at which photo size or level of network QoS to gain an acceptable QoE. Figure 7 shows an example of the estimated QoE for loading a photo under the impact of network delay and packet loss. The x-axis indicates the packet loss rates. The y-axis shows the estimated

MOS values which represent the user perceived QoE. The MOS can take the following values: (1) bad; (2) poor; (3) fair; (4) good; (5) excellent. The darker lines depict the QoE behaviour at higher delay. As shown in Fig. 7, when the packet loss is not presence in the network, the QoE for photo loading is better at smaller delay. However, the MOS value decreases dramatically with the increase of delay and packet loss. This is because of the retransmission of lost packets take longer and consequently, the time until information is successfully transmitted between server and client increases, which results in a rapid drop of MOS values as indicated in Eq. (6).

From the equations (2), (3), (5) and (8), the trade-off between the size of photo and its geographical placement can be estimated. Figure 8 shows the relationship between the distance and the user perceived QoE represented by MOS values. We assume that the photos taken from Tab. I are transferred on a typical ADSL link, which has downstream rate of 8 Mbit/s following the ITU-T G.992.1 standard. Packet loss is assumed not to occur on the link, the round trip time is calculated by Eq. (3) and Eq. (5). In the figure, the x-axis shows the various distances between server and client in kilometer, the y-axis indicates the corresponding estimated MOS values which represent the user perceived QoE. The darker lines depict the QoE behaviour of larger photo sizes.
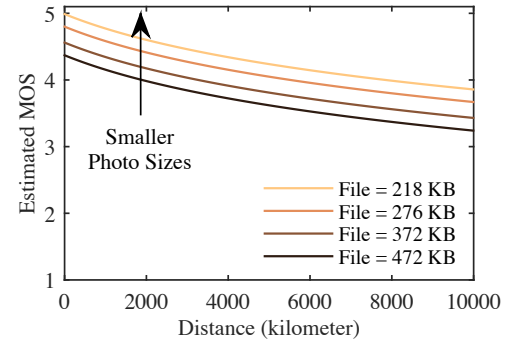
Fig. 8. Estimated QoE for Photo Loading at Different Distances

The figure shows that the MOS values decline gradually at every longer distance and the smaller photo sizes (i.e., smaller photo resolutions as shown in Tab. I) gain better QoE. We observe that the QoE for loading a photo 472 KBytes with 1658×1105 pixels in resolution is acceptable if the distance between server and client shorter than 4000 kilometers. Besides, the photo has 218 KBytes in size with 1058×705 pixels in resolution still gains a good QoE even it is transferred through a long distance. However, the packet loss may occur on the link and the probability of occurrence might be higher at longer distance. In this case, the MOS values will decrease rapidly as described in Fig. 7. To solve this problem, the service providers can rescale the photos resolution or reduce the photos size to meet an acceptable QoE as indicated in Eq. (8). After all, if both adjusting photos quality and improving the network QoS do not meet the user's satisfaction, a migration of the user's photo album to the edge-server nearby the geographical location of the user may be recommended.
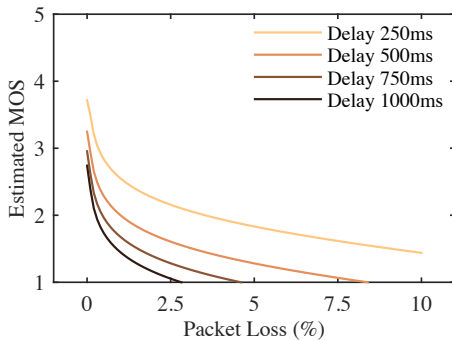

Fig. 7. The Impact of Delay and Packet Loss on QoE for Photo Loading $C = 3750 KByte/s$, $size = 218 KBytes$ in Tab. I, $D_{pg} \approx 0$ in testbed

V. Conclusion

Although the Internet users can benefit enormously from the cloud services and the SaaS paradigm, the challenge is how to achieve a high user perceived QoE for these Internet-based applications. An edge network cloud service refers to a location-aware, flexible placement of resource application. Specifically, the placement of its content is one of the key factors that affects the user's satisfaction. A long distance access is characterized by a high delay and possible packet loss which results in a longer data loading time. Thus, the user perceived QoE for the service is dramatically dropped.

To increase the performance of services, the placement of content must be considered. The closer the content to the user geographically is, the faster it will be delivered to the user that will also increase the user perceived QoE. To achieve this perception, we propose in this study a trade-off between the size of photo and its placement to acquire a high QoE for photo loading time in a particular usage of a cloud service, an EPC. We first validate a TCP throughput model and use it to calculate the photo loading time from a given photo size and network QoS. Thereafter, we map a QoE logarithmic function to the TCP throughput model. From this mapping function, we can estimate QoE for photo loading time from a given photo size, its placement and network QoS. Our results show that, we can achieve a good QoE of photo loading time by optionally adjusting the size of photo, improving network QoS or moving the EPC nearby the user. Our contribution may help cloud service providers to have another method to estimate the behaviour of QoE for photo loading time based on various parameters. Future work may extend this study to other cloud services that might benefit from a movement to the edge.

References

[1] J. Padhye et al. "Modeling TCP throughput: A Simple Model and Its Empirical Validation". In: *ACM SIG-COMM Computer Communication Review*. 1998.

[2] G. Bertrand et al. *Use Cases for Content Delivery Network Interconnection*. RFC 6770. Nov. 2012.

[3] M. Reiners and W. van der Bijl. "Content Delivery Networks". In: *Communications Network* 3.3 (2004).

[4] R. Jain. "Quality of Experience". In: *IEEE MultiMedia* (Jan. 2004).

[5] ITU-R Recommendation P.800. "Methods for Subjective Determination of Transmission Quality". In: *International Telecommunication Union* (Aug. 1996).

[6] M. Fiedler, T. Hoßfeld, and P. Tran-Gia. "A Generic Quantitative Relationship between Quality of Experience and Quality of Service". In: *IEEE Network Special Issue on Improving QoE for Network Services* (June 2010).

[7] ITU-T Recommendation G.1030. "Estimating End-to-End Performance in IP Networks for Data Applications". In: *International Telecommunication Union* (Nov. 2005).

[8] R. K. Mok, E. W. Chan, and R. K Chang. "Measuring the Quality of Experience of HTTP Video Streaming". In: *IFIP/IEEE International Symposium on Integrated Network Management (IM)*. Dublin, Ireland, May 2011.

[9] P. Casas and R. Schatz. "Quality of Experience in Cloud services: Survey and Measurements". In: *Computer Networks* 68 (2014), pp. 149–165.

[10] S. Egger et al. "Waiting Times in Quality of Experience for Web Based Services". In: *Fourth International Workshop on Quality of Multimedia Experience (QoMEX)*. IEEE. Melbourne, Australia, July 2012.

[11] S. Egger et al. "Time is Bandwidth? Narrowing the Gap between Subjective Time Perception and Quality of Experience". In: *2012 IEEE International Conference on Communications (ICC)*. Ottawa, Canada, June 2012.

[12] W. Bao, V. W. Wong, and V. Leung. "A Model for Steady State Throughput of TCP CUBIC". In: *Global Telecommunications Conference (GLOBECOM)*. Florida, USA, Dec. 2010.

[13] S. Poojary and V. Sharma. "Analytical Model for Congestion Control and Throughput with TCP CUBIC Connections". In: *Global Telecommunications Conference (GLOBECOM)*. Texas, USA, Dec. 2011.

[14] J. Balej, O. Komnek, and M. Rajnoha. "Geographic Distance Estimation for IP Geolocation". In: *Proceedings in EIIC-The 2nd Electronic International Interdisciplinary Conference*. Sept. 2013.

[15] L. Q. Chen et al. "A Visual Attention Model for Adapting Images on Small Displays". In: *Multimedia Systems* 9.4 (2003), pp. 353–364.

[16] R. Mohan, J. R. Smith, and C. S Li. "Adapting Multimedia Internet Content for Universal Access". In: *Multimedia, IEEE Transactions on* 1.1 (1999), pp. 104–114.

[17] S. Hemminger. "Network Emulation With NetEm". In: *Australia's National Linux conference*. Canberra, Australia, Apr. 2005.

[18] ITU-T Recommendation G.114. "One-Way Transmission Time". In: *International Telecommunication Union* (May 2000).

[19] ETSI TR 101 329-6. *Actual Measurements of Network and Terminal Characteristics and Performance Parameters in TIPHON Networks and Their Influence on Voice Quality*. July 2001.

[20] V. Paxson et al. *Computing TCP's Retransmission Timer*. RFC 6298. June 2011.

[21] R. Braden and V. Jacobson. *TCP Extensions for Long-Delay Paths*. RFC 1072. Oct. 1988.

[22] J. Shaikh, M. Fiedler, and D. Collange. "Quality of Experience From user and Network Perspectives". In: *Annals of Telecommunications-Annales des Télécommunications* 65.1-2 (2010), pp. 47–57.