Analyzing the Impact of Delay and Packet Loss on Google Docs

Lam Dinh-Xuan[1(✉)], Christian Schwartz[1], Matthias Hirth[1], Florian Wamser[1], and Huong Truong Thu[2]

[1] University of Würzburg, Würzburg, Germany
lam.dinh-xuan@informatik.uni-wuerzburg.de
[2] Hanoi University of Science and Technology, Hanoi, Vietnam

Abstract. Software as a Service allows end users to use complex software directly from their browsers, transferring heavy computation to servers in the cloud. One use of this paradigm is word processing, former a classic use cases of Thin-Client computing. Similar to Thin-Clint systems, the network parameters are an important influence factor for the cloud application performance.

In this paper, we study Google Docs as an example for online word processing tools. We consider a traditional single user scenario as well as a collaborative scenario with two users working on one document simultaneously. We identify multiple relevant sub-processes per scenario as performance metrics and use a testbed to automatically evaluate the performance of Google Docs under varying network parameters. The main contributions of the paper are (1) the quantification of the impact of network parameters such as delay and packet loss on application performance metrics for both scenarios and (2) linear regression models to derive the application performance for giving network parameters.

Keywords: Google docs · Network parameter · QoS

1 Introduction

Cloud Computing and Software as a Service (SaaS) have received considerable interest by both the research as well as the industrial community. In recent time SaaS solutions have begun to absorb markets traditionally occupied by Thin Client products. One such SaaS application is *Google Docs*, an Internet based word processor. While a traditional desktop word processing application such as Microsoft Word provides a more complete feature set, Google Docs is a lightweight utility with sufficient office features and high flexibility. As an additional feature, Google Docs enables users to share created documents with other users or even collaboratively edit them.

However, as Internet-based cloud application, the performance of Google Docs depends on the network quality between server and client. Our study evaluates the performance of Google Docs with regard to different network conditions

R. Agüero et al. (Eds.): MONAMI 2015, LNICST 158, pp. 211–224, 2015.
DOI: 10.1007/978-3-319-26925-2_16

in two scenarios. First, a *single user scenario* is studied. In this scenario, a user has to take several sub-processes such as *Login* or *Typing*. We consider the time required to complete the sub-processes as a metric for the performance of the service. In the *collaborative scenario*, two users login to Google Docs. The first user edits a document while the other user observes the editing. Here, we consider the time both users require to complete the total process as well as all composite sub-processes as a measure of the application performance. To evaluate the influence of different network conditions on the processing time in both scenarios, we emulate various network delay conditions and packet loss settings in a local testbed.

Increased network delay or packet loss can cause an increase of the duration of the whole process or certain sub-processes on client. Therefore, in order to perform a better service, it is important to know which network characteristic influence the total processing time on the client. Furthermore, different sub-processes may be affected differently by network conditions and it is necessary to know which sub-process is most sensitive to delay or packet loss. Specifically, this paper answer the following research questions:

1. How do delay and packet loss influence the duration of sub-processes if a single user interacts with Google Docs?
2. How do delay and packet loss affect the duration of sub-processes in a collaborative scenario?
3. How does the combination of delay and packet loss influence the total processing time in collaborative tasks?

To answer those questions, we use a local testbed at the University of Würzburg to measure the duration of processes when users interact with Google Docs in both scenarios. We emulate one or two users which automatically perform workflows relevant to the scenario and record the required time. In this study, the duration of total process or each sub-process is the main criteria to evaluate the performance of Google Docs regarding different network conditions.

The contribution of this paper is threefold. First, we introduce a testbed for the measurement of SaaS applications w.r.t. varying network parameters. Second, we analyze the performance of the Google Docs application regarding the identified performance metrics. Finally, we provide a model used to derive Google Docs performance metrics given a set of network parameters and quantify the goodness of fit.

This paper is structured as follows. Section 2 presents the background of this study and related work. The testbed setup as well as the methodology is described in Sect. 3. Then, Sect. 4 discusses results gathered from the measurements. Finally, Sect. 5 concludes this work.

2 Background and Related Work

In this section, we introduce Google Docs as well as the two use cases considered in this paper. Thereafter, we present an overview of related work.

2.1 Background

Google Docs is a web based word processing application [1] whose client side front end is based on HTML and JavaScript and can be accessed using any modern web browser. In contrast to standalone office software products, e.g., Microsoft Office or LibreOffice, Google Docs requires a permanent Internet connection as documents are not stored locally on the client but on the Google server infrastructure. Google Docs does not provide rich feature sets like stand alone office products, however, it offers an easy way to share documents and enable collaboratively editing with up to 10 users simultaneously by sharing a link to the document or granting explicit rights to other registered users.

We consider two scenarios, which are derived from common Google Docs use cases. First, we discuss the single user scenario with one user editing a document. Here, a session is divided in five steps, which we will refer to as *sub-processes*. In a first step, the user logs into the system to gain access to a previously created document or to create a new document (*Login*). In the next step, the user creates a new document (*Creating*). Then, the user starts typing while the client continuously sends updates to the server to stores entered text at the server (*Typing*). After entering the text, it takes a shot amount of time to save the last changes to the text (*Saving*). The session is then ended with the logout of the user (*Logout*).

The durations Δt_{login}, $\Delta t_{\text{creating}}$, Δt_{typing}, Δt_{saving}, and Δt_{logout} of the five sub-processes *Login*, *Creating*, *Typing*, *Saving*, and *Logout*, as well as the total time of the session Δt_{total} are considered as an objective metric to asses the impact of network conditions on the quality of service for Google Docs. While it is intuitive that most of the aforementioned metrics depend on the network parameters, Emmert et al. [2] showed that the effective typing speed of an user also depends on network parameters in thin client environments. We will refer to this scenario as *single user scenario* in the remainder of this work.

As mentioned before, one of the major benefits of Google Docs is collaborative editing. In this case, the user session is more complex than in the single user case. For this scenario, we assume that *user 1* is creating the document and shares it with a collaboration partner *user 2*. Therefore, the work flows of *user 1* and *user 2* are almost similar to the work flow in the single user scenario. However, to share the document, *user 1* sends a link to *user 2* which grants him access rights to the newly created document. In this scenario, we additionally define two waiting times: (1) the duration *user 1* has to wait until *user 2* is ready to receive text, (2) the duration that *user 2* has to wait until *Receiving* starts. After *user 2* accessed the document, *user 1* starts writing and the content is automatically synchronized with *user 2* via Google Docs. As *user 2* is not actively editing the document, he does not observe a *Saving* phase.

2.2 Related Work

The impact of network conditions on Internet applications and remote desktop systems has be widely studied. However, to the best of our knowledge, the evaluation of quality of service for Google Docs has not been taken into account so far.

Schlosser et al. [3] analyzed the behaviour of Microsoft Word and Excel running in a remote desktop environment under different network conditions. They considered the Microsoft's Remote Desktop Protocol (RDP) and Citrix Presentation Server (CPS) as possible thin-client solutions. Their results showed that delay $\leq$ 500 ms or packet loss $\leq 2\,\%$ does not have any influence. However, the combination of delay and packet loss results in measurable impairments.

In [4], the authors focused on how Input Buffer and Speedscreen options can improve the performance of Citrix Presentation Server (CPS) in a WAN scenario. The author performed measurements with a user typing a text, scrolling a text, and selecting specific sub-menus on Microsoft Word and Textpad, respectively. The test duration under different network conditions was the main criteria to evaluate the performance of CPS. From the results, the author concluded that with the increasing of network delay up to 500 ms in combination with packet loss $\leq 2\,\%$, CPS with the combination of Speedscreen and Input Buffer took less time to finish the test than without these options.

In both studies, the applied methodology is similar to the one used in this work. However, we are not focusing on traditional thin-clients but rather study a web based solution. Further, we also consider a collaborative use case, which is not studied in previous publications. Other publications study cloud services and theirs network requirements. The authors of [5] focused on five fundamental challenges for wide adoption of cloud computing using the *OPTIMIS* toolkit. Amrehn et al. concluded that for general file storage services, the upload and download speed, financial aspects, privacy, and security are important QoE influence factors [6]. The authors of [7] used a prediction system to forecast the CPU demands for web based cloud services. Other studies evaluated the subjective user satisfaction, i.e. Quality of Experience, with cloud services [8,9].

These studies focus on different aspects of cloud computing. However, the authors did not evaluate a specific cloud application or investigate the impact of network conditions on the performance of cloud applications.

3 Methodology and Testbed Setup

We use a dedicated testbed including a network emulator to analyze the influence of different network parameters on the behaviour of Google Docs, allowing for an easy adaption of network parameters such as delay and packet loss. In the following we first detail on the test setup for the single user case in Sect. 3.1, thereafter we describe the setup for the collaborative scenario in Sect. 3.2.

3.1 Testbed Setup for Single User Scenario

The testbed setup for the single user scenario is schematically depicted in Fig. 1 and consists of one measurement server, one network emulator, and a control PC. The measurement server hosts the virtual machine VM1 used as Google Docs client for *user 1*. The virtual machine is connected to the Internet via another

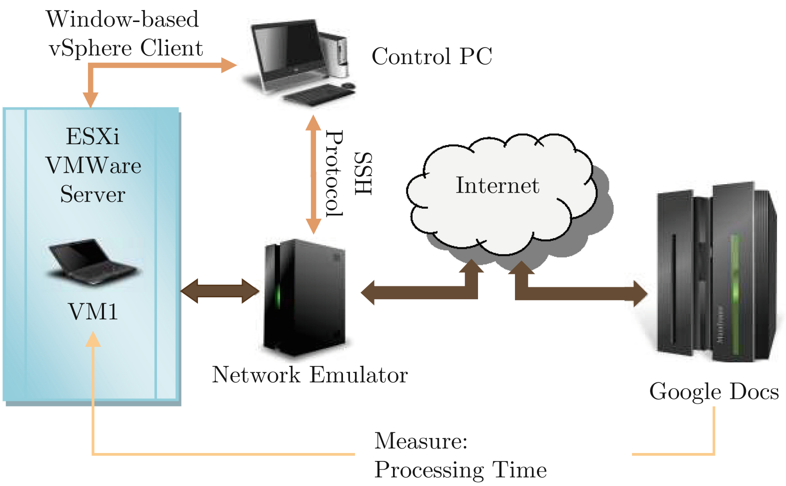


Fig. 1. Overview of testbed setup for the single user scenario

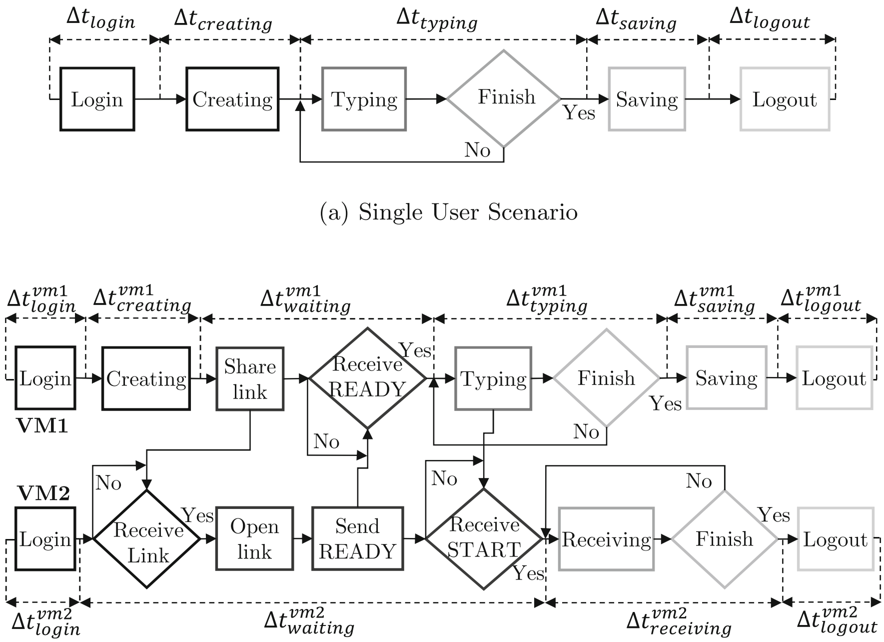
server running NetEm[1], which enables us to adjust packet loss and delay on the connection. To control the measurements, we use a control PC that is connected to the network emulator and the Google Docs client via a dedicated control network to avoid interference with the tests.

The measurement server and the network emulator are SUN FIRE X4150 servers with 8 CPUs 2.5 GHz, 16 Gb RAM, and 4 Ethernet 1 Gbps NICs. VMware ESXi 5.5[2] is used as virtualization solution and both the Google Docs client and the network emulator use Ubuntu 12.04 LTS as operation system. The testbed is connected to the Internet with a research network. We measure the baseline network parameters with a round trip time of 3.91 ms and no packet loss over 1000 packets. For later evaluation, we consider network delays from that baseline up to 1000 ms. Such high delay values can, e.g., occur due to long distance Internet access [10] or bottlenecks [11]. We consider packet loss from the baseline up to 4 % which may occur in a wireless link in urban area [12].

As discussed in Sect. 2, we assess the influence of the network parameters by measuring the duration of the sub-processes. To this end we use the Selenium Webdriver[3] to automatically generate user interactions. Figure 2(a) depicts the program flow of the measurement script and the recording of the time stamps used for measuring the duration of the sub-processes.

First the control PC sets up the network emulator with the desired configuration. Thereafter the Selenium script is started, which signs in to Google Docs and creates a new document. The content entered by the script is an English text taken from the introduction part of Selenium webpage. To evaluate the

1 http://www.linuxfoundation.org/collaborate/workgroups/networking/netem.
2 https://www.vmware.com.
3 http://www.seleniumhq.org.



(a) Single User Scenario

(b) Collaborative Scenario

Fig. 2. Measurement workflows

influence of the length of the text of the duration of the typing process, we use a short text of 1548 characters, which corresponds approximately on paragraph in a document. Besides this we also use a long text with about 6189 characters, which corresponds to about two pages of A4 document. After the automatic typing is complete, the Selenium script waits until the document is saved and logs out of the Google Docs. For each network parameter setting we produce 50 replications within several days to avoid measuring diurnal effects.

3.2 Testbed Setup for Collaborative Task Scenario

In the collaborative scenario we consider two users working on the same document, with one user editing the content of the document and the other user reading the document. To analyze this scenario we extend the testbed configuration described in Fig. 1 by adding another virtual machine (VM2) as *user 2* on the measurement server. In this scenario we require synchronized clocks for both client PCs. While this is challenging when using two different physical machines, it can be accomplished using two virtual machines sharing the host clock. Similar to VM1, VM2 is connected to the Internet via the network emulator, so that both VMs share the same network parameters. VM2 is also connected to the control PC using a dedicated control network. Additionally a second control network is established between the two virtual machines to synchronize the workflows of the machines as describe below. In the measurement, we use short sample text from the single user scenario and the same network settings.

Figure 2(b) shows the workflow in the collaborative scenario. The upper and the lower part of figure represents the processes on VM1 and VM2, respectively. The workflow for VM1 is similar to the one in the single user scenario. However, after creating the new document, VM1 shared the document with VM2 by sending a link. The workflow of VM2 differs in such a way that VM2 does not create a new document itself, but just waits for the link to the shared document. In order to synchronize the workflows of the two virtual machines, VM1 waits after sending the link to VM2, until VM2 places a marker in the shared document. Thereafter, VM1 starts tying in the document and VM2 observes the changes.

In addition to the times measured in the single user scenario, we also consider the waiting times of the two virtual machines in this case. This is for VM1 the time between creating the document and the notification from VM2 that it successfully accessed the shared document, and for VM2 the time between logging in and observing the first changed by VM1 in the shared document. Moreover, we also measure the time it takes until all changes on the document made by VM1 are visible on the document seen by VM2.

4 Results

Based on the measurement setup discussed in Sect. 3 we analyze the scenarios introduced in Sect. 2.1. We study the impact of network parameters, i.e. packet loss and delay, and text length on the single user and collaboration scenarios, with regard to the sub-process and total durations introduced earlier.

All measurements were performed between February 12, 2015 and March 24, 2015. For each parameter setting 50 repetitions of the measurement were performed, in order to increase statistical significance. The measurement settings are chosen according to the values discussed in Sect. 3. In order to avoid measuring diurnally effects, we did not perform measurements with the similar settings consecutively, but distributed them over different times of day.

In Sect. 4.1 we evaluate the effect of individual network parameter on the single user scenario. Then, in Sect. 4.2 we extend the study to the collaboration scenario. Finally, in Sect. 4.3, we consider the impact of the combination of delay and packet loss on the collaboration scenario.

4.1 Impact of Different Network Conditions on Sub-processes in Single User Measurements

We first investigate the single user scenario, as described in Sect. 3.1 and show the results in Fig. 3. For all figures, the y-axis gives the sub-process duration with 95 % confidence intervals in seconds. For sake of readabilities the y-axis is cropped to 30 s, but the measurement values are given in the figure description.

In Figs. 3(a) and (b), we study the impact of different network parameters for different text lengths. Here, the x-axis in the left sub figures shows the different

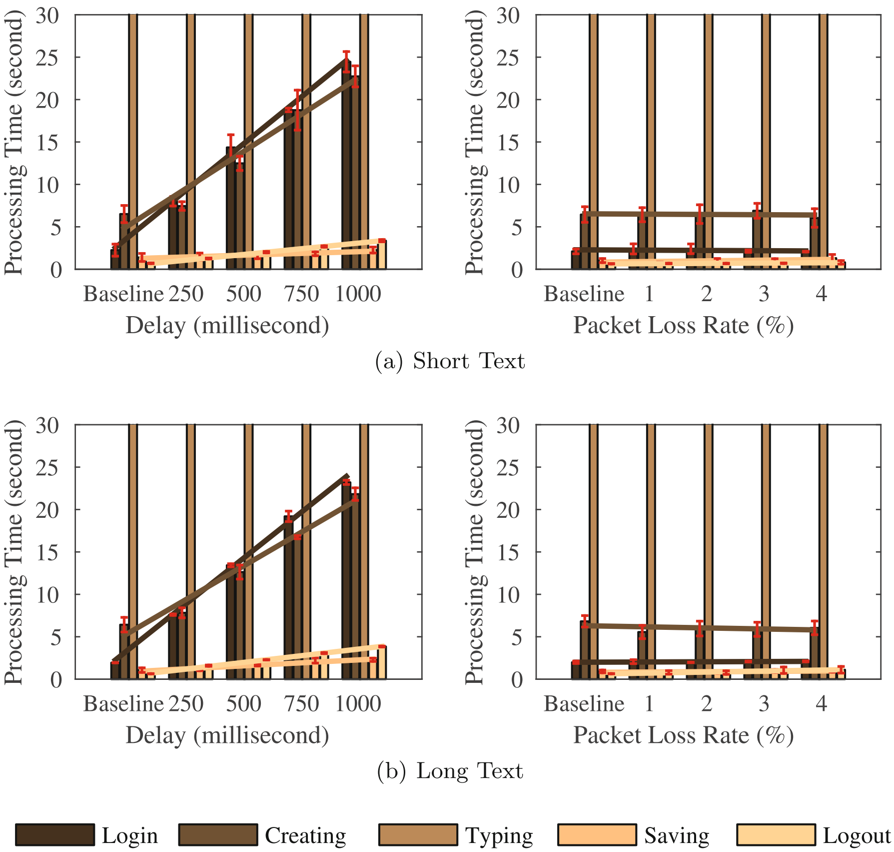


Fig. 3. Impact of network conditions on sub-process durations in single user scenario

delay settings in milliseconds, from the baseline unmodified delay, to an additional delay of 1000 ms in increments of 250 ms. The right sub figures show the impact of packet loss on the single user scenario. Here, the x-axis gives the additional induced packet-loss from the baseline setting without additional packet loss, up to 4 % in increments of 1 %.

All figures give sub-process durations as bars, colored depending on the sub-process type. Additionally, for each sub-process a linear regression is performed, which is shown as a colored line, depending on the sub-process type. Table 1 shows the detailed results of the linear regression depending on the delay, including the coefficient of determination r^2 as a measure for the goodness of fit. Figures 3(a) and (b) show the packet loss in the considered range up to 4% does not affect the processing time for any sub-process. However, the increase of network delay results in increased processing times for all sub-processes except the *Typing* time, with *Login* and *Creating* document being the most sensitive to network delays. When delay increases from baseline to 1000 ms, the *Typing* time remains

Table 1. Linear Regression of Sub-processes for Delays in Single User Scenario

Sub-processes	Short-text Measurement	r^2	Long-text Measurement	r^2
Login	$22.07 \times 10^{-3} \cdot x + 2.54$	0.99	$21.64 \times 10^{-3} \cdot x + 2.26$	0.99
Creating	$17.50 \times 10^{-3} \cdot x + 4.83$	0.96	$15.89 \times 10^{-3} \cdot x + 5.14$	0.94
Typing	$3.70 \times 10^{-3} \cdot x + 56.71$	0.49	$-4.95 \times 10^{-3} \cdot x + 247.35$	0.32
Saving	$0.84 \times 10^{-3} \cdot x + 1.29$	0.87	$1.41 \times 10^{-3} \cdot x + 0.95$	0.92
Logout	$2.70 \times 10^{-3} \cdot x + 0.65$	0.99	$3.16 \times 10^{-3} \cdot x + 0.71$	0.99

almost constant at 60 s and 247 s for the short and the long text, respectively. This is due to the fact that updates to the server are sent asynchronously and the typing process does not depend on the reply of the server. Particularly, the duration of the *Login* process is about 7 times longer than for a delay of 500 ms then for the baseline measurement and 12 times longer at a delay of 1000 ms. The duration of the *Creating* document process doubles and almost triples for the corresponding delay values in comparison to baseline measurement. This is due to the fact that the *Login* and *Creating* sub-processes rely on multiple communications between client and server which are executed in serial order. In contrast to this, the *Saving* time only slightly increases and the *Logout* time takes approximately 3.3s at 1000 ms delay compared to 0.60 s at baseline delay. Due to the synchronization of the typed text in a background process, the saving of a document relies only on few communications with the server and thus is not influenced by a large measure. In the measurement of the long text as shown in Fig. 3(b), the behaviour of the *Login*, *Creating*, *Saving*, and *Logout* sub-processes is similar to the behaviour observed for the short text. Table 1 summarizes the results of the linear regression for the sub-processes, both for short text and long text measurements for a given delay x. Packet loss is not considered, as the impact on sub-process duration is negligible in this scenario.

We observe that increasing delay results in a large increase of the *Login* time and *Creating* time, while the effect on the other sub-processes is less significant. The linear function of *Typing* time has small slope coefficient of 3.70 compared to its intercept of 56.71. Therefore, the length of text is primary factor changing the *Typing* time, not the network delay or the packet loss. The durations of the *Typing* sub-process vary non-lineary with increasing delay, resuling in a bad fit and a low r^2 value.

Our measurements show the in the single user scenario, Google Docs is robust against packet loss. However, delay affects the system negatively, especially during processes depending on multiple serial communication between client and server, e.g. login or while creating new documents. The actual typing process, which represents interaction between the user and the client, is insensitive to the network conditions, as it is basically a background process, which does not affect the user directly. The measurements show that the duration of the typing process is mainly depending on the length of the text.

4.2 Impact of Different Network Conditions on Sub-processes in Collaborative Task

We now analyze the influence of different network conditions on the sub-processes in the collaborative scenario. As described in Sect. 3.2, we use two virtual machines. The document is edited by VM1 while VM2 observes the creation process. In Fig. 4, the y-axis shows the duration of each sub-process in seconds, while the x-axis in the left figures shows the network delay and the packet loss rates in the right figures. Bars are colored by sub-process time and give the mean and 95 % confidence interval of duration, the lines show the linear regression.

Similar to the results from Sect. 4.1, Figs. 4(a) and (b) indicate that a packet loss of less then 4 % has no influence on the observed sub-process durations. Increasing delay results in an increasing duration of almost all sub-processes in both VM1 and VM2, with *Login*, *Creating* document and *Waiting* being the most sensitive processes. In contrast to this, the *Typing* time on VM1 and *Receiving* time in VM2 are only slightly fluctuating around 60 s even for higher delays. Again, this is due to the fact that synchronization between both VMs occurs asynchronously and does not depend on the responses of the server. As expected, the *Login* times are similar for both machines, because both experience the same network parameters. Furthermore, we observe that the *Waiting* time for VM2 is approximately the sum of the *Creating* and *Waiting* time of VM1. This can be explained, by the fact that both machines start with the login process at about the same time but VM1 has to create the document first. Thereafter starts the synchronization process for both workflows, c.f. Figure 2(b), which ends with the start of the *Typing* process on VM1 and the start of the *Receiving* process on VM2. These process again mark the end of the waiting periods of both machines. Interestingly, the *Typing* and *Receiving* process take about the same amount of time on both machines, independent of the network conditions. Again the parameters for the linear regression models are summarized in Table 2. The missing values in the table indicate the sub-processes does not occur on the corresponding virtual machine.

Considering the r^2, we again observe that the *Login*, *Logout*, *Creating*, and *Saving* sub-process can be fit using a linear model, in contrast to the *Typing* and *Receiving* which do not show linear behaviour regarding the considered parameters. For the *Typing* and *Receiving* process the intercept is again much larger then the slope coefficient, which indicates that the network parameters again have only little influence on the sub-process durations. Again, we do not provide a linear regression concerning packet loss due to the negligible impact of the variable.

Our measurements show the in the collaborative scenario, Google Docs behaves similar to the single user case. It is rather robust against packet loss and more sensitive to delay. Processes depending on repeated communication between client and server are more affected by additional delay, then e.g., the typing process which uses an asynchronous communication pattern.

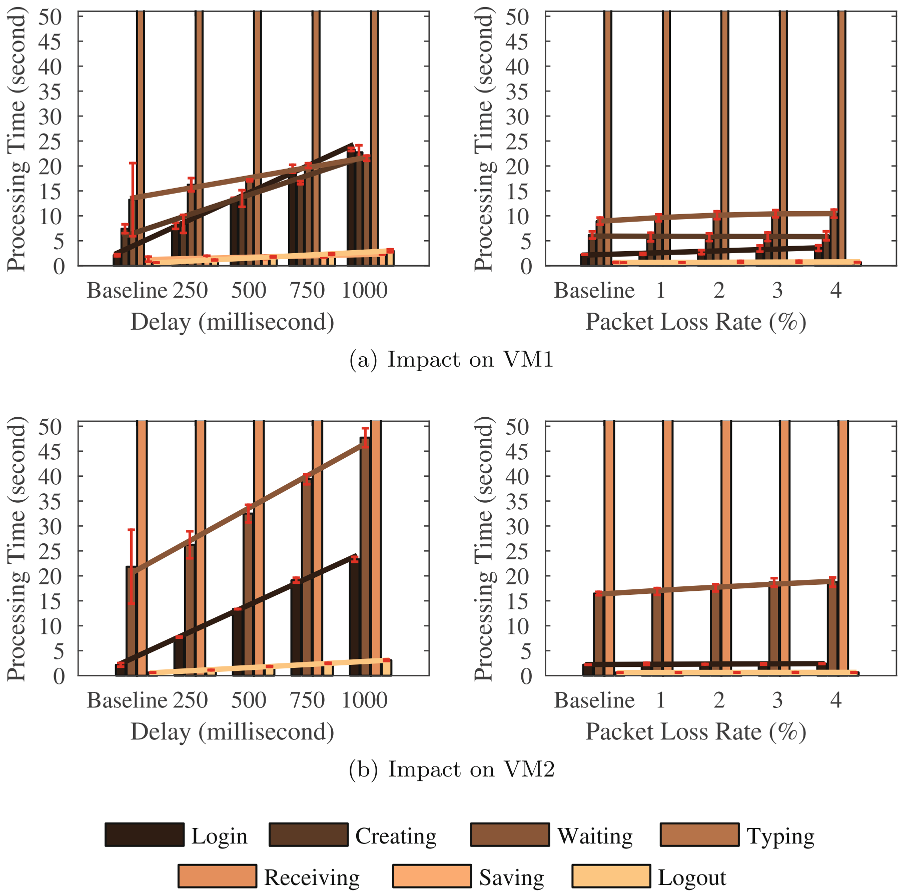


Fig. 4. Impact of network conditions on sub-process durations in collaborative scenario

Table 2. Linear Regression of Sub-processes for Delay in Collaborative Scenario

Sub-processes	VM1	r^2	VM2	r^2
Login	$21.53 \times 10^{-3} \cdot x + 2.51$	0.99	$21.47 \times 10^{-3} \cdot x + 2.41$	0.99
Logout	$2.39 \times 10^{-3} \cdot x + 0.59$	0.99	$2.50 \times 10^{-3} \cdot x + 0.58$	0.99
Creating	$15.56 \times 10^{-3} \cdot x + 5.95$	0.96	-	-
Typing	$-1.73 \times 10^{-3} \cdot x + 59.68$	0.29	-	-
Saving	$1.07 \times 10^{-3} \cdot x + 1.24$	0.83	-	-
Waiting	$8.16 \times 10^{-3} \cdot x + 13.57$	0.97	$25.94 \times 10^{-3} \cdot x + 20.56$	0.98
Receiving	-	-	$-2.81 \times 10^{-3} \cdot x + 60.08$	0.49

4.3 Impact of Combined Delay and Packet Loss on Total Processing in Collaborative Task

After analyzing delay and packet loss separately, we now consider the total process duration given packet loss and delay occurring at the same time. We consider total processing time Δt_{total} required for inputting the short text in the collaborative scenario. The results for the measurement show that the differences of $\Delta t_{\text{total}}^{\text{vm1}}$ and $\Delta t_{\text{total}}^{\text{vm2}}$ are negligible. Therefore we focus our discussion on obtained values for $\Delta t_{\text{total}}^{\text{vm1}}$ depicted in Fig. 5.

In Fig. 5, the y-axis shows $\Delta t_{\text{total}} = \Delta t_{\text{total}}^{\text{vm1}}$. The x-axis shows the different packet loss values from baseline to 4 %, different delay values are shows as grouped bars, including the 95 % confidence intervals for each measurement setting. We show a examples of a more general linear regression parameterized for the measurement parameters as lines colored according to the specific delay.

For the baseline packet loss and the considered delay values we observer that Δt_{total} increases, as discussed in the previous sections. We also see, that Δt_{total} is almost independent of the packet loss as long as the delay is small, i.e., at the baseline. This is intuitive, because in this case retransmission of lost packets can be considered as almost instantaneous and does not affect the transmission at all. However, in case of larger delays, the impact of packet loss starts to increase, as retransmissions take longer and consequently the time until information is successfully transmitted between server and client increases, as well.

This can be modeled using a linear regression with a r^2 value of 0.943 as

$$\Delta t_{\text{total}} = 62.247 + 0.077962 \cdot x + 809.54 \cdot y$$

for a delay x and a packet loss of y. Comparing the predicted results of the model with our measurements, we observe that while fluctuations up to 10 % occur at the bounds of our parameter set, the results are of sufficient quality to be used in general cases.

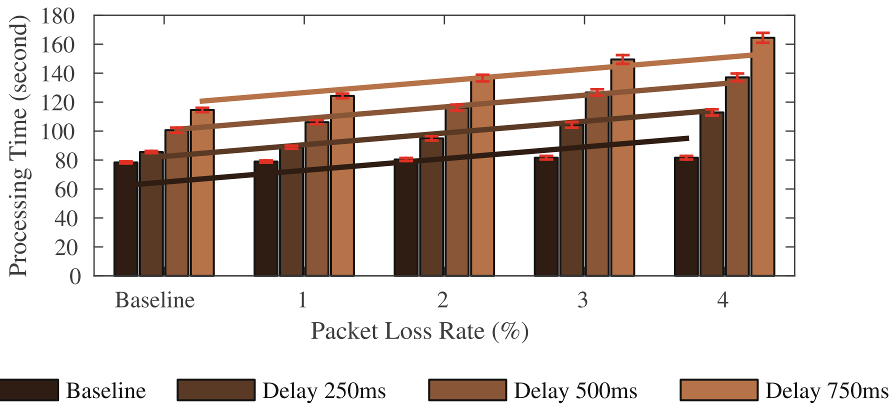


Fig. 5. Impact of combined delay and packet loss on the total duration of the collaborative task

These results show, that while packet loss alone has no significant impact on application performance, a combination of both packet loss and delay can negatively impact application behaviour. In real world scenarios, especially if WiFi or cellular access is concerned, both network parameters can be degraded noticeably. However, results from Sects. 4.1 and 4.2 show, that the impact can be mitigated by using asynchronous communication patterns between client and server.

5 Conclusion

Cloud services and SaaS products gained considerable interest recent times as a replacement for traditional centralized infrastructure and locally installed software products. Despite many advantages varying network conditions can significantly affect their performance. To assess the impact of network impairment, we analyzed the impact on a representative SaaS product, Google Docs.

In the first scenario, we considered a single user editing a Document, in the second one a collaborative use case, with one user editing the document and a second user observing the changes. To quantify the impact of network delay and packet loss objectively, we measure the time it takes to complete the whole process as well as certain parts of it, e.g., the login or the creation of the document. The measurements were performed using a local testbed which is connected to the Internet via a NetEM network emulator. The user interactions were automated using Selenium.

Our results show that in both scenarios, packet loss below 4 % does not influence the duration of sub-processes, if there is no network delay. In contrast to this, network delay negatively influenced the performance of Google Docs, even in the absence of packet loss. Hereby, the login process as well as the process of creating a new document are the most delay-sensitive sub-processes. Furthermore, we also analyzed the impact of combined delay and packet loss. Here the results show a significant degradation of the Google Docs performance even for small values of delay and packet loss, if both occur at the same time.

The results from this study can help to shed a first light of the behaviour of Google Docs as an exemplary SaaS solution under varying network conditions from an objective point of new. This in turn can later be used to evaluate the impact of this application behaviour on the perceived quality of the end-user (QoE). Here especially the obtained linear regression models can be used in analytical models for optimization and trade-off analysis network resources, energy consumption and QoE. In a next step, these model can be extend to other Google cloud services such as Google Spread Sheet and Presentation.

Acknowledgment. This work is supported by the Deutsche Forschungsgemeinschaft under Grants HO TR 257/41-1 “Trade-offs between QoE and Energy Efficiency in Data Centers”.

References

1. Dan, H.: Google this!: using google apps for collaboration and productivity. In: SIGUCCS Fall Conference. St. Louis, USA, October 2009
2. Emmert, B., Binzenhöfer, A., Schlosser, D., Weiß, M.: Source traffic characterization for thin client based office applications. In: Pras, A., van Sinderen, M. (eds.) EUNICE 2007. LNCS, vol. 4606, pp. 86–94. Springer, Heidelberg (2007)
3. Daniel, S., Andreas, B., Barbara, S.: Performance comparison of windows-based thin-client architectures. In: ATNAC 2007. Christchurch, New Zealand, December 2007
4. Daniel, S., et al.: Improving the QoE of Citrix Thin Client Users. In: International Conference on Communications. South Africa, May 2010
5. Ferrer, A., et al.: OPTIMIS: a holistic approach to cloud service provisioning. In: Future Generation Computer Systems, vol. 28 (2012)
6. Amrehn, P., et al.: Need for speed? on quality of experience for file storage services. In: Workshop on Perceptual Quality of Systems. Vienna, Austria, September 2013
7. Reig, G., Guitart, J.: On the anticipation of resource demands to fulfill the QoS of SaaS web applications. In: Conference on Grid Computing. Beijing, China, September 2012
8. Hoßfeld, T., et al.: Challenges of QoE management for cloud applications. IEEE Commun. Mag. **50**(4), 28–36 (2012)
9. Jarschel, M., et al.: An evaluation of QoE in cloud gaming based on subjective tests. In: Innovative Mobile and Internet Services in Ubiqui-tous Computing (IMIS), pp. 330–335. IEEE (2011)
10. O3B Networks. What is Network Latency and Why Does It Matter? White Paper. http://tinyurl.com/nv8agu8. 2008
11. Ningning, H., et al.: A measurement study of internet bottlenecks. In: INFOCOM. Miami, Florida, USA (2005)
12. Anmol, S., et al.: Packet loss characterization in wifi-based long distance networks. In: International Conference on Computer Communications. Anchorage, Alaska, USA (2007)