Improving the QoE of Citrix Thin Client Users

Daniel Schlosser, Barbara Staehle
University of Würzburg, Institute of Computer Science
Würzburg, Germany
{schlosser,bstaehle}@informatik.uni-wuerzburg.de

Andreas Binzenhöfer, Björn Boder
DATEV eG
Nuremberg, Germany
{andreas.binzenhoefer,bjoern.boder}@datev.de

***Abstract*—During the last years so-called "Thin Client" architectures have become very popular. Thin Clients are simple end user devices, which are provided with content and services by powerful servers. Originally this concept was developed for Local Area Networks (LANs), which provide a high Quality of Service. If they are used in Wide Area Networks (WANs) they need to be adjusted accordingly in order to guarantee a good Quality of Experience (QoE). In this paper we investigate possibilities for configuring Citrix based Thin Client architectures to improve the QoE in WAN environments. We consider the benefits on the QoE as well as the costs on network layer.**

I. Introduction

Today, the predominant office computer environment is mostly based on the traditional personal computer architecture where each employee has his own PC, which runs all applications he needs for his daily work. Software and data is kept locally making maintenance error-prone and expensive. Thin Client architectures are believed to offer an answer to these problems. In a Thin Client environment the user device is merely a simple terminal for user interaction. Furthermore, the user device does neither need excessive memory nor a powerful CPU, which is why it is called a "Thin Client". Applications are run remotely on a server farm, which provides the required computational power and also stores all user data. This architecture has the advantage, that the administration and maintenance of the system can be done centrally.

The low hardware requirements enable the use of small and lightweight devices with low energy consumption in order to access a Thin Client system. All the device needs is a network connection to the server. Thereby, the bandwidth consumption is not very demanding as we have shown in [1]. Thus even wireless access services like e.g. UMTS are able to provide sufficient bandwidth.

As we have shown in [2] a small change in the Quality of Service (QoS), e.g. an increase of delay as it might occur in cellular networks, can affect the QoE of the user dramatically. Although this might be tolerable for private usage during short periods of time, these effects are undesirable in a productive environment. The administrator therefore needs to adjust the Thin Client system in order to optimize the Quality of Experience perceived by the users.

In this paper we focus on a Citrix based Thin Client architecture, as it is widespread in professional environments and offers special settings to improve the QoE in low QoS environments. We examine how these settings affect the network, how far the QoE of the user can be improved in a network environment with low QoS, and explain how these results can be used to optimize a Thin Client system.

The remainder of this paper is organized as follows. Section II gives a brief overview of related work. The measurement setup and methodology are explained in Section III. We discuss the results of our measurements and their implications in Section IV, while Section V concludes this work.

II. Related Work

In the area of distributed system research and Thin Client architectures in particular the Quality of Service can be described by parameters like packet loss, end-to-end latency or jitter, which are easy to measure. Quality of Experience of the end user, which expresses to what amount a user is satisfied with the results of a service, is harder to capture, as it is a subjective measure. Thus, it is not only based on the performance of the service itself but also on the individual expectations and perceptions of the user.

In the field of multi-media transmissions like VoIP and IPTV, significant progresses on the problem of mapping QoS to QoE have been made during the last years. For example [3] reports an exponential relationship between QoS and QoE in terms of the *Mean Opinion Score* (MOS) [4] for all edge-based multimedia services.

In contrast, research on Thin Client computing has focused on typical traffic characteristics and neglected the user perceived service quality. In [1], the traffic caused by Thin Client based office applications is characterized. In [5] the differences between several Thin Client architectures are analyzed. In [6] the response time of text editing, presentation creating, and image processing applications accessed via VNC was measured. For this purpose, all response times are classified to be between "crisp", if smaller than 150 ms and "unusable", if over 5 s. Based on this simple QoE measure, it is shown that the performance of highly interactive applications is more sensitive to network delays than that of simpler applications.

However, none of these studies deals with the problem, which is of key interest to application service providers: How can the QoE of an end user be improved under a given network QoS? To some extend Citrix Inc. and other companies selling network solutions are already advising their costumers on how to optimize their working setup. One example can be found in [7], where the optimal settings for the Juniper WX/WXC platform and the Citrix client and server for low bandwidth conditions are described . A Citrix white paper [8] addresses the problem of using the MetaFrame presentation server in a wireless WAN. However, all previous works only roughly explain *why* a certain parameter value should be chosen, while no details are given on the qualitative influence.

Another study [5] comparing the performance of different Thin Client solutions when faced with slow motion (i.e. using a text editor and browsing) and high motion (watching a video, playing a 3D game) tasks illustrates, that all tested protocols are well suited for low motion applications. The study showed that in high-motion scenarios, however, not all clients are able to send the graphical data fast enough to the user. Citrix is suitable for this task and outperforms all other protocols in terms of bandwidth and used server CPU, if the Speedscreen Multimedia Acceleration is activated (cf. [11] and Section III-B3 for details).

Our paper is self-contained, but it can also be seen in the context of our previous work, which has been dedicated to the interaction of Thin Client systems and WANs. [1] characterizes the network behavior of Thin Client Systems. In [2] we presented a sophisticated concept which allows mapping given network QoS to user satisfaction by measuring the duration of pre-defined tests and analyzing the distribution of the test durations. Our experiments with test persons showed, that this "objective" and easy to assess QoE measure captures the subjective user satisfaction to a good degree. The comparison of different Thin Client systems regarding the QoE in low QoS networks is presented in [12]. We showed, that Citrix based Thin Client systems may outperform RDP and VNC based systems in high QoS networks, but only reveal a declined QoE under low QoS conditions without any adjustments.

In this paper we focus how Citrix based Thin Client Systems might be tuned to these situations by adjusting built in parameters in order to optimize the Quality of Experience perceived by the user.

III. Measurement Setup and Methodology

In the following, we describe the methods used to conduct our study. For this purpose, we first briefly introduce our test bed and how we measure the QoE of a Thin Client user. Thereafter we describe the Citrix parameters that are considered in our study.

A. Testbed Environment and Performance Metrics

In order to emulate a typical Thin Client architecture used in productive environments, we set up a testbed as depicted in Figure 1. For the server side we use two 3.4 GHz Intel Xeon servers with 3.5 GB RAM each, running on Windows 2003 Server standard edition with Service Pack 1. The Windows Terminal Server in Figure 1 is responsible for hosting the server side applications of the tested terminal services. The second server is set up as a file server and used to store user data. In order to emulate varying network conditions we use a NistNet [13] machine in the middle of the communication channel. For this task we use a Dual Pentium III 500Mhz computer with 512 MB RAM running OpenSuSE 10.0 and NistNet 2.1012.c. Note that both the processing power as well as the memory of this machine are well above the minimum requirements for NistNet. On the client side we use a Pentium IV 2.6 GHz machine with 1 GB RAM running Windows XP with Service Pack 2. All hosts are connected using 100 Mbit. We dimensioned the hosts and the network in such a way, that none of these components is a bottleneck and the performance of the applications is only affected by the used terminal service solution as well as by the emulated network conditions.

For this work we focus on network parameters round-trip-time (RTT) and packet loss. The RTT is emulated with NistNet in a constant manner, equally split to both directions. The packet loss emulated is uniformly distributed and applied to both directions.

We used Microsoft Word and Textpad as typical office applications. Both were used to compare the performance of the Citrix Metaframe Presentation Server 4.0 in combination

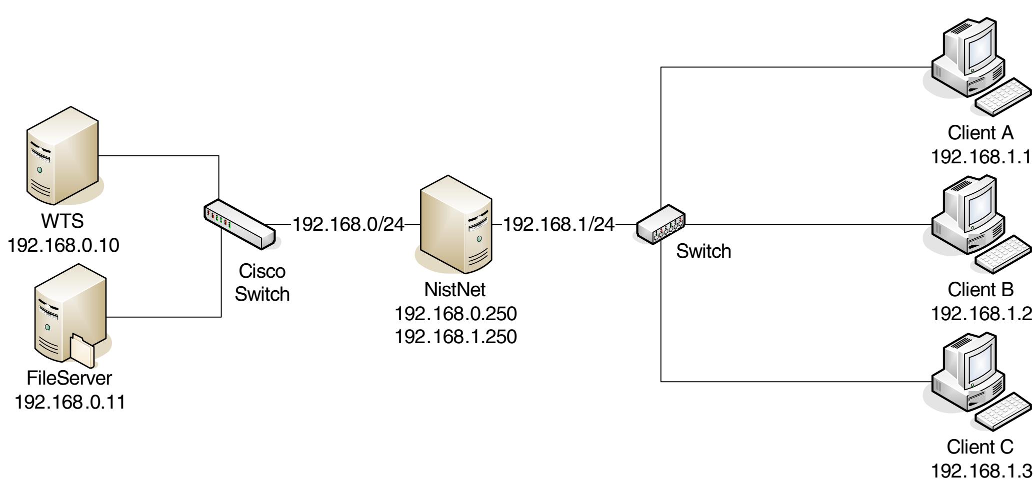

Fig. 1. Overview of the measurement testbed

with the ICA Client in version 9.237. Data compression and session reliability were enabled and the color depth was set to 16 bit. Note that the ICA Protocol is based on TCP. Thus, a lost packet will be resent by the TCP fast retransmit mechanism. Hence, the case of a timeout is very unlikely as Citrix transmits many small packets with a short intersent times, c.f. [1]. However, on application layer a lost packet will be noticed as a large gap in the information flow followed by a bulk arrival at the end.

We approach the problem of measuring the QoE by quantifying the time required to complete typical office tasks on application layer. As shown in [2] this is suitable as a QoE metric. The tasks we use for our experiments are typing a text, scrolling this text, and selecting specific submenus.

In order to make this task repeatable, we automate the user interaction with the open-source tool AutoHotkey [14], which is able to carry out keystrokes as well as mouse movements and clicks. In order to emulate a real user in front of the screen, we check the output on the screen after some input has been passed to the application. Hence, our script waits for the correct visual response on the screen after a given input before it continues to perform the next task. For example if the script emulates some keystrokes it waits until the corresponding characters appear on the screen before it continues typing the next letters. Under optimal network conditions, i.e. without delay or any other network disturbances, the measured task duration is deterministic. However, for a congested network this is no longer true. Therefore we repeat each task continuously for one hour, to get enough measurements for a credible statistical analysis.

B. Investigated Citrix Options

With the increasing popularity of terminal services, Citrix Inc. integrated options in their products to cope with poor network conditions. In this paper we take a look at the most important options introduced for WAN optimizations: Color Depth, Input Buffer settings, and the Speedscreen (Latency Reduction) option. They all have a different purpose and special features on which we will detail in the following.

1) Color Depth: We first investigate the option to configure the color depth for the visualization of the remote application. The color depth can be adjusted to 8 bit, 16 bit, and 24 bit. Changing this parameter will obviously alter the visualization on the client screen and should therefore also modify the amount of data the server has to send to the client. We expected that with fewer colors the bandwidth is reduced. However, the measurements show that neither on the network level nor on the application level any effect was statistically significant. The used bandwidths and the task duration times stay the same for all color depth settings. Only the colors on the screen changed. The compression of visual data before sending it over the network is obviously able to level the differences in the color information. It is therefore reasonable to always use 24 bit color depth. Hence, we focus on the other features in the following.

2) User Input Buffer : As we have shown in [12], a Citrix client tries to maximize the responsiveness by sending all user input to the server as fast as possible. With the Input Buffer option it is possible to make the Citrix client collect user input information at the client and send the information after some predefined intervals, which can be independently adjusted for mouse and keyboard input. By default these values are 50 ms for mouse input and 100 ms for keyboard activities.

3) Speedscreen Latency Reduction : The Citrix Speedscreen Latency Reduction (or shortly Speedscreen) is promoted to be a very powerful tool for improving the QoE. This technique anticipates the server response at the client. Without Speedscreen, the client acts as a dumb terminal and merely displays the pixels sent by the server. Enabling Speedscreen changes mainly two things. First the client supports higher interaction by sending packets more frequently to the server. Second this option lets the client ”guess” the servers response. If the users types a character on the keyboard, the client will show the character on the screen even though the servers response has not arrived. While this sounds great in theory, there are limitations to what is possible in practice. For example in MS Word the typed character is shown on the screen, but often in a different font. After the feedback of the server was received, the font printed on the screen changed to the intended one. Our experience shows that the rendering of an incorrect font on the screen is acceptable for the user as long as the correct letter is shown.

4) Combining Speedscreen and Input Buffer : From our earlier descriptions it becomes clear, that the last two mechanisms have contradicting influences, which might neutralize each other: While the Input Buffer is supposed to *decrease* the consumed bandwidth and the QoE, the Speedscreen Latency reduction leads to an *increase* of the bandwidth and the QoE. We therefore combine these two mechanisms and examine whether it is possible to reduce the bandwidth consumption while improving the user experience or if a combination of the two options leads to a smaller QoE and uses more bandwidth compared to the default Citrix option set.

IV. Measurement Results

In this section we report on the most interesting results of the Citrix settings, which we observed in our test bed environment. In particular, we consider the Input Buffer, the Speedscreen Latency Reduction, and a combination of both and analyze the performance of these settings. As costs and QoE metrics we use the consumed bandwidth on the network layer and the test completion time, as explained in Section III-A. Our aim is to analyze the general behavior of the considered features as well as what effect they have under different network conditions.

A. Input Buffer

The Input Buffer directly influences the frequency at which information is sent from the client to the terminal server. Therefore when using this option it should be possible to reduce the bandwidth consumed by the client in upload

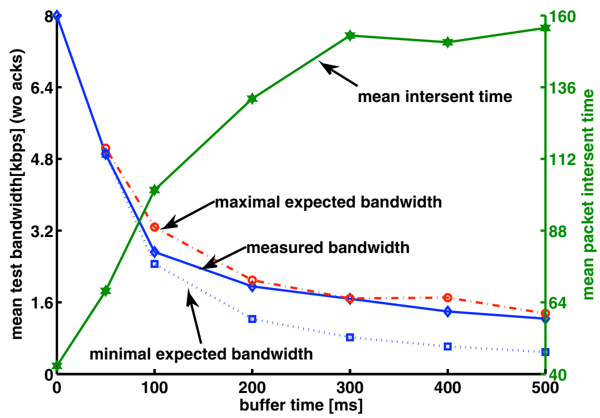

Fig. 2. Influence of increasing input buffer settings

direction, as e.g. mouse movements can be subsumed. In order to get a better understanding of how far this feature actually influences the bandwidth sent by the client, we first set both mouse and keyboard buffer to the same value. Then we increase this value progressively and study the effects on traffic sent by the client.

In order to illustrate the benefits and the costs we plot the latency introduced and bandwidth saved by the Input Buffer in Figure 2. The graph for the intersent times depicts the mean bandwidth usage during one hour tests of each Input Buffer Setting using MS Word. For settings smaller than 100 ms the mean intersent time changes according to the preset value for the Input Buffer. Thus, the used bandwidth drops rapidly. Note, that for a disabled Input Buffer the mean intersent time is about 40 ms. For higher presets of the Input Buffer time the measured intersent time does not increase accordingly. Instead the mean intersent times of the packets converges against 160 ms. This seems to be an upper bound, which the client does not exceed. The mean bandwidth during each test decays for increasing Input Buffer times as expected. To get a better understanding of the traffic reduction we additionally visualized the minimal expected bandwidth. We calculate these values from the bandwidth usage of the default value by dividing it through the relative increase of the buffer setting. If we e.g. double the Input Buffer setting, the minimal expected bandwidth is reduced by half. Furthermore the plot for the maximal expected bandwidth represents the bandwidth usage derived by dividing the bandwidth without the Input Buffer by the increase of the mean intersent times as measured during the tests. Comparing these plots we see that the bandwidth usage is not as low as it would be expected from the preset value of the Input Buffer times. However, the bandwidth gain is better than we would expect from the measured intersent times in most of the cases, e.g. for an Input Buffer time setting of 100 ms.

It has to be noted that the traffic sent from the server to the client also decays for an increased Input Buffer setting. This is caused by the fact, that in the default case, i.e. with disabled Input Buffer, about 80% of the packets sent by the server are TCP acknowledgements, cf. [1]. Hence, as the input updates are sent less frequently, the number of acknowledgements per timeframe decreases accordingly.

Figure 3 compares the influence of enabling the Input Buffer with Citrix default settings on the QoE perceived by the user. We show the median of the test durations for increasing round trip times (RTT), while the error bars visualize the inter quartile range of the test completion times. The two solid plots at the bottom depict the QoE measured for a Citrix client with disabled and enabled Input Buffer in a lossless network. The plot for the client with the activated Input Buffer reveals a slight increase of the test completion time. This is caused by the small delays occurring for each packet in the Input Buffer, which accumulate to a measurable increase of the time it takes to perform the test tasks.

The QoE measurements in an environment, where the network randomly drops packets, reveal different effects. The upper two dashed plots in Figure 3 depict the measurements made for 1% packet loss in the network. In this case, the plot for the activated Input Buffer is below the one for the task duration times of a Citrix client without the Input Buffer, i.e. the Input Buffer improves the QoE in case the network drops packets. Recall that under TCP, whenever a data packet is lost in the network the receiver will have to wait for at least two later packets to arrive in order to trigger the fast retransmission mechanism. The more packets are sent (no buffer) within a given period of time, the more packets could be lost causing the user to wait for retransmissions, thus slowing down his workflow. Hence, in the presence of packet loss in the network it is better to gather user input and transmit with less packets.

B. Speedscreen Latency Reduction

As mentioned before the Speedscreen Latency Reduction is a technique, which is supposed to affect the rendering of the

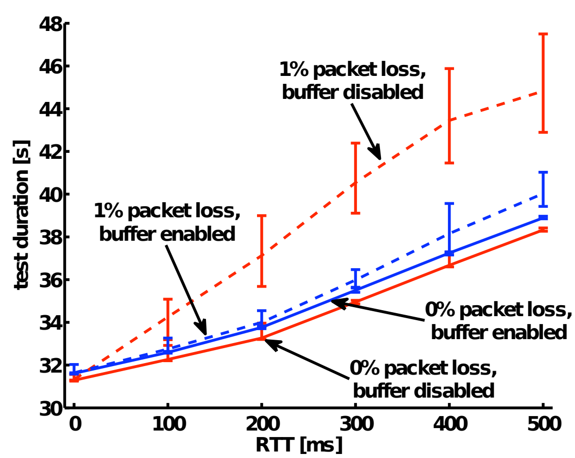

Fig. 3. Influence on Input Buffer on QoE

application on the client. This shall improve the responsiveness of the system by the costs of an increased bandwidth used by the client in upload direction.

The histograms in Figure 4 show how the Speedscreen option increases the used bandwidth. About 50% of the packet sizes sent by the client are three bytes larger, if Speedscreen is enabled. However the distribution of intersent times does not change much. Together this leads to a mean bandwidth increase of about 1 kbps between the client and the server (c.f. also Figure 6).

Figure 5 depicts the influence of Speedscreen on the QoE for different network scenarios. The two plots at the bottom represent the task duration measured with and without the Speedscreen option for an increasing delay in a lossless network. Under perfect network conditions, i.e. no delay and no packet loss, the performance of both Citrix clients is the same. With increasing round-trip times the tests are completed faster with the Speedscreen enhanced Citrix client than for the client without Speedscreen. Thus, the Speedscreen option can improve the responsiveness of the application by anticipating the server response. However with round-trip times beyond 200 ms the test duration with Speedscreen also increases. This depicts the evident fact that even with Speedscreen enabled the client is not independent from the server. From time to time it has to wait for the server response and thus the test completion time increases. The upper two dashed plots in Figure 5 reveal that the Speedscreen option will improve the task completion times even in situations with packet loss.

In summary, Speedscreen can improve the responsiveness by the cost of higher network load, which might be undesirable if the link is already overloaded.

C. Combining the Input Buffer and Speedscreen

The Input Buffer and the Speedscreen option can each increase the user perceived QoE. Thus, the question arises if a combination of both can also improve the QoE and how this combination affects the network level.

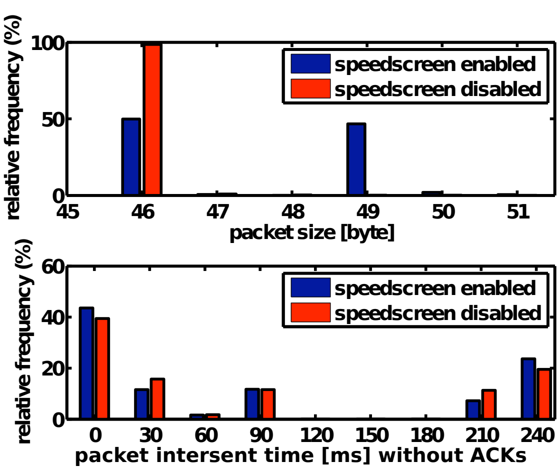

Fig. 4. Packet pattern sent by the client enabling Speedscreen

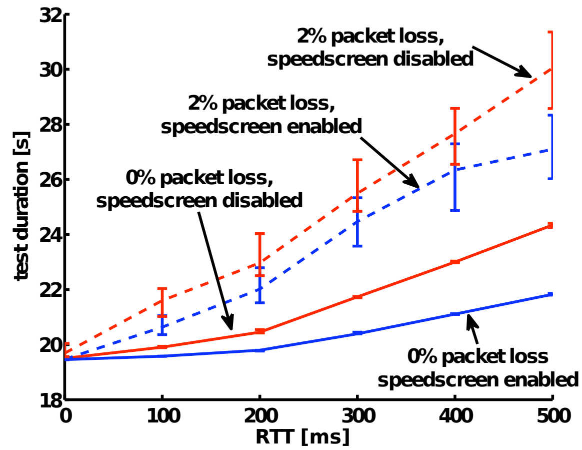

Fig. 5. Influence of Speedscreen on the user perceived QoE

Figure 6 depicts moving average values of the client and server bandwidth measured. We smoothened the representation by using a window size of 100 seconds and chose a duration of 200 seconds, as the bandwidth heavily varies within each test run. It summarizes the bandwidth usage of the Textpad tests with default settings (no Speedscreen), with enabled Speedscreen (Speedscreen), and the combination of Speedscreen and Input Buffer (Buffer & Speedscreen). The lower half of Figure 6 reveals that the bandwidth consumption of a Citrix client combining the Input Buffer and the Speedscreen option is smaller than the used bandwidth of both the default client and the client with only Speedscreen enabled. As can be anticipated from the Input Buffer measurements in Section IV-A, a change in the client upload bandwidth results in an accordingly modified bandwidth consumption of the server, i.e. a higher client upload bandwidth leads to an increased bandwidth received in download direction. Again, a change in the packet rate sent by the client causes the server to send TCP acknowledgments accordingly and therefore modifies the bandwidth usage in both directions.

If we take a look at the QoE metrics shown in Figure 7 we see how the combination of Input Buffer and Speedscreen affects the test duration. The solid lines depict the QoE measured in a lossless network. Observe that the Citrix client running with a combination of Speedscreen and Input Buffer provides better QoE than the unmodified client. However, it is a little bit slower than the Citrix client running with Speedscreen only. This result is reasonable, as on the one hand the Speedscreen accelerates the visualization on the screen and therefore the user gets more direct feedback. On the other hand even the Speedscreen enhanced client is not independent from the server, and therefore the additional delay introduced by enabling the Input Buffer slightly reduces the improvement of the QoE.

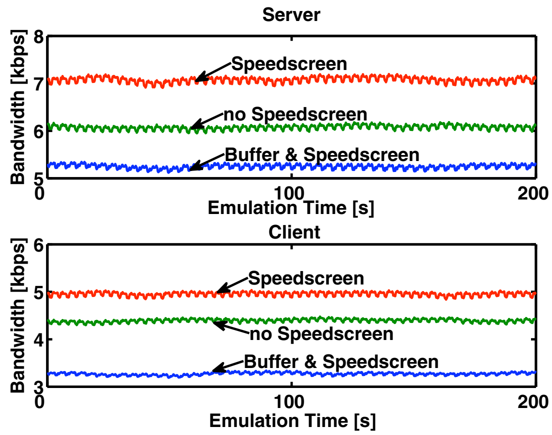

Fig. 6. Influence of different Speedscreen combinations on used bandwidth

Under network conditions with packet loss the effects on the QoE are different. The dashed lines in Figure 7 depict the test durations measured in a network with 2% packet loss. In this case a combination of Speedscreen and Input Buffer clearly outperforms the other two options. This improvement is based on the user feedback enhancement of Speedscreen as well as on the fact that the Input Buffer reduces the mean time the client has to wait for the server response in a given time interval under loss conditions, as explained in Section IV-A.

Considering these results, we come to the conclusion that combining Speedscreen and Input Buffer can further improve the responsiveness in congested networks.

V. Conclusion

Thin Client architectures are becoming increasingly popular as they offer more efficient utilization of hardware and software resources, energy savings, easier software maintenance, and a significant reduction of the total cost of ownership.

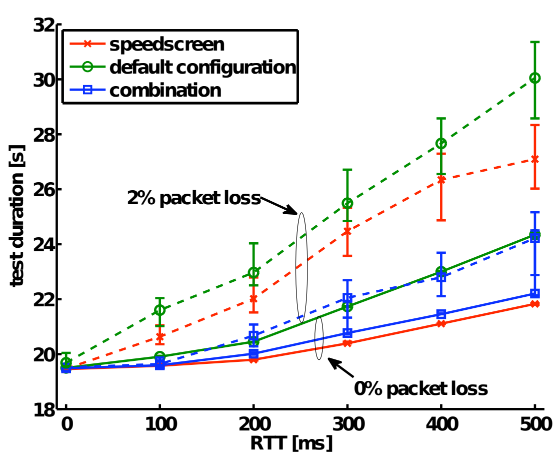

Fig. 7. Influences of combing Speedscreen with Input Buffer on the QoE

Used in WAN environments the Thin Client technology faces problems, which were not considered by design. However, the administrator is able to optimize the QoE of Citrix Thin Client users by adopting several options.

In highly congested networks it seems reasonable to activate the Input Buffer to reduce traffic and improve the responsiveness. Speedscreen will improve reaction times in scenarios without packet loss but with high delays, e.g. long distance connections with low utilization. Whenever packet loss cannot be avoided, enabling Speedscreen and the Input Buffer can improve the QoE noticeable.

Although the Input Buffer is a Citrix specific option it would be easily implementable in other Thin Client systems. Therefore the tradeoff between the delay saved by fewer retransmissions and the delay introduced by buffering the user input will be topic of our future work, as it is an interesting option for adapting of Thin Client systems to congested networks.

Acknowledgements

The authors would like to thank Prof. Tran-Gia for the fruitful discussions and his valuable comments.

References

[1] B. Emmert, A. Binzenhöfer, D. Schlosser, and M. Weiß, "Source Traffic Characterization for Thin Client Based Office Applications," in *13th EUNICE*, (University of Twente, the Netherlands), July 2007.
[2] B. Staehle, A. Binzenhöfer, D. Schlosser, and B. Boder, "Quantifying the Influence of Network Conditions on the Service Quality Experienced by a Thin Client User," in *MMB 2008*, (Dortmund, Germany), April 2008.
[3] T. Hoßfeld, P. Tran-Gia, and M. Fiedler, "Quantification of Quality of Experience for Edge-Based Applications," in *ITC20*, (Ottawa, Canada), June 2007.
[4] ITU-T Recommendation P.800.1, "Mean Opinion Score (MOS) Terminology," 2002.
[5] L. Deboosere, J. D. Wachter, P. Simoens, F. D. Turck, B. Dhoedt, and P. Demeester, "Thin Client Computing Solutions in Low- and High-Motion Scenarios," in *ICNS '07*, (Athens, Greece), June 2007.
[6] N. Tolia, D. G. Andersen, and M. Satyanarayanan, "Quantifying Interactive User Experience on Thin Clients," *IEEE Computer Society Press Computer*, vol. 39, no. 3, 2006.
[7] Juniper Networks, "Tuning Citrix for Optimal Performance with the WX/WXC Platforms." Application Note, August 2006.
[8] Citrix Systems, Inc., "Optimizing Citrix technology for operation over wireless Wide Area Networks." White Paper, April 2003.
[9] Citrix Systems, Inc., "Networking Issues Affecting Citrix MetaFrame Environments." White Paper, April 2003.
[10] Citrix Systems, Inc., "Citrix Independent Computing Architecture client on Nokia 9210 Communicator." White Paper, October 2003.
[11] Citrix Systems, Inc., "SpeedScreen Latency Reduction Explained." White Paper, December 2000.
[12] D. Schlosser, A. Binzenhöfer, and B. Staehle, "Performance Comparison of Windows-based Thin-Client Architectures," in *ATNAC'07*, (Christchurch, New Zealand), December 2007.
[13] M. Carson and D. Santay, "NistNet: A Linux-based Network Emulation Tool," *SIGCOMM Computer Communications Review*, vol. 33, no. 3, 2003.
[14] AutoHotkey. http://www.autohotkey.com/, last accessed: January 2008.