Performance Evaluation of Browser-Based Throughput Measurement for 100 Gb/s Infrastructure

Yannick Huber*, Marco Haeberle†, Benjamin Steinert†‡, Michael Menth†
* University of Stuttgart, BelWü-Koordination, Stuttgart, Germany
† University of Tuebingen, Chair of Communication Networks, Tuebingen, Germany
‡ University of Tuebingen, Zentrum für Datenverarbeitung, Tuebingen, Germany
Email: huber@belwue.de
{marco.haeberle,benjamin.steinert,menth}@uni-tuebingen.de

***Abstract*—Browser-based throughput tests are commonly used to evaluate end-to-end performance metrics in IP networks. They provide an easy-to-use throughput test tool. Unlike native software solutions such as iperf3, browser-based tests are restricted by the capabilities of the web browsers they use, which can potentially affect the accuracy and reliability of test results. This work investigates the limitations of browser-based throughput tests in a controlled lab environment equipped with a 100 Gb/s link. It is demonstrated that browser-based tests do not reach full link utilization, with the choice of browser having a significant impact on the measured throughput rate. The cause of the bandwidth limitation is determined by measuring the individual components of the test setup separately. The bottleneck is introduced by running the tests in a browser and limits the measurable throughput rate to below 10 Gb/s. The study reveals that browser-based tests are currently not suitable for the evaluation of connection speeds in networks with throughput rates of 10 Gb/s and beyond.**

I. Introduction

Throughput tests in general are commonly used to measure the bandwidth of connections in IP networks. One implementation of such throughput tests are browser-based tests. Browser-based throughput tests consist of a test server and a client executing the throughput test in a web browser via a web interface. Browser-based tests have the benefit of being easy to deploy since they require no special software on the test clients and are usable by non-technical users. Consequently, there are multiple examples of browser-based throughput tests used for measuring connections in existing research [1]–[6].

Running speed tests in a browser could potentially affect the maximum measurable throughput. In this work, these possible limitations of browser-based speed tests are investigated by measuring the maximum throughput speeds achievable under ideal conditions. This is done by running tests between two servers connected via a 100 Gb/s link. By comparing the results of browser-based throughput tests to the results measured by iperf3 and LibreSpeed CLI, the origin of possible throughput limitations are explored.

The rest of the paper is structured as follows. The paper first reviews the existing research regarding limitations of browser-based throughput tests in Section II. In Section III, the paper outlines the lab environment employed to evaluate browser-based throughput test implementations. The results measured in the lab environment are then presented in Section IV. Furthermore, the different test components are examined without the use of a browser, which identifies that the browser introduces a limitation to the measured bandwidth. Finally, in Section V this paper discusses the extent to which browser-based speed tests can be used to evaluate connection speeds.

This work was supported by the bwNET 2.0 project which is funded by the Ministry of Science, Research and the Arts Baden-Württemberg (MWK). The authors alone are responsible for the content of this paper.

II. Related Work

Existing research focuses on various limitations of browser-based speed tests. One focus is the evaluation of client devices and shows how outdated hardware significantly impacts the measured speeds, even at connection speeds below 1 Gb/s [7]. The authors also point out possible issues with using browser-based speed tests to evaluate Internet access speeds due to bottlenecks in home networks. Prior research also investigates best practices for deploying browser-based speed tests aimed at mitigating performance bottlenecks [8]. The work focuses on the use of browser-based speed tests for the network performance of household connections to their respective internet service provider.

Another area of research is the impact of latency on browser-based speed tests. It is documented how latency impacts the measured throughput rate, but also stated that a comparison of using a browser client and a command line client displays "no significant difference in the results reported from each modality" [9] under ideal conditions in a controlled lab environment. A key difference of the present work to existing work is the use of higher link speeds. Previous research demonstrates no difference between running a command line client and a browser-based client because both clients reach the limit introduced by the link speeds used in their experiments. By using a lab environment with higher link speeds, this particular limitation is removed, which allows exploring the limitations of the other test components. It becomes visible that running tests through a browser client greatly restricts the measured throughput rate compared to using a special build command line interface (CLI) application.

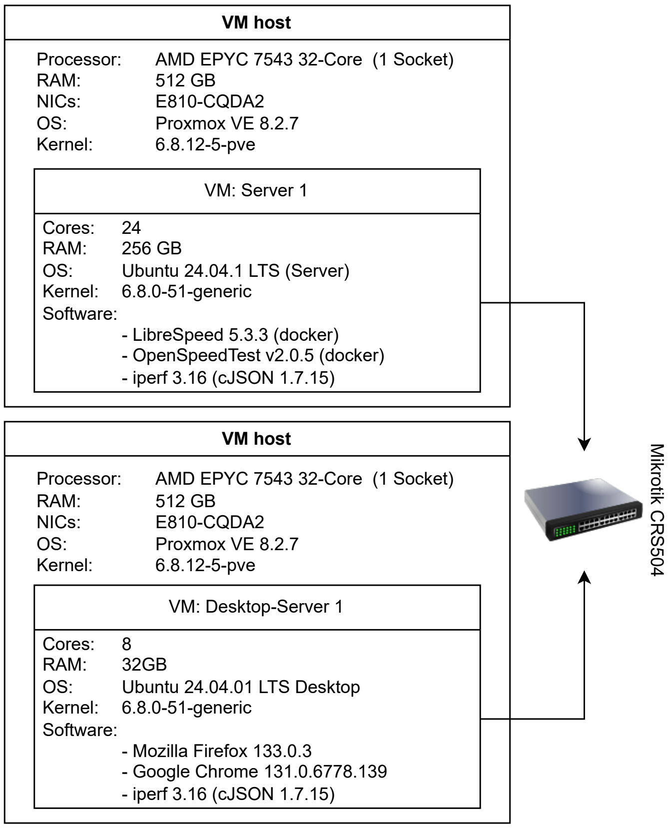

Fig. 1. Hardware and software configuration of the testbed.

III. Test Environment

A testbed is used to examine the possible throughput rates of browser-based throughput tests. It consists of two servers that are directly connected via a 100 Gb/s connection, providing ideal conditions regarding latency. The exact hardware and software used is specified in Figure 1. The bandwidth of the test setup is validated using iperf3, which is a throughput test tool that can be used for “active measurements to determine the maximum achievable bandwidth on IP networks“ [10]. A measurement utilizing iperf3 reports a data transfer rate of 94.1 Gb/s for the connection between the servers. LibreSpeed [11] and OpenSpeedTest [12] are selected for the evaluation of browser-based speed tests. This is done because they are prominent open-source speed test implementations that can be self-hosted. Both test tools use multiple parallel JavaScript XMLHttpRequests [13] to download large files and determine the network speed by monitoring the transfer rates. In the testbed, LibreSpeed and OpenSpeedTest are used without any additional test parameter tuning.

IV. Results

The execution of speed tests using the previously explained setup reports an average upload speed of 2.46 Gb/s and an average download speed of 3.41 Gb/s using Firefox and OpenSpeedTest. This is significantly lower than the speeds measured using Google Chrome, which achieves an average upload speed of 9.7 Gb/s and an average download speed of 9.46 Gb/s.

Similar results are examined using LibreSpeed speed instead of OpenSpeedTest. In this scenario, the measured average upload and download speeds using Firefox are 2.47 Gb/s and 2.98 Gb/s respectively. The average upload and download speeds with Google Chrome are measured at 10.64 Gb/s and 9.97 Gb/s. Though the throughput rate measured by Chrome is higher than the one measured by Firefox, the measured rate is still significantly lower than the rate measured by native throughput test implementations. LibreSpeed provides a CLI application that enables the execution of throughput tests against LibreSpeed servers without using a browser [14]. As shown in Table I, tests run over the same connection using this CLI application prove that the server can support higher speeds than the ones measured using a browser. The measurement results obtained using iperf3 additionally illustrate that the connection between the test server and client is not causing the limited bandwidth measured in the browser-based tests. Consequently, through the application of exclusion procedures, the browser implementation is identified as a limiting factor for the measured throughput rate. The additional overhead required for running the test inside a browser environment requires significantly more CPU resources and causes a bottleneck.

By monitoring hardware performance indicators, it is evident that Firefox approaches full CPU usage on the test client, as shown in Figure 2. The CPU usage when running the tests

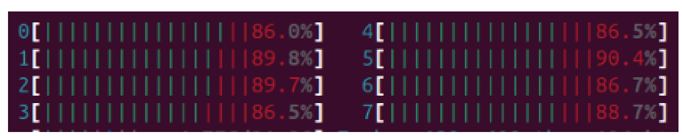

Fig. 2. Screenshot: CPU usage while running OpenSpeedTest in Firefox.

in Google Chrome differs from Firefox. Google Chrome is only fully utilizing a few instead of all available CPU cores as shown in Figure 3. Its maximum possible throughput rate appears to be restricted by the single core performance of the test server. For both browsers, the limitation of measured

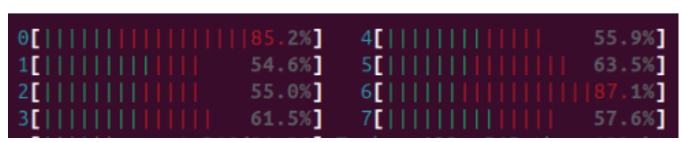

Fig. 3. Screenshot: CPU usage while running OpenSpeedTest in Google Chrome.

bandwidth can be attributed to the observed CPU usage. A direct comparison to executing the test via the LibreSpeed

	Openspeedtest						LibreSpeed					
	Firefox			Chrome			Firefox			Chrome		
	Min	Max	Average	Min	Max	Average	Min	Max	Average	Min	Max	Average
Upload (Gb/s)	2.4	2.56	2.46	9.15	9.96	9.7	2.30	2.61	2.47	10.09	11.06	10.64
Download (Gb/s)	3.16	3.6	3.41	9.31	9.65	9.46	2.78	3.2	2.98	9.52	10.79	9.97
							Baseline LibreSpeed-Server Capabilities using CLI tool					
							Max Upload: 49.45 Gb/s			Max Download: 43.33 Gb/s		

TABLE I
COMPARISON OF UP- AND DOWNLINK THROUGHPUT MEASURED WITH DIFFERENT TEST APPROACHES.

CLI tool demonstrates that using a browser creates additional overhead and reduces the measured bandwidth.

V. CONCLUSION

Browser-based throughput tests are used in current research to evaluate the throughput speed of connections in IP networks due to their ease of use. Running the tests inside a browser might falsify test results. To evaluate the use of browser-based throughput tests, the present work examined possible limitations of browser-based throughput tests in a controlled lab environment with ideal conditions regarding latency. The measured results demonstrate that browser-based throughput tests are not sufficient for testing throughput rates in the upper single-digit Gb/s range, due to the computational overhead introduced by executing the tests through a browser. Consequently, browser-based throughput tests should not be used to examine faster connections or when using devices with low computational power. Throughput tests run in Firefox fail to saturate a 2.5 Gb/s link even under ideal conditions regarding latency and using a client with sufficient hardware resources. Higher throughput rates are achievable with Google Chrome. However, the examined capabilities of Chrome are still not sufficient to measure 10 Gb/s connections. Tools specifically designed for conducting throughput tests like iperf3 are required for measuring at higher link speeds. If the use of browser-based throughput tests is required, monitoring the system resources of the client device is necessary to detect possible bottlenecks caused by the test running through a browser.

REFERENCES

[1] S. B. Mallikarjun, C. Schellenberger, C. Hobelsberger, and H. D. Schotten, "Performance Analysis of a Private 5G SA Campus Network," in *Mobile Communication - Technologies and Applications; 26th ITG-Symposium*, 2022, pp. 1–5.

[2] J. Wallace, M. Vukovic, T. Karimovic, W. Edwards, J. Islam, B. Snajder, S. Kovacevic, and A. V. Wallace, "A 5G/6G Infrastructure for Secure, High-Performance, Low-Latency Application Services," in *2023 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2023, pp. 926–933.

[3] E. Davies, A. Chung, M. Broadbent, A. Macleod, and N. Race, "5G in the Wild: Performance of C-Band 5G-NR in Rural Low-Power Fixed Wireless Access," in *2022 IEEE Future Networks World Forum (FNWF)*, 2022, pp. 18–23.

[4] P. Vanichchanunt, I. Yamyuan, P. Sasithong, L. Wuttisittikulkij, and S. Paripurana, "Implementation of Edge Servers on an Open 5G Core Network," in *2023 International Conference on Information Networking (ICOIN)*, 2023, pp. 642–645.

[5] H. Urasawa, H. Soya, K. Yamaguchi, and H. Matsue, "Throughput performance of local 5G downlink in SU-MIMO under outdoor environments," in *2024 International Conference on Information Networking (ICOIN)*, 2024, pp. 357–362.

[6] P. Vanichchanunt, O. Ritruechai, N. Wuttiananchai, P. Thossaporn, L. Wuttisittikulkij, and S. Paripurana, "Implementation of 5G Network Slicing Using Open Source Software," in *2024 12th International Electrical Engineering Congress (iEECON)*, 2024, pp. 1–6.

[7] N. Feamster and J. Livingood, "Measuring internet speed: current challenges and future recommendations," *Commun. ACM*, vol. 63, no. 12, p. 72–80, Nov. 2020. [Online]. Available: https://doi.org/10.1145/3372135

[8] K. MacMillan, T. Mangla, M. Richardson, and N. Feamster, "Best Practices for Collecting Speed Test Data," *SSRN Electronic Journal*, 01 2022.

[9] K. MacMillan, T. Mangla, J. Saxon, N. P. Marwell, and N. Feamster, "A Comparative Analysis of Ookla Speedtest and Measurement Labs Network Diagnostic Test (NDT7)," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 7, no. 1, Mar. 2023. [Online]. Available: https://doi.org/10.1145/3579448

[10] ESnet, "iperf3," https://software.es.net/iperf/, accessed: 2025-01-21.

[11] librespeed.org, "LibreSpeed GitHub," https://github.com/librespeed/speedtest, accessed: 2025-01-21.

[12] OpenSpeedTest™, "OpenSpeedTest Website," https://openspeedtest.com/, accessed: 2025-01-21.

[13] WHATWG community™, "XMLHttpRequest Living Standard," https://xhr.spec.whatwg.org/, accessed: 2025-03-21 - Last Updated 11 March 2025.

[14] librespeed.org, "LibreSpeed-Cli GitHub," https://github.com/librespeed/speedtest-cli, accessed: 2025-01-22.