



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 Tübingen, Chair of Communication Networks, Tübingen, Germany

‡ University of Tübingen, Zentrum für Datenverarbeitung, Tübingen, Germany



- Motivation
- Browser-based Throughput Tests
- Test Environment
- Results
- Conclusion & Future Work

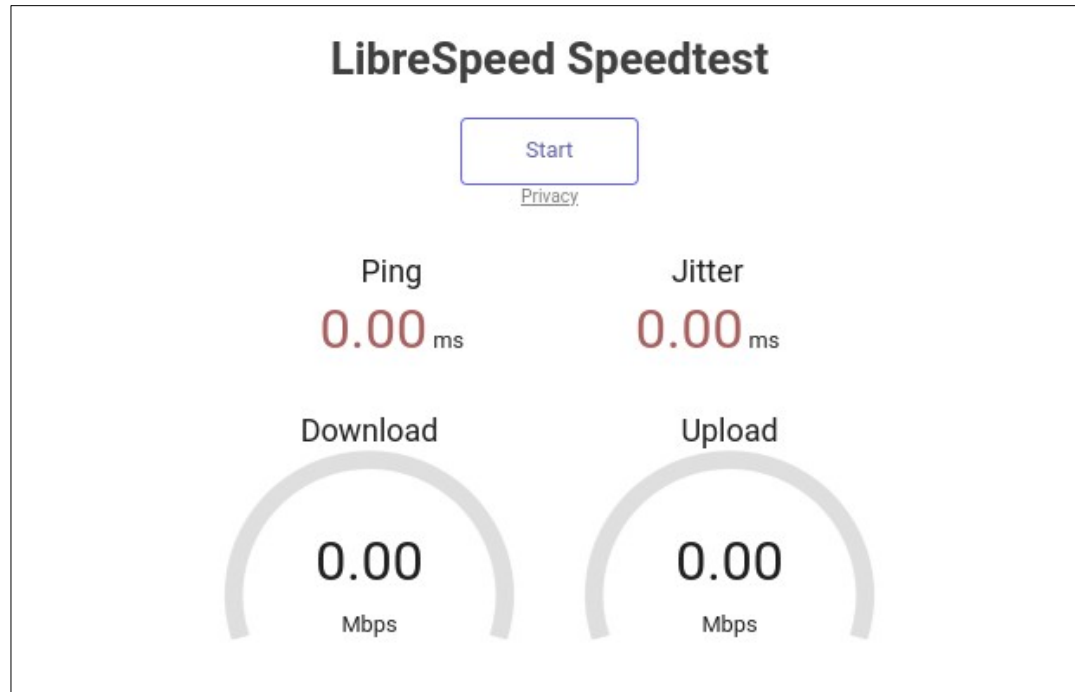
1. Motivation

- Customer complains about connection issues
 - First verification step
 - Executable by the customer
- Use in research

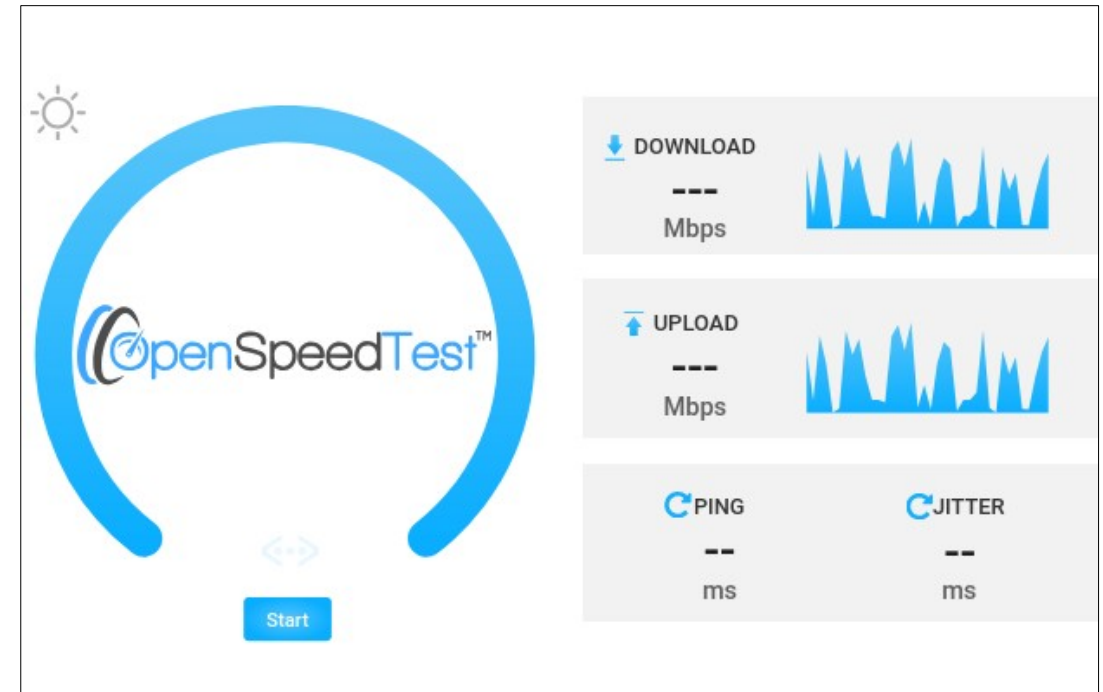
2. Browser-Based Throughput Tests

- Tool for measuring bandwidth
 - Server & client application
 - Done by saturating the connection
- No special client-side software required
 - Accessible to none technical users
 - Prominent as easy deployable solution
- Tested implementations
 - LibreSpeed v5.3.3
 - OpenSpeedTest v2.0.5

2. Web-GUIs



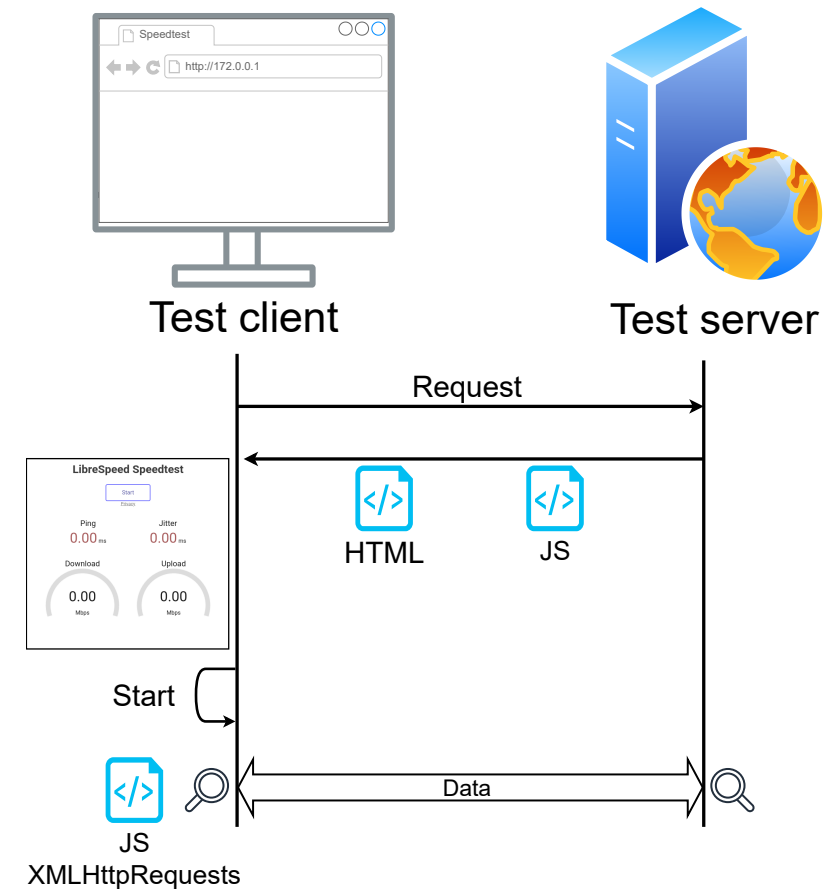
<https://github.com/librespeed/speedtest>



<https://github.com/openspeedtest/Speed-Test>

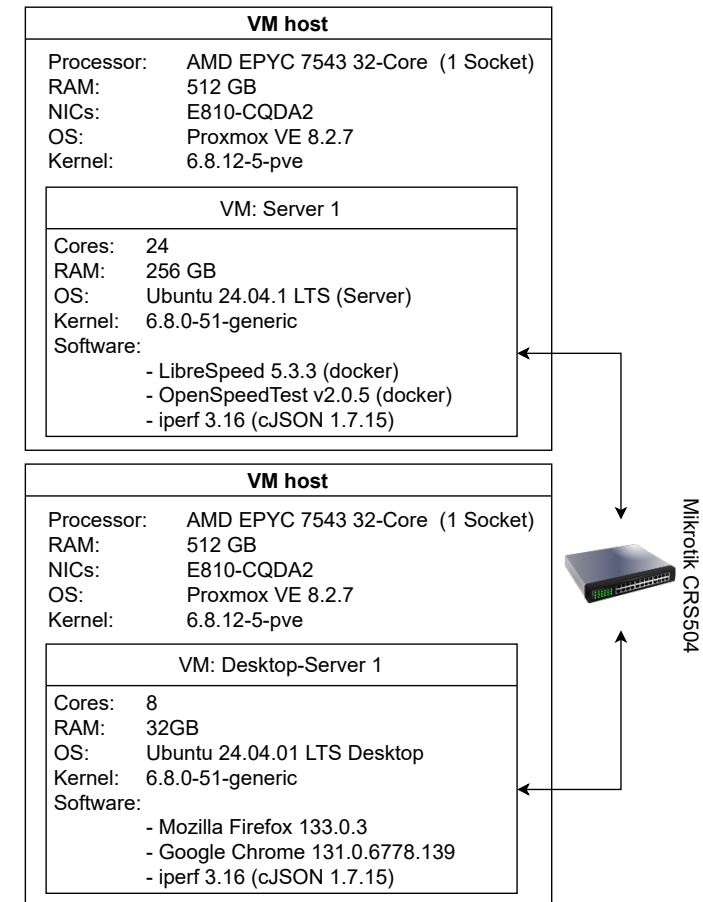
2. Test Execution

- Web server providing static files
- Client loading test application via HTTP
 - Client web interface
 - Test execution script
- Test execution in the browser
 - Downloading large files using parallel JavaScript XMLHttpRequests
 - Monitoring data transfer



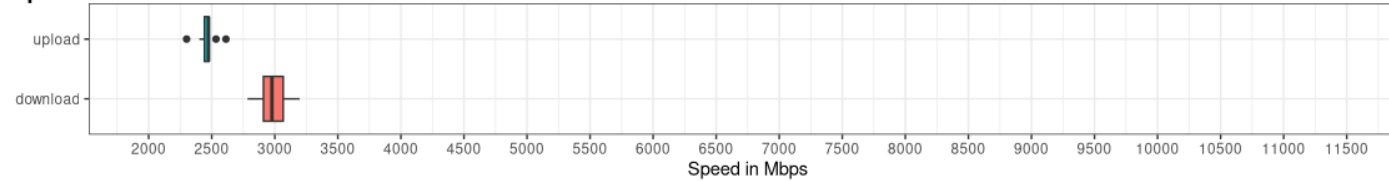
3. Test Environment

- 2 Servers
 - Connected via 100 Gb/s Link
 - Connection verified using iperf3
- Server 1 running test server application
- Server 2 running test browsers
 - Google Chrome
 - Firefox
- Multiple test executions

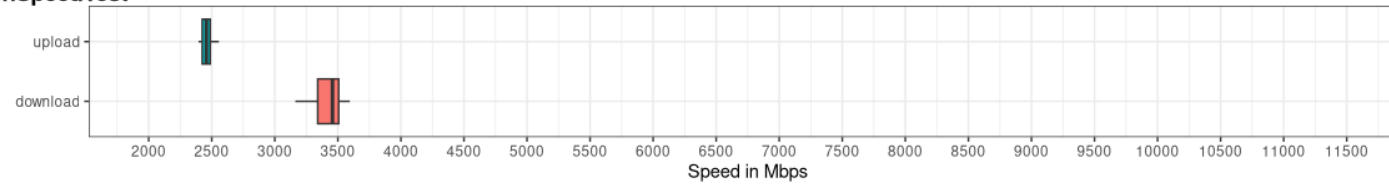


4. Results

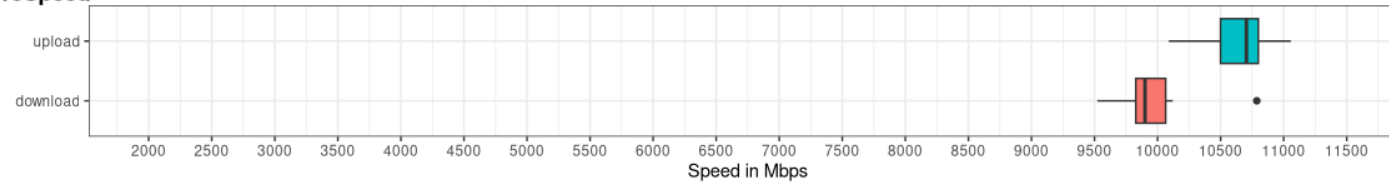
Firefox LibreSpeed



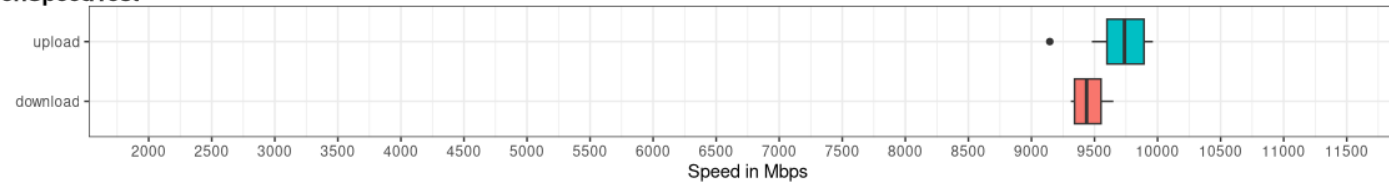
Firefox OpenSpeedTest



Chrome LibreSpeed



Chrome OpenSpeedTest



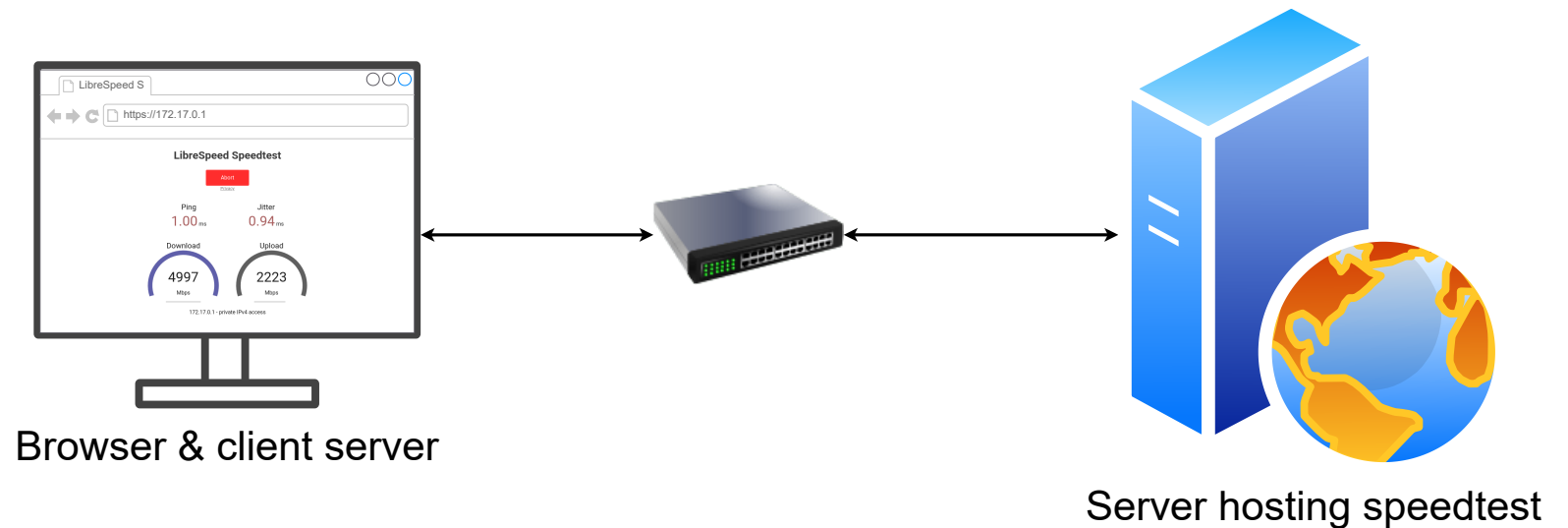
4. Results

- Comparable between test tools
- Differences between browsers
- Significantly slower than available link speed

		Firefox		Chrome	
		Min	Average	Min	Average
OpenSpeed Test	Upload (Gb/s)	2.4	2.46	9.15	9.7
	Download (Gb/s)	3.16	3.41	9.31	9.46
LibreSpeed	Upload (Gb/s)	2.30	2.47	10.09	10.64
	Download (Gb/s)	2.78	2.98	9.52	9.97

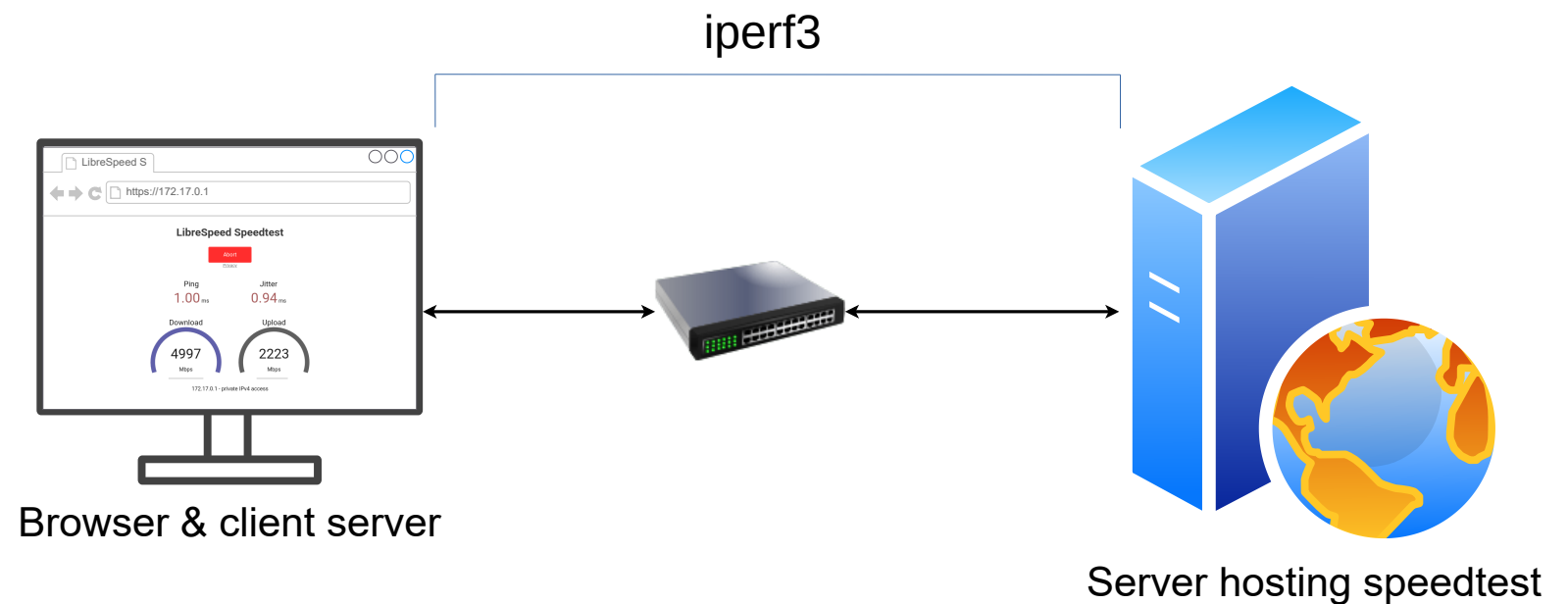
4. Results – Possible Causes

- Connection
- Test server
- Client system
- Browser



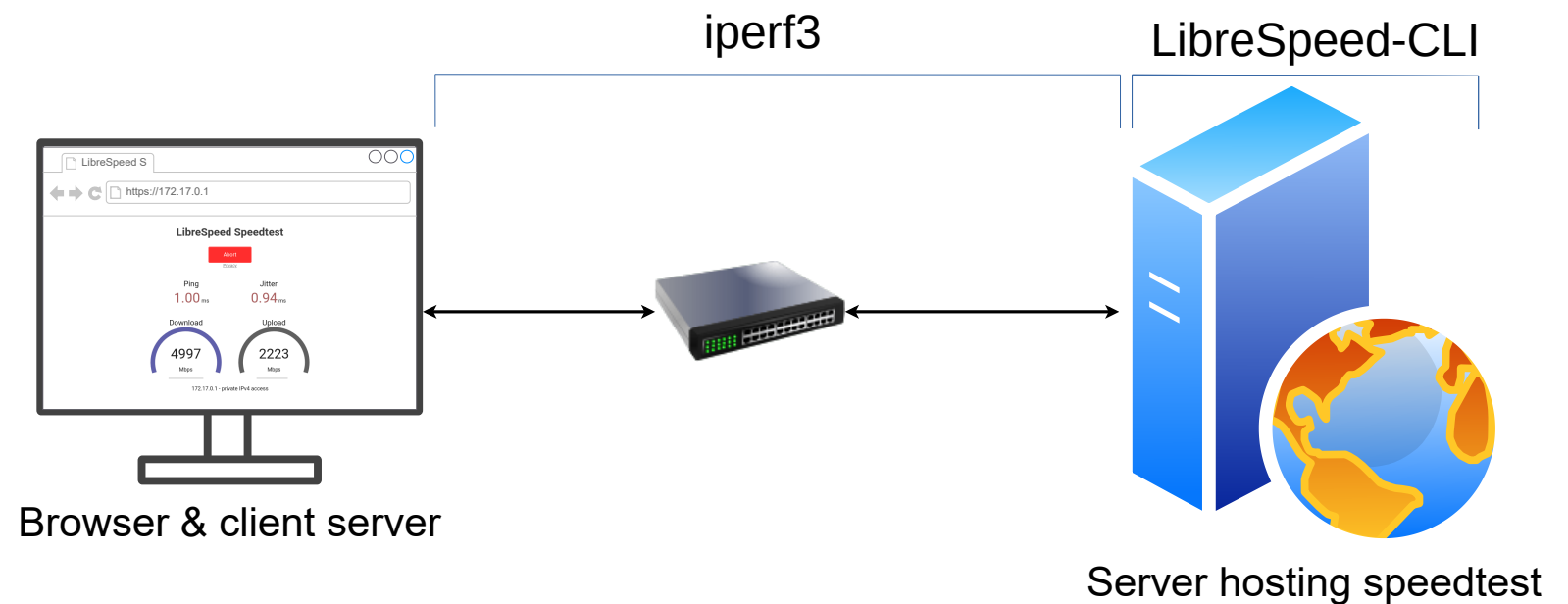
4. Results – Possible Causes

- Connection
- Test server
- Client system
- Browser



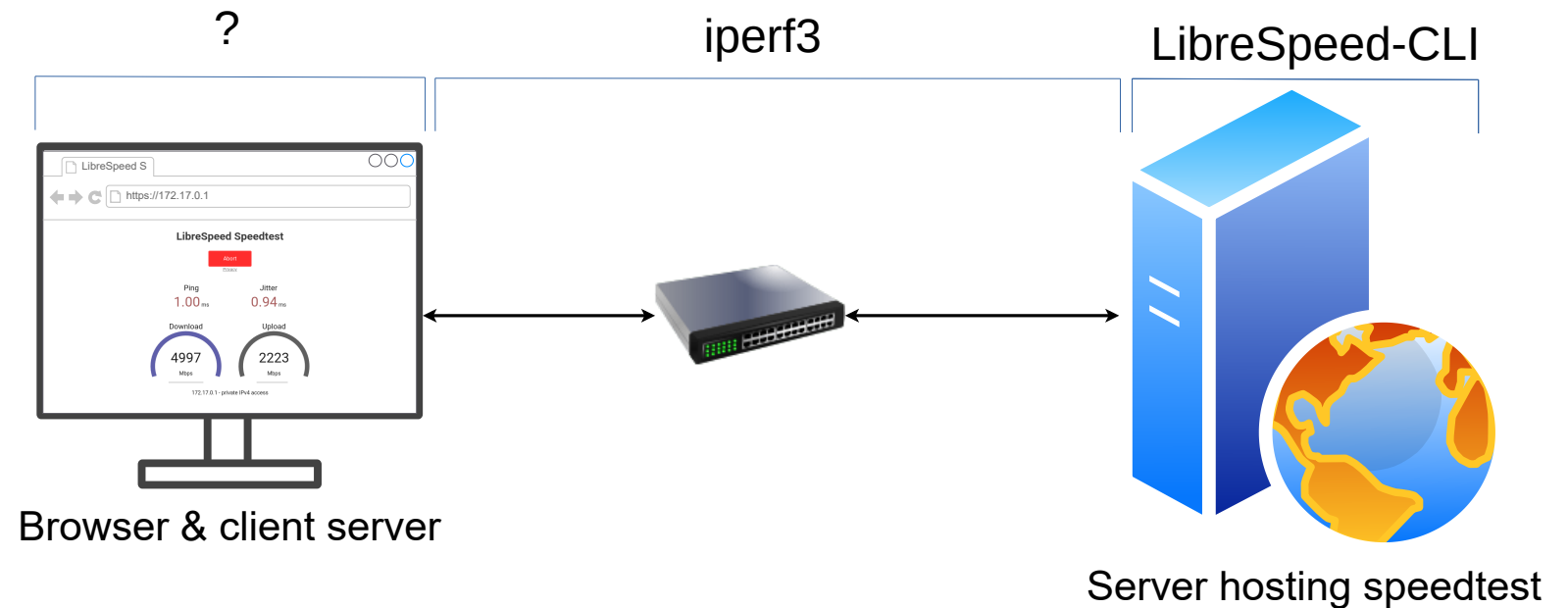
4. Results – Possible Causes

- Connection
- Test server
- Client system
- Browser



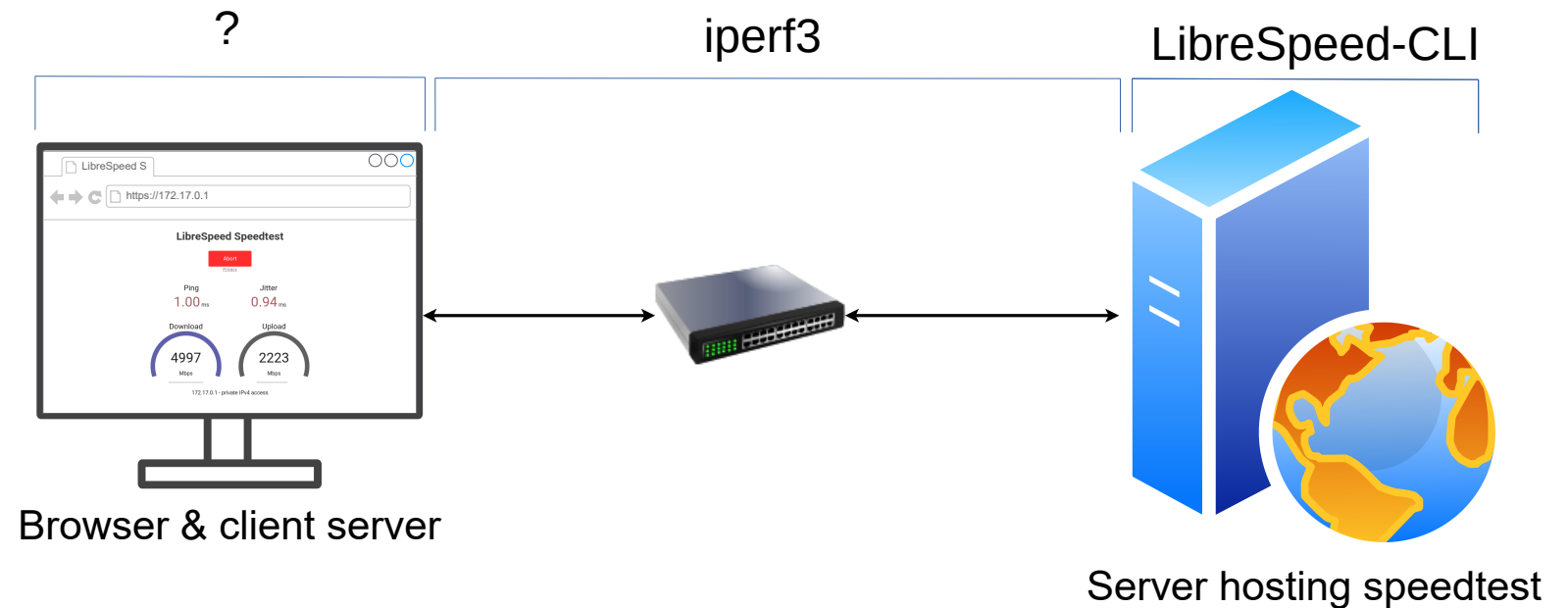
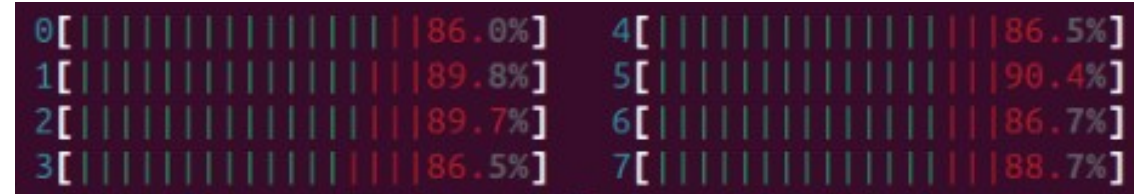
4. Results – Possible Causes

- Connection
- Test server
- Client system
- Browser



4. Results – Possible Causes

- Connection
- Test server
- Client system*
- Browser



4. Results – Findings

- Client-CPU fully utilized
 - 8 Cores of a AMD EPYC 7543
- Specialized tools like iperf3 not limited
- Efficiency of the implementation?

- Conclusion
 - Browser-based throughput tests are not sufficient for testing throughput rates in the upper single-digit Gb/s range
 - If use of browser-based tests is required the system resources must be closely monitored
 - Useful tool at lower speeds
- Future Work
 - WebAssembly Implementation of a Browser based Throughput Test

Thank you for your attention!

Questions?

This work was supported
by the bwNET 2.0 project
which is funded by the
MWK.



E-mail: huber@belwue.de

