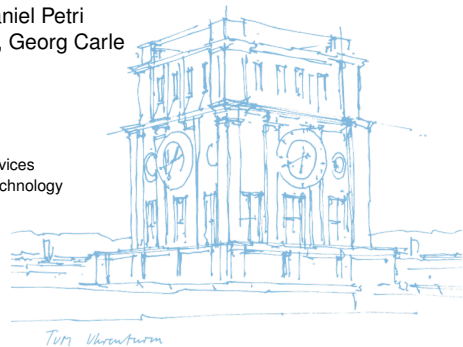
BrowsEm: Model-based Web Site Loading Emulation

Kilian Holzinger, Florian Klein, Daniel Petri
Stefan Lachnit, Sebastian Gallenmüller, Georg Carle

Tuesday 22nd July, 2025

Chair of Network Architectures and Services
School of Computation, Information, and Technology
Technical University of Munich



Web protocol research commonly based on prototypes and measurements

Fully synthetic evaluations:

- Allows for targeted understanding of behavior
- Unknown effect on complex web applications

In-browser measurements:

- Complex software systems, hard to modify
- Flexibility? Reproducibility?

Introduction

Web protocol research commonly based on prototypes and measurements

Fully synthetic evaluations:

- Allows for targeted understanding of behavior
- Unknown effect on complex web applications

In-browser measurements:

- Complex software systems, hard to modify
- Flexibility? Reproducibility?

BrowsEm:
Emulator for reproducing network activity during web site loading

- Model of recorded browser behavior and request dependencies
- Emulation of network path characteristics
- Flexible support for wide range of standards and implementations, e. g., HTTP, QUIC, or TLS
- Wide-spread components (e. g., libcurl) and modifiable building blocks

Introduction

Web protocol research commonly based on prototypes and measurements

Fully synthetic evaluations:

- Allows for targeted understanding of behavior
- Unknown effect on complex web applications

In-browser measurements:

- Complex software systems, hard to modify
- Flexibility? Reproducibility?

BrowsEm:
Emulator for reproducing network activity during web site loading

- Model of recorded browser behavior and request dependencies
- Emulation of network path characteristics
- Flexible support for wide range of standards and implementations, e. g., HTTP, QUIC, or TLS
- Wide-spread components (e. g., libcurl) and modifiable building blocks

⇒ Reproducible measurement platform for realistic web workloads

- For **protocol researchers:** Assessment of experimental implementations
- For **web site operators:** Evaluation of deployment changes

Related Work: Approaches for Web Traffic Modeling

Analytical models

- Statistical models of user interaction with website, e. g., [1, 7, 5]
- Commonly derived from Internet measurements, e. g., SURGE [2] or others [6, 10]
- Implemented in `ns-3` simulator module [9]

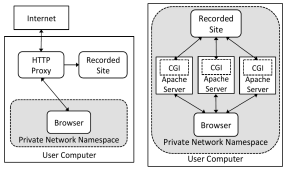
Related Work: Approaches for Web Traffic Modeling

Analytical models

- Statistical models of user interaction with website, e. g., [1, 7, 5]
- Commonly derived from Internet measurements, e. g., SURGE [2] or others [6, 10]
- Implemented in `ns-3` simulator module [9]

Trace-based

- “Record and replay” of HTTP requests
- Mahimahi [8]
  - Proxy to record HTTP requests
  - Network emulation, supporting multiple connections
  - Browser for loading website



(a) RecordShell (b) ReplayShell

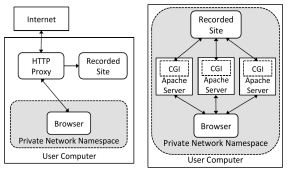
Related Work: Approaches for Web Traffic Modeling

Analytical models

- Statistical models of user interaction with website, e. g., [1, 7, 5]
- Commonly derived from Internet measurements, e. g., SURGE [2] or others [6, 10]
- Implemented in `ns-3` simulator module [9]

Trace-based

- “Record and replay” of HTTP requests
- Mahimahi [8]
  - Proxy to record HTTP requests
  - Network emulation, supporting multiple connections
  - Browser for loading website



(a) RecordShell (b) ReplayShell

This work

- Trace-based
- Includes request dependencies
- Additional modeling parameters
- Not browser based, allows for protocol modifications

Approach

1. Scraping
   - devtools[1]-controlled headless Chromium browser
   - Write key log file
   - Export HTTP Archive (HAR) file containing detailed request information, such as
     - HTTP header information
     - Request subtimings: *blocked*, *dns*, *connect*, *tls*, *send*
   - Traffic sniffing
     - PyShark[2] to decrypt and analyze
     - Estimation of path RTT: TCP `tsopt`, handshake timestamps, ICMP echo request / reply, (QUIC spin bit)
     - Estimation of path capacity: PPrate [4, 3]

[1] https://chromedevtools.github.io/devtools-protocol/

[2] http://kiminewt.github.io/pyshark/

Approach

1. Scraping

- devtools[1]-controlled headless Chromium browser
- Write key log file
- Export HTTP Archive (HAR) file containing detailed request information, such as
  - HTTP header information
  - Request subtimings: *blocked*, *dns*, *connect*, *tls*, *send*
- Traffic sniffing
  - PyShark[2] to decrypt and analyze
  - Estimation of path RTT: TCP `tsopt`, handshake timestamps, ICMP echo request / reply, (QUIC spin bit)
  - Estimation of path capacity: PPrate [4, 3]

2. Modeling

- Replace interpretation of received data with delay and transaction dependency information
- HAR *initiator* field, complemented by timing inference
- Request and response timings, headers, and body sizes
- Network path parameters (delay, capacity, loss)
- Stored as editable JSON configuration file

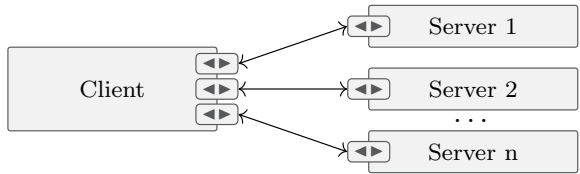
[1] `https://chromedevtools.github.io/devtools-protocol/`

[2] `http://kiminewt.github.io/pyshark/`

Approach

3. Emulation

- Linux namespaces and NetEm for separate paths
- Rust application, asynchronous tasks for each server, client, and transaction
- Included server options:
  - Actix web for HTTP/1.0, HTTP/1.1, HTTP/2
  - quinn for HTTP/3
  - nginx
- Client
  - libcurl through bindings
  - Five QUIC libraries available
  - Requests started in accordance with timing information



4. Postprocessing

Evaluation

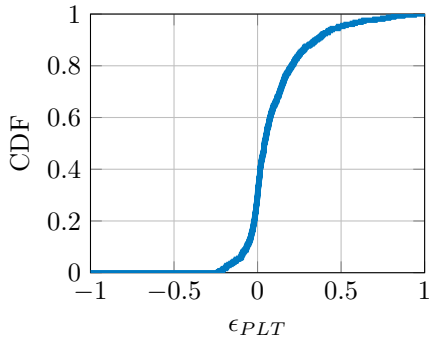
Goals: Assess emulation quality

Dataset: Top 1000 web sites[3], 922 usable samples

Approach:
- Assess page load time (PLT)
- Compare subtimings of scraped and emulated web sites
- Investigate model stability

[3] https://www.domcop.com/top-10-million-domains

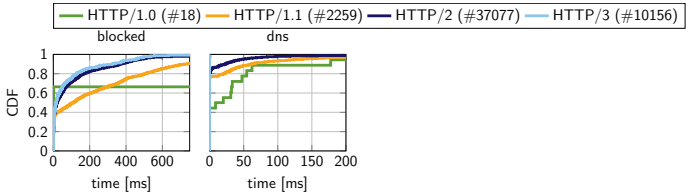
Evaluation: Page Load Time



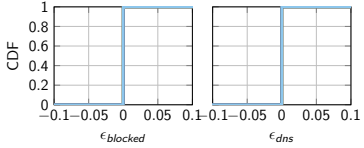
Relative differences $\epsilon_{PLT} = \frac{s_{PLT} - e_{PLT}}{s_{PLT}}$ of scraped s_{PLT} and emulated PLT e_{PLT}

- $\approx$ 80% page loads have less than $\pm 0.25 \epsilon_{PLT}$
- Well met dependency tree of requests and timings

Evaluation: Transaction Subtimings



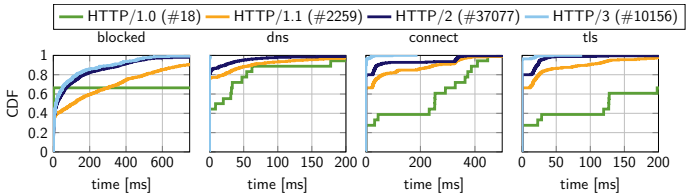
Transaction phase durations of scraped data set



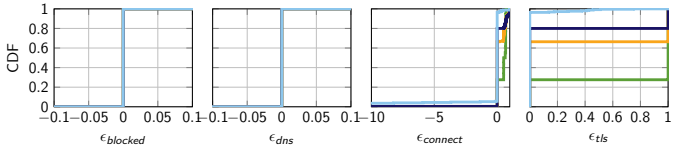
Relative difference to emulation; $\epsilon_P = \frac{s_P - e_P}{s_P}$

- *blocked* and *dns* are met

Evaluation: Transaction Subtimings



Transaction phase durations of scraped data set

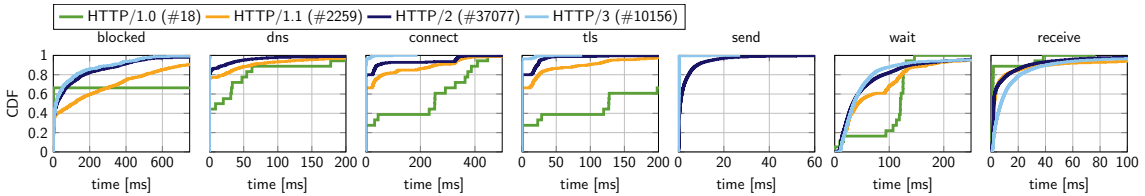


Relative difference to emulation; $\epsilon_P = \frac{s_P - e_P}{s_P}$

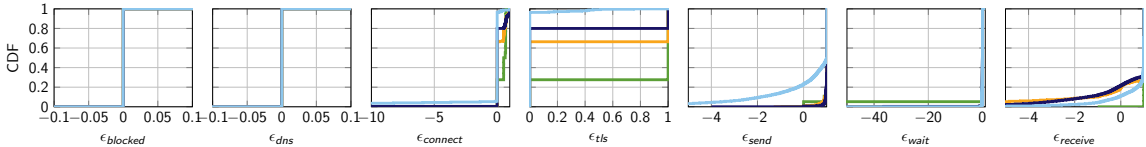
- *blocked* and *dns* are met

- Many reused connections
- Faster emulation
- Crypto libraries? RTT estimates?

Evaluation: Transaction Subtimings



Transaction phase durations of scraped data set



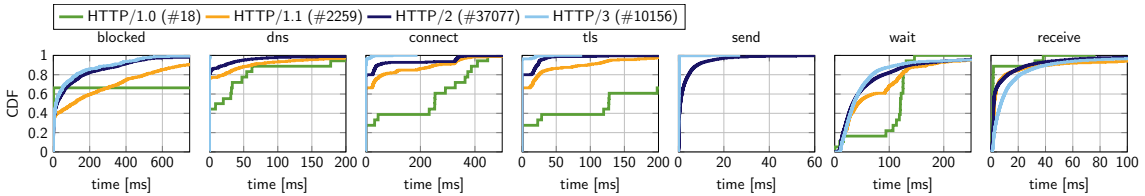
Relative difference to emulation; $\epsilon_P = \frac{s_P - e_P}{s_P}$

- *blocked* and *dns* are met

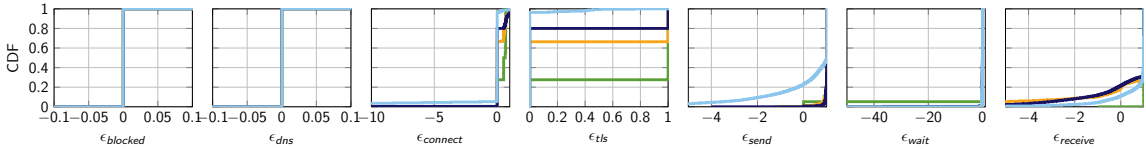
- Many reused connections
- Faster emulation
- Crypto libraries? RTT estimates?

- *send* absolute value is short, emulation faster
- Application limited? Wrong path estimates?
- *wait* is mostly met, emulator depends on RTT
- *receive* emulation is faster, c. f. *send*

Evaluation: Transaction Subtimings



Transaction phase durations of scraped data set



Relative difference to emulation; $\epsilon_P = \frac{s_P - e_P}{s_P}$

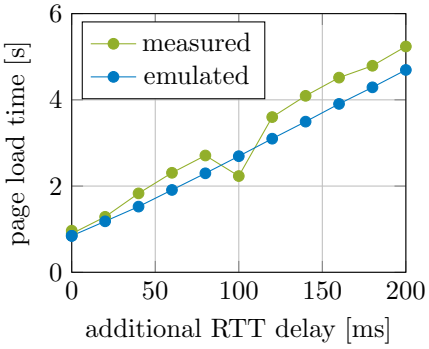
- *blocked* and *dns* are met

- Many reused connections
- Faster emulation
- Crypto libraries? RTT estimates?

- *send* absolute value is short, emulation faster
- Application limited? Wrong path estimates?
- *wait* is mostly met, emulator depends on RTT
- *receive* emulation is faster, c. f. *send*

⇒ Transaction phases are emulated reasonably well

Evaluation: Stability Analysis with Changed RTT



- Alter RTT parameter of emulation model
- Scrape same site with added delay as ground-truth
- Emulation follows trend of measurement
- Model captures impact of RTT on PLT

Limitations and Future Work

- Inherent limitations
  - Browser behavior (e. g., JS execution, rendering) captured implicitly
- More model parameters
- Improve quality of parameter estimation
  - Path RTT and capacity estimation
  - Complement browser-reported timing with network trace information
- Add and use more information sources (e. g., packet traces)
  - Chunked / application limited transmissions
- Support other browser with similar interfaces, e. g., Firefox
- Broader stability testing

Conclusion

- Network workload emulator for web applications
- Focus on transport and HTTP application layer
- Solves orchestration and reproducibilty
- Extensible design, ready for your prototype
- Useful for protocol research or performance optimization

Slides (pdf)[4]

Paper (pdf)[5]

Code[6]

[4] http://www.net.in.tum.de/fileadmin/bibtex/publications/papers/holzinger2025browsem_slides.pdf

[5] http://www.net.in.tum.de/fileadmin/bibtex/publications/papers/holzinger2025browsem.pdf

[6] https://github.com/holzingk/BrowsEm/

Bibliography

[1] P. Barford and M. Crovella.
Measuring web performance in the wide area.
27(2):37–48.

[2] P. Barford and M. Crovella.
Generating representative web workloads for network and server performance evaluation.
In *Proceedings of the 1998 ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '98/PERFORMANCE '98, page 151–160, New York, NY, USA, 1998. Association for Computing Machinery.

[3] S. Bauer, J. Janelidze, B. Jaeger, P. Sattler, P. Brzoza, and G. Carle.
On the Accuracy of Active Capacity Estimation in the Internet.
In *2023 IEEE/IFIP Network Operations and Management Symposium (NOMS 2023)*, Miami, USA, May 2023.

[4] T. En-Najjary and G. Urvoy-Keller.
Pprate: A passive capacity estimation tool.
In *2006 4th IEEE/IFIP Workshop on End-to-End Monitoring Techniques and Services*, pages 82–89. IEEE, 2006.

[5] Hyoung-Kee Choi and J. Limb.
A behavioral model of web traffic.
In *Proceedings. Seventh International Conference on Network Protocols*, pages 327–334. IEEE Comput. Soc.

[6] J. Lee, M. Gupta, and I. Corp.
A new traffic model for current user web browsing behavior.
2007.

[7] B. A. Mah.
An empirical model of http network traffic.
In *Proceedings of INFOCOM '97*, volume 2, pages 592–600 vol.2, 1997.

Bibliography

[8] R. Netravali, A. Sivaraman, S. Das, A. Goyal, K. Winstein, and H. Balakrishnan.
Mahimahi: Accurate record-and-replay for HTTP.
In *2015 USENIX Annual Technical Conference (USENIX ATC 15)*, 2015.

[9] U. Paro, F. Chiariotti, A. A. Deshpande, M. Polese, A. Zanella, and M. Zorzi.
Extending the ns-3 QUIC module.
In *Proceedings of the 23rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 19–26. ACM.

[10] R. Pries, Z. Magyari, and P. Tran-Gia.
An HTTP web traffic model based on the top one million visited web pages.
In *Proceedings of the 8th Euro-NF Conference on Next Generation Internet NGI 2012*, pages 133–139. IEEE.