On the Cost of Using Happy Eyeballs for Transport Protocol Selection

Giorgos Papastergiou†, Karl-Johan Grinnemo*, Anna Brunstrom*, David Ros†, Michael Tüxen×, Naeem Khademi+, Per Hurtig*

†Simula Research Laboratory, *Karlstad University,

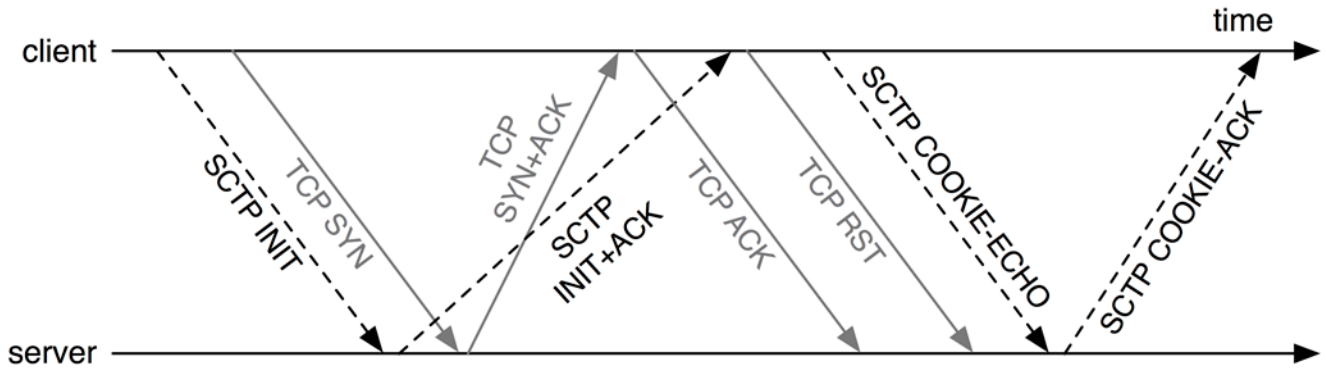
×Fachhochschule Münster, +University of Oslo

Horizon 2020
European Union funding
for Research & Innovation

Introduction

- Deployment of new transport protocols is a difficult task
- How to know if a new protocol is supported along the whole end-to-end path?
  - Try it
- If not supported a fallback mechanism is needed
  - Testing protocols serially can be time consuming
- Happy Eyeballs for transport protocol selection

Happy Eyeball Example: TCP and SCTP



- **Concurrent** initiation of TCP and SCTP
- **Preferred** connection attempt wins

Related Work

- Happy Eyeballs introduced as a way of **promoting the use of IPv6** [RFC6555]
- Transport Happy Eyeballs was proposed as a mecahnism to **run HTTP over SCTP** [Wing10]
- Ongoing work in IETF **TAPS** WG
  - "… explain how to select and engage an appropriate protocol and how to discover which protocols are available for the selected service between a given pair of end points"

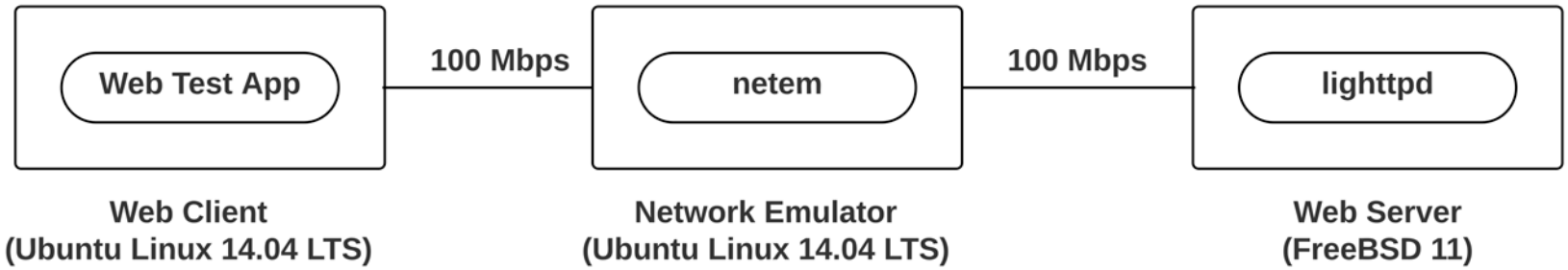
[RFC6555] D. Wing and A. Yourtchenko. Happy Eyeballs: Success with Dual-Stack Hosts. RFC 6555 (Proposed Standard), Apr. 2012.
[Wing10] Wing, D. and P. Natarajan, "Happy Eyeballs: Trending Towards Success with SCTP", https://tools.ietf.org/html/draft-wing-tsvwg-happy-eyeballs-sctp-02 (work in progress), October 2010.

Cost of Happy Eyballs?

- Increased server load
  - CPU load
  - Memory usage
- Increased network traffic

- Goal: Assess impact of happy Eyeballs on server load

Experiment

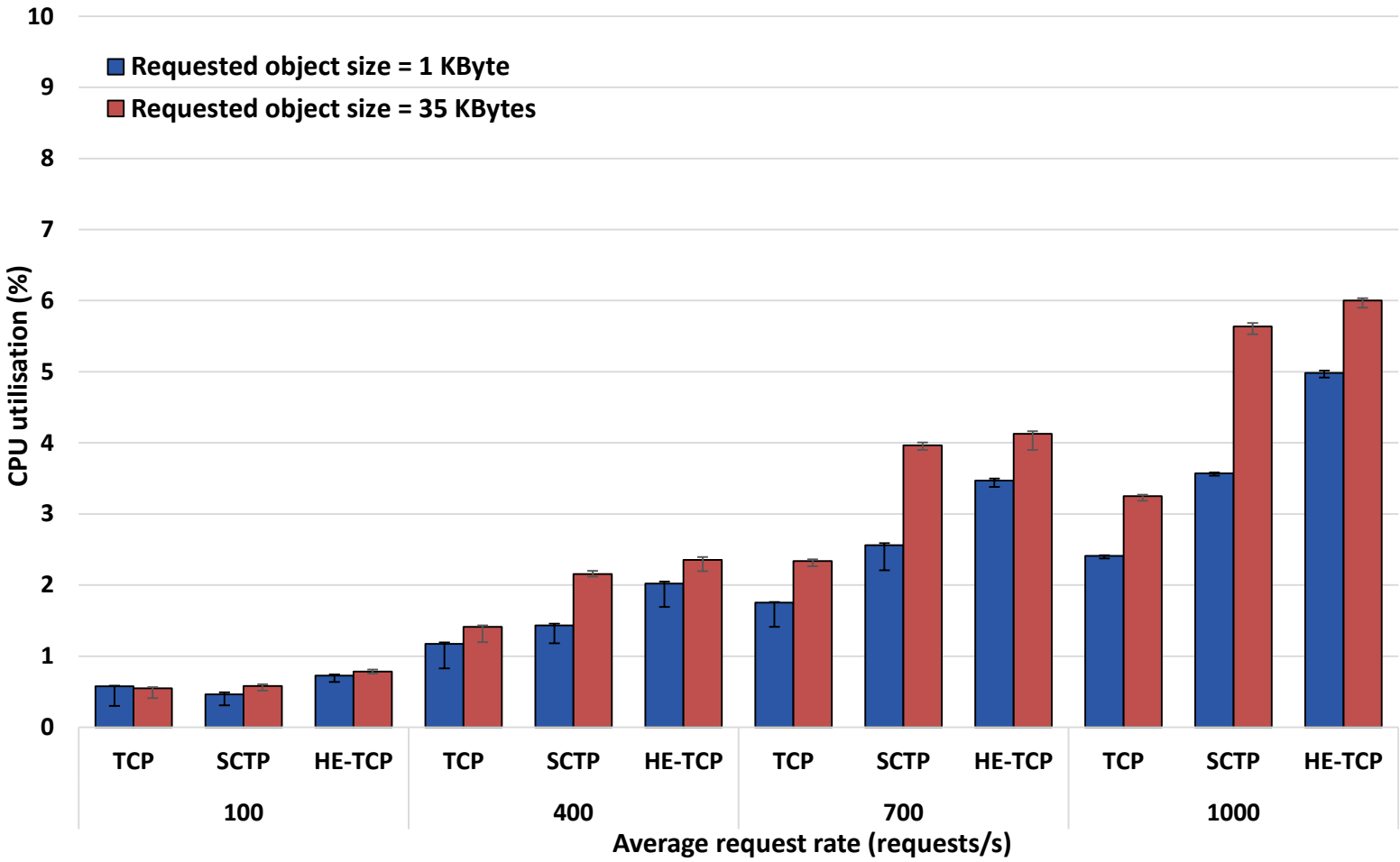


- Length of each test run: 600 s
- Exponentially distributed http reqs.
  - [100,1000] reqs./s
- Metrics:
  - CPU utilization
  - Kernel memory usage

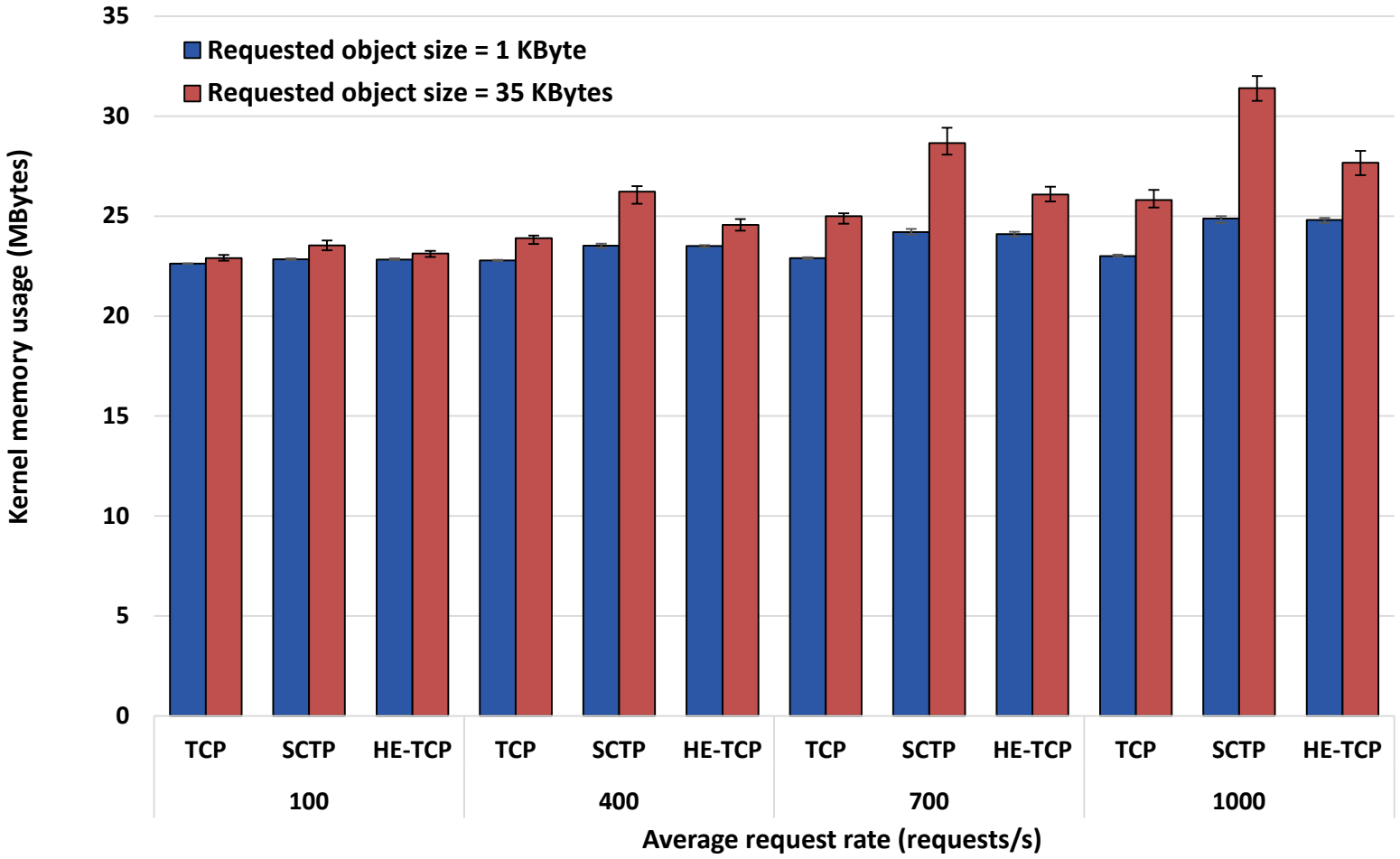
Basic Test Case

- Always results in a TCP connection
- No caching of previous connection attempts
- Unencrypted connections

CPU Utilization in Basic Test Case



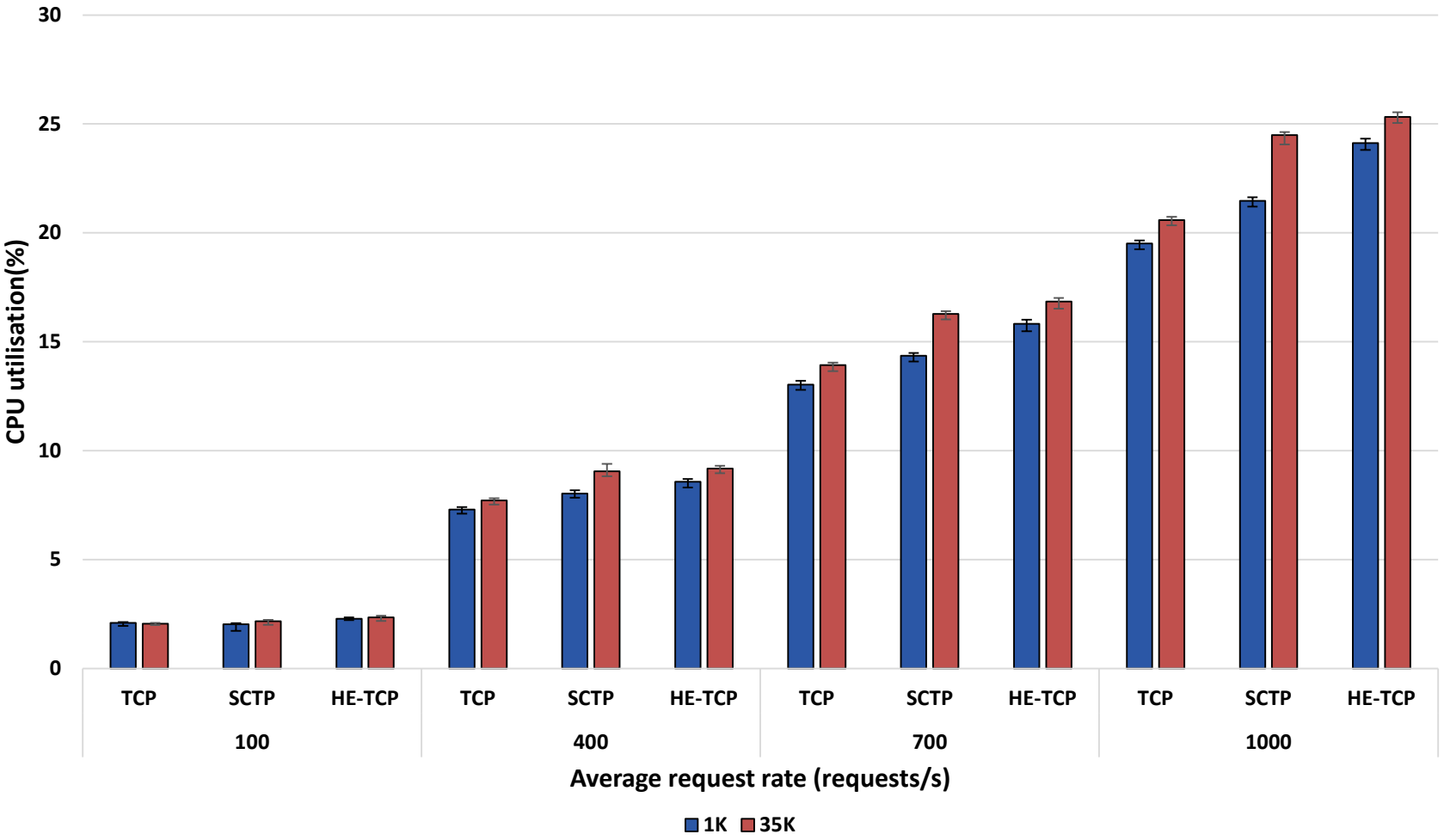
Memory Usage in Basic Test Case



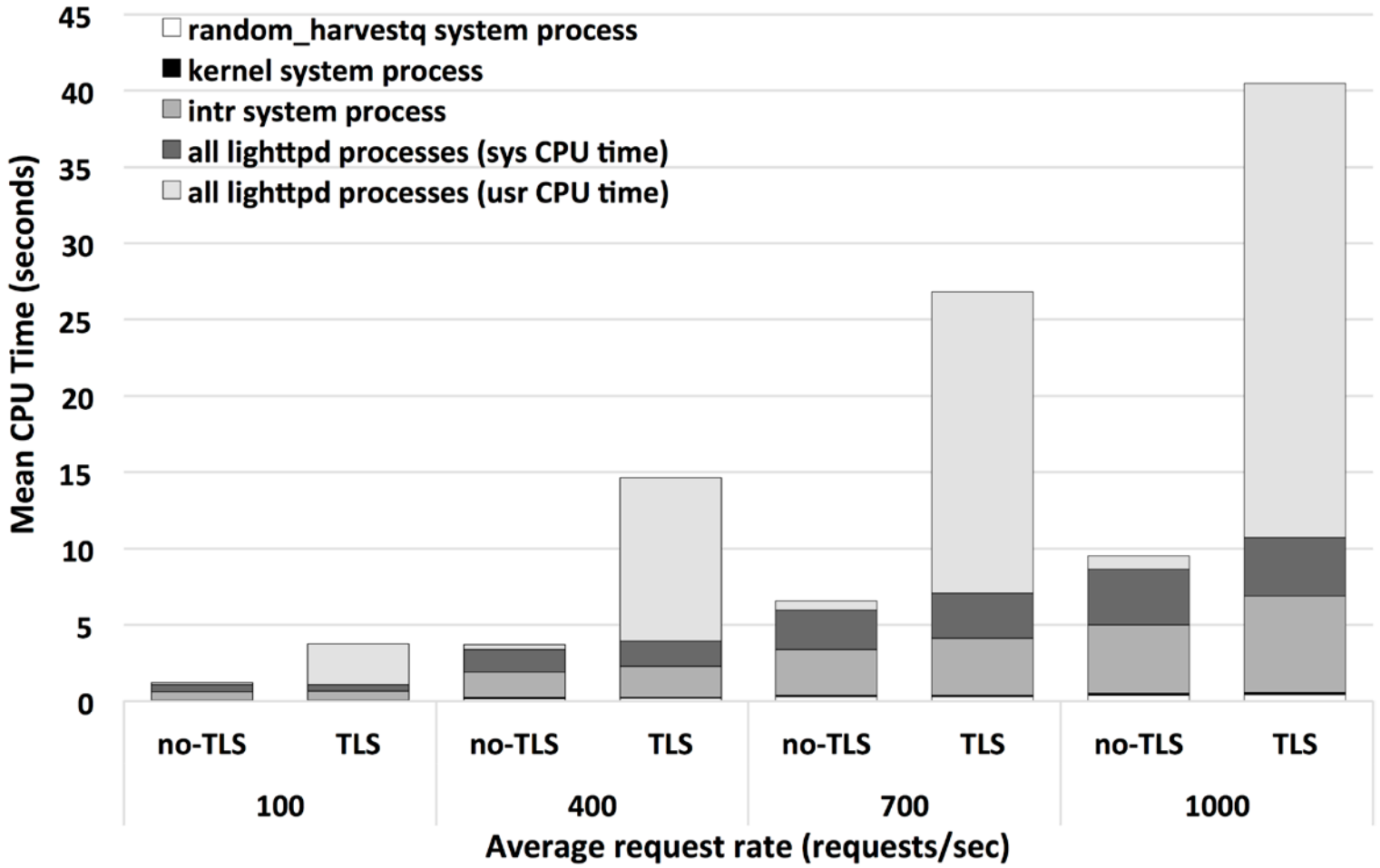
TLS Test Case

- No caching of previous connection attempts
- TLS-encrypted connections

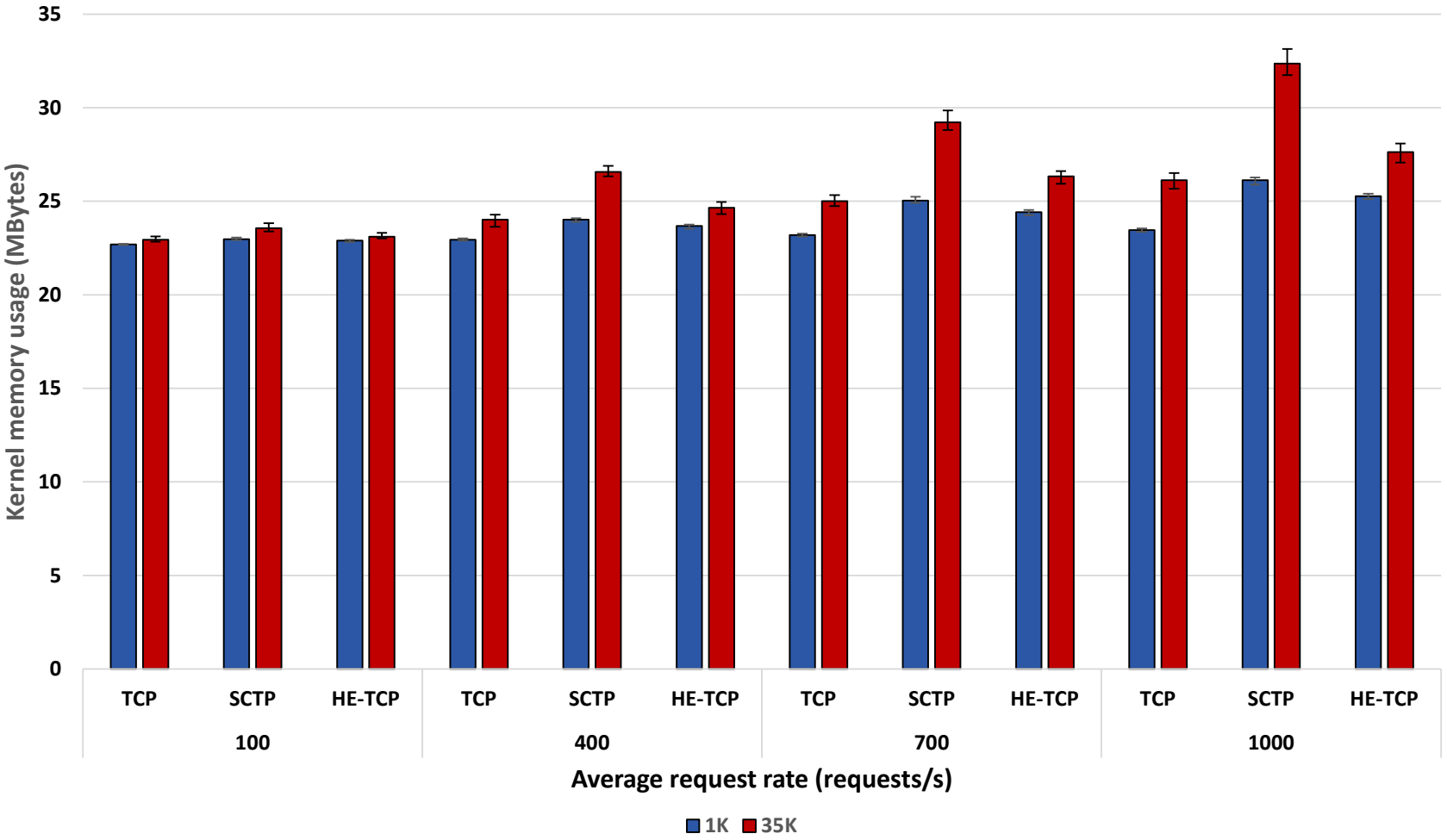
CPU Utilization in TLS Test Case



Sharing of CPU load in TLS Test Case



Memory Usage in TLS Test Case



Cache Test Case

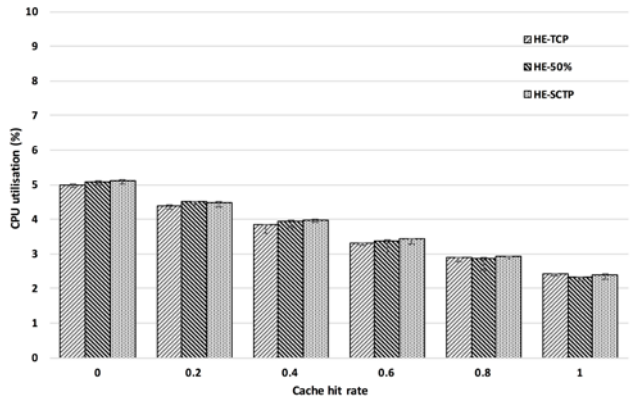
- Caching of previous connection attempts
- Both unencrypted and TLS-encrypted connections
  - HE-TCP: always results in a TCP connection
  - HE-SCTP: always results in a SCTP connection
  - HE-50%: 50% chance TCP/SCTP connection

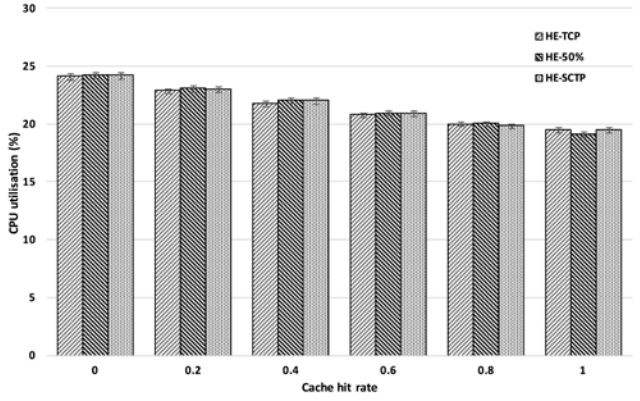
Cache Hit Ratio vs. CPU Utilization

Unencrypted

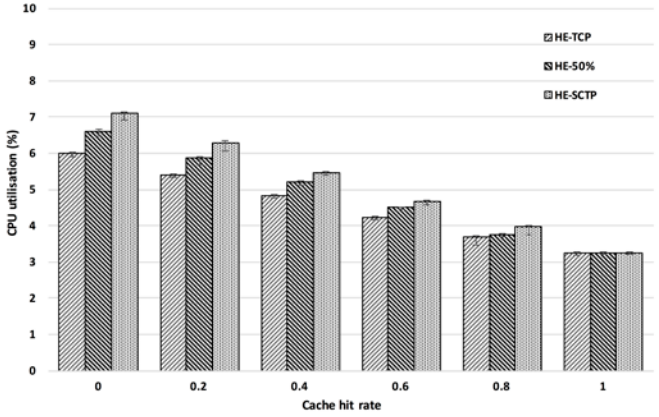
TLS-encrypted

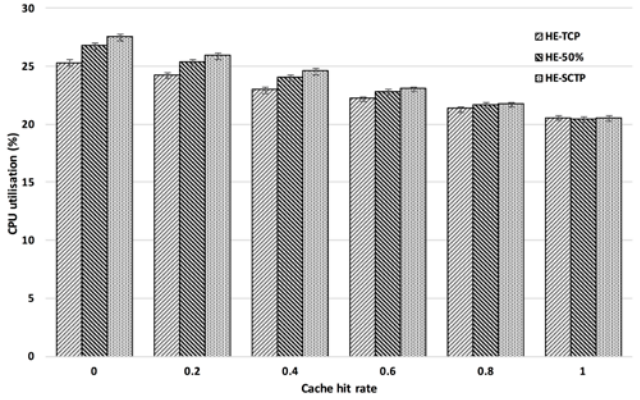
1 KiB





35 KiB





Conclusion

- Happy Eyeball is a feasible transport-selection mechanism
- Small increase in CPU utilization
  - Around 10% for 35 KiB web objects
  - Encryption has larger impact on CPU load
  - Caching can reduce load further
- Basically no increase in memory usage

Future Work

- More extensive evaluations with more than two feasible transport solutions
- Transport service library with Happy Eyeballs support
  - https://github.com/NEAT-project/neat
- Evaluation of Happy Eyeball in real-world scenarios with middleboxes etc.