Performance and Fairness Evaluation of IW10 and Other Fast Startup Schemes

Michael Scharf <michael.scharf@googlemail.com>

This work was performed at the Institute of Communication Networks and Computer Engineering (IKR) at the University of Stuttgart.

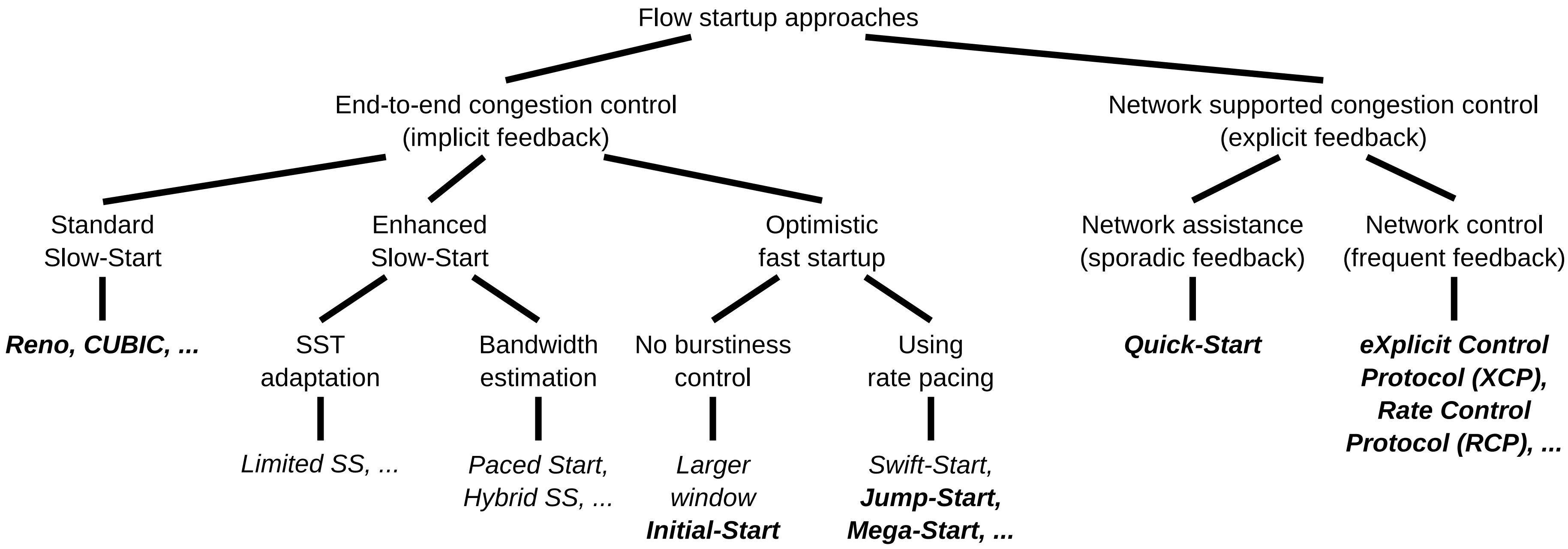
Disclaimer
Individual contribution

- Report of **old work** from the years 2008 and 2009
  - Original intention was to show that a network-supported scheme like Quick-Start is indeed required
  - IW10 was considered as alternative (called Initial-Start)
  - Quite surprisingly, IW10 outperformed all other variants
- **First, preliminary results**:

  M. Scharf. Quick-Start, Jump-Start, and other fast startup approaches: Implementation issues and performance. Presentation at 73rd IETF Meeting, ICCRG, Nov. 2008
- Full **reference for this work**:

  M. Scharf. Fast Startup Internet Congestion Control for Broadband Interactive Applications. PhD thesis, University of Stuttgart, submitted Nov. 2009

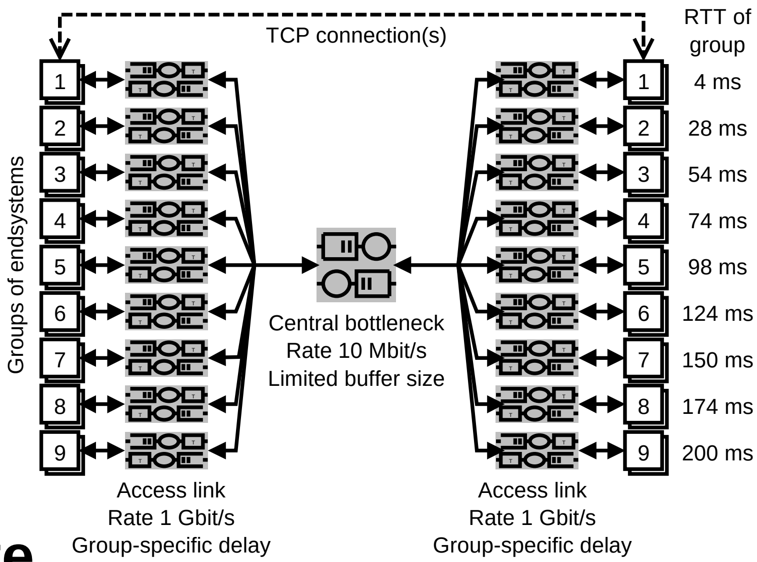
Fast startup congestion control
Scope of the study



- TCP's standard Slow-Start with CUBIC (**SS**)
- **Initial congestion window of 10 MSS**, called Initial-Start (**IS**)
- Jump-Start of M. Allman et al., slightly modified to reduce aggressiveness (**JS**)
- Quick-Start TCP extension according to RFC 4782 (**QS**)
- Rate Control Protocol (**RCP**)
- … and others

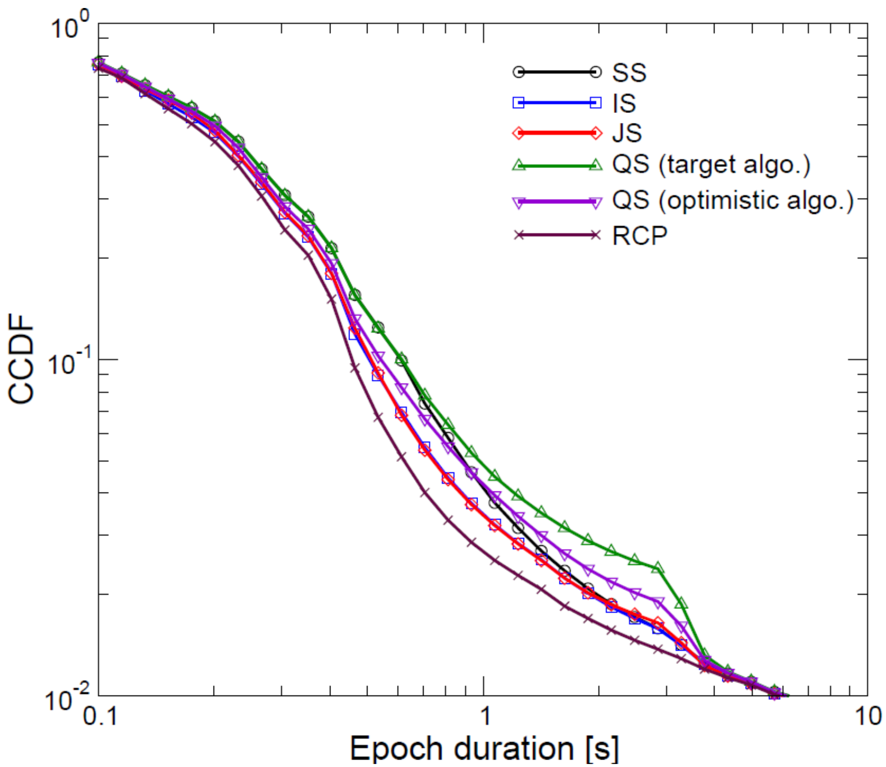
Fast startup congestion control
Evaluation methodology

- **Simulations**
  - Simulation with **Linux code** using the NSC framework
  - Own Linux patches for all TCP extensions, and an own tool for RCP

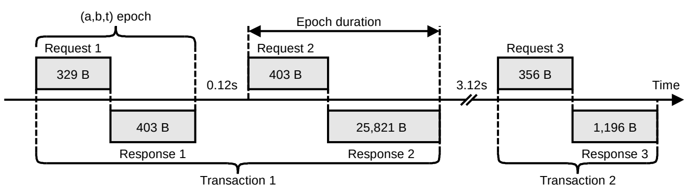


- **Considered scenarios**
  - Subset of the **TCP evaluation suite**
  - **Dumbbell topology** with 450 endsystems and 9 different RTTs
  - Bottleneck typically 10 Mbit/s, 50 packets buffer, drop tail
  - Replay of measured **Internet traces** in a-b-t format as recommended in TCP evaluation suite
- **Implementations** verified by testbed measurements

Selected performance results
Possible speedup of the different variants

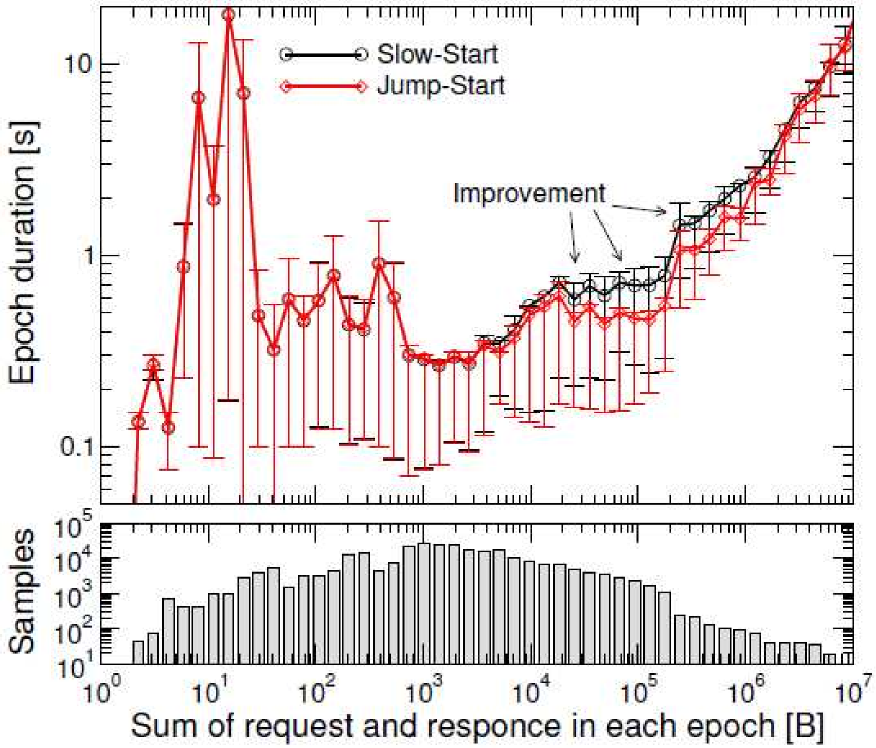


- Simulation with Linux 2.6.18
- Dumbbell topology with 10 Mbit/s bottleneck and 9 different RTTs
- 450 clients and 450 servers
- Default TCP configuration, except for larger buffer sizes (8 MiB)
- Replayed traces in a-b-t format
- Mean downlink load 35%
- Metric: Epoch duration

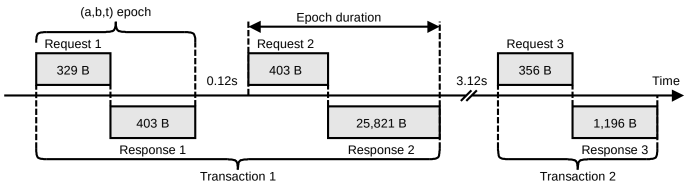


- Performance metric: **Response time** of a-b-t transfers (“epoch duration”)
- **Speedup of mid-sized transfers** by larger initial window
- **Overall benefit is rather small**: Many short transfers, many small RTTs

Selected performance results
Insight into the workload

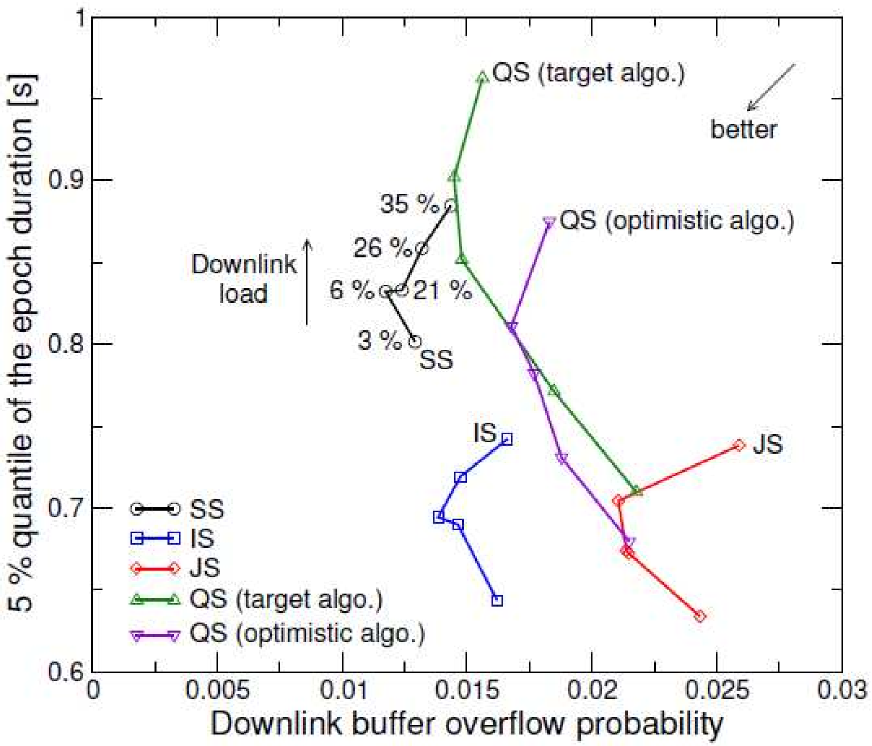


- Simulation with Linux 2.6.18
- Dumbbell topology with 10 Mbit/s bottleneck and 9 different RTTs
- 450 clients and 450 servers
- Default TCP configuration, except for larger buffer sizes (8 MiB)
- Replayed traces in a-b-t format
- Mean downlink load 35%
- Metric: Epoch duration



- Most TCP connections are rather short in the workload traces
- **Only transfers larger than 10 KB** can benefit
- Average **improvement less than 1 s** even for larger transfers

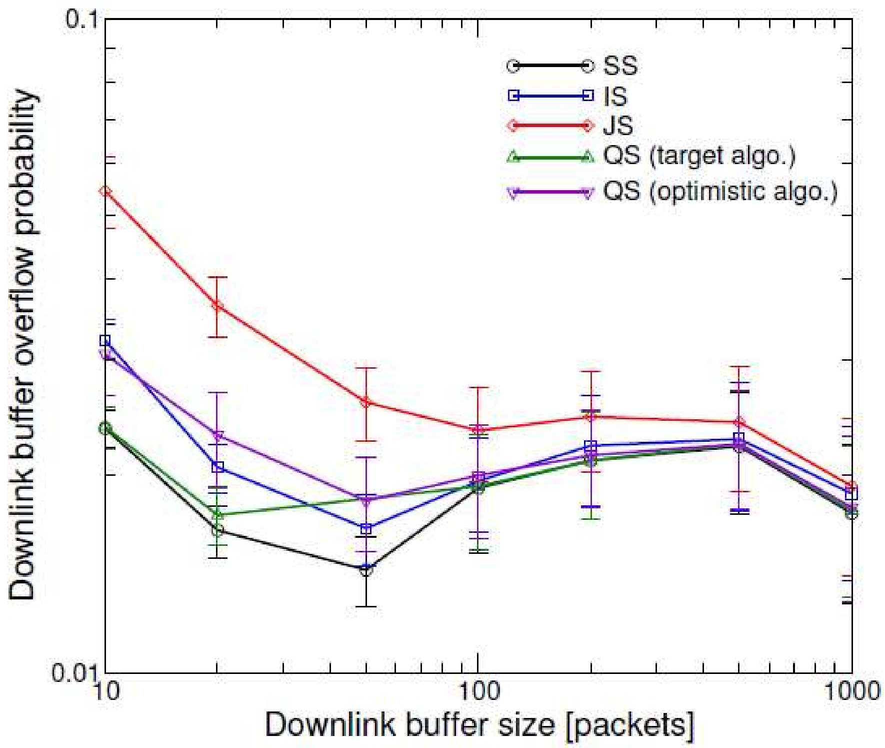
Selected performance results
Trade-off between speedup and packet loss



- Simulation with Linux 2.6.18
- Dumbbell topology with 10 Mbit/s bottleneck and 9 different RTTs
- 450 clients and 450 servers
- Default TCP configuration, except for larger buffer sizes (8 MiB)
- Replayed traces in a-b-t format
- Variable load up to ca. 40% (due to tool limitation to ca. 1000 stacks)

- **IW10 increases loss probability by 0.5%**
- **Other considered schemes** are not faster, but have a larger loss rate
- Result: **IW10 outperforms other schemes**

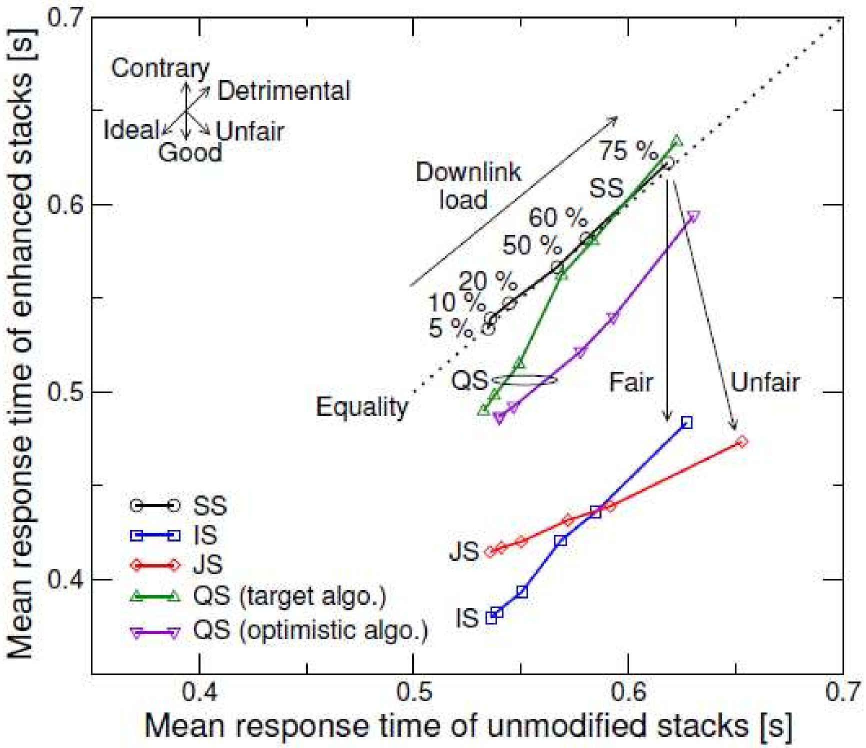
Selected performance results
Sensitivity to bottleneck buffer size



- Simulation with Linux 2.6.18
- Dumbbell topology with 10 Mbit/s bottleneck and 9 different RTTs
- 450 clients and 450 servers
- Default TCP configuration, except for larger buffer sizes (8 MiB)
- Replayed traces in a-b-t format
- Mean downlink load 35%

- Obviously, **small buffers** (<50 packets) are a problem
- Fast startups only **moderately increase the packet loss rate** if reasonably sized buffers (50-100 packets, or AQM) present

Selected performance results
Fairness to unmodified stacks



- Simulation with Linux 2.6.18
- Dumbbell topology with 10 Mbit/s bottleneck and 9 different RTTs
- 450 clients and 450 servers 50% CUBIC, 50% fast startup
- Default TCP configuration, except for larger buffer sizes (8 MiB)
- Synthetic workload model for HTTP/1.0, response sizes from truncated pareto distribution with mean 14 KB, shape parameter 1.1, truncation at 10 MB

- **Scenario**: 50% of stacks use fast startup, 50% unchanged (CUBIC)
- **IW10** is rather fair and hardly impacts other flows
- Result: **IW10 outperforms other schemes**

Conclusion

Results

- **Moderate benefit** of fast startups for larger transfers
- **IW10 works rather well and is quite fair**
- **More sophisticated schemes** tend to be worse
- **Network support** such as Quick-Start can overcome some limitations, but it has **problems of its own**

Recommendations for further work

- Study more extensively the use of **rate pacing**, even if results suggests that it may not be needed for 10 MSS
- Rethink **error recovery algorithms** after fast startup, since there are many degrees of freedom there, too

Selected references

Evaluation results of IW10 (amongst others)

- M. Scharf. Comparison of end-to-end and network-supported fast startup congestion control schemes. Computer Networks, 2011
- **M. Scharf. Fast Startup Internet Congestion Control for Broadband Interactive Applications. PhD thesis, University of Stuttgart, submitted Nov. 2009**
- M. Scharf. Performance evaluation of fast startup congestion control schemes. In Proc. IFIP Networking 2009, LNCS 5550, Springer-Verlag, pp. 716–727, 2009
- M. Scharf. Quick-Start, Jump-Start, and other fast startup approaches: Implementation issues and performance. Presentation at 73rd IETF Meeting, ICCRG, Nov. 2008

Studies of network-supported fast startup congestion control schemes

- S. Hauger, M. Scharf, J. Kögel, and C. Suriyajan. Evaluation of router implementations of explicit congestion control schemes. Journal of Communication, vol. 5, no. 3, 2010, pp. 197-204.
- M. Scharf, M. Eissele, C. Mueller, and T. Ertl. Speeding up the 3D Web: A case for fast startup congestion control. Proc. PFLDNeT, 2009
- M. Proebster, M. Scharf, and S. Hauger. Performance comparison of router assisted congestion control protocols: XCP vs. RCP. Proc. 2nd International Workshop on the Evaluation of Quality of Service through Simulation in the Future Internet, 2009
- M. Scharf and H. Strotbek. Performance evaluation of Quick-Start TCP with a Linux kernel implementation. Proc. IFIP Networking 2008, LNCS 4982, Springer-Verlag, pp. 703–714, 2008
- S. Hauger, M. Scharf, J. Kögel, and C. Suriyajan. Quick-Start and XCP on a network processor: Implementation issues and performance evaluation. Proc. IEEE HPSR 2008, 2008.
- M. Scharf, S. Hauger, and J. Kögel. Quick-Start TCP: From theory to practice. Proc. PFLDnet, 2008
- M. Scharf. Performance analysis of the Quick-Start TCP extension. Proc. IEEE Broadnets, 2007