Performance analysis for TCP BBR in Mininet

Massimo Girondi

February 12, 2019

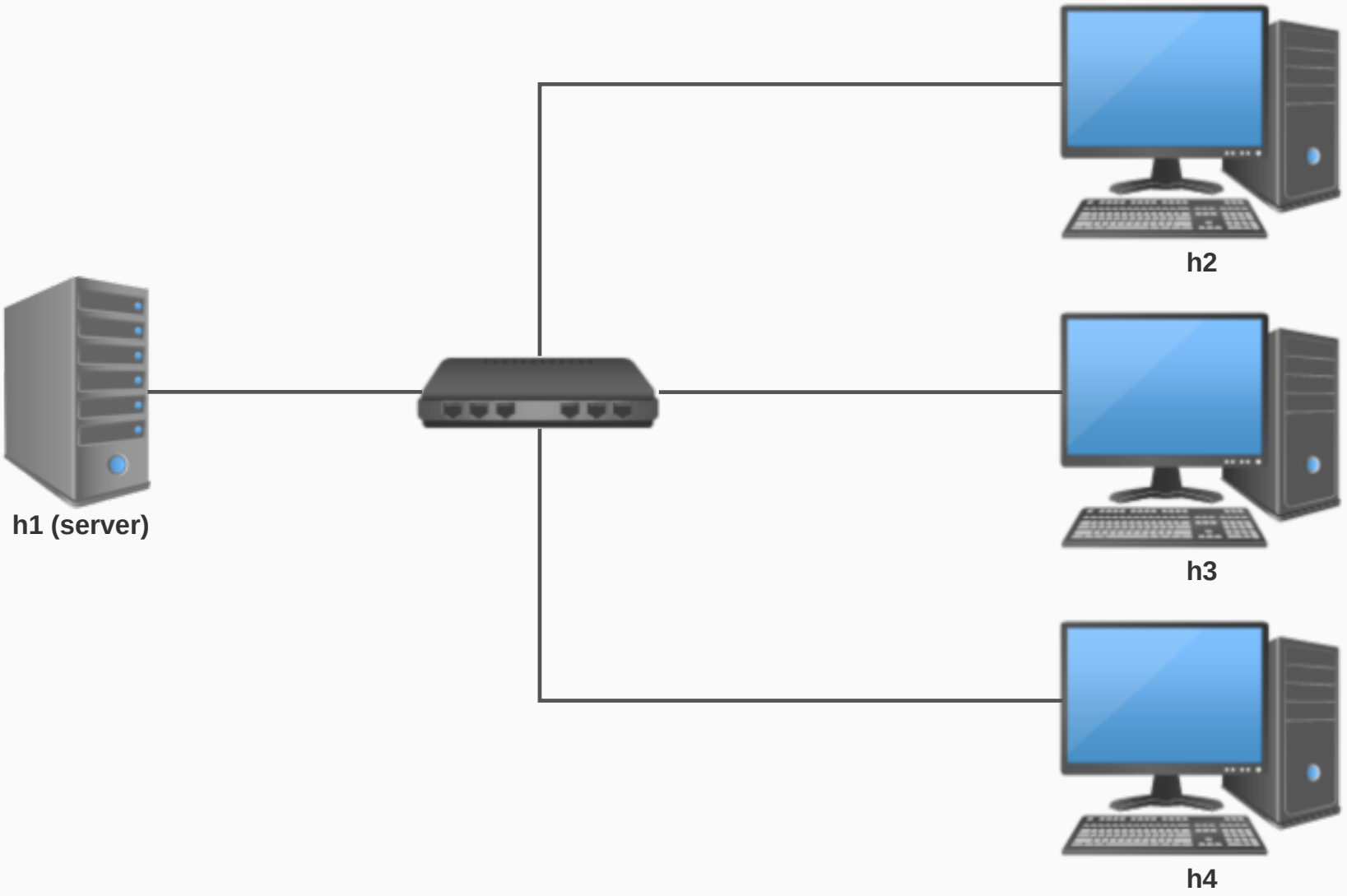
TCP BBR (from English Wikipedia)

- **B**ottleneck **B**andwidth and **R**ound-trip propagation time
- Developed by Google in 2016
- OpenSource
- For YouTube, BBR yielded an average of 4% higher network throughput (up to 14%)
- Available in Linux Kernel from version 4.9
- Seems not too fair to non-BBR streams (cit. Geoff Huston)

Test environment

- Mininet 2.3 on Debian 9
- Virtual machine running on VirtualBox
- Provisioned through Vagrant
- Switched star network, Gigabit links with 5ms delay
- 4 machines, only two used for tests
- Congestion protocol changes for every machine: kernel is shared

Topology



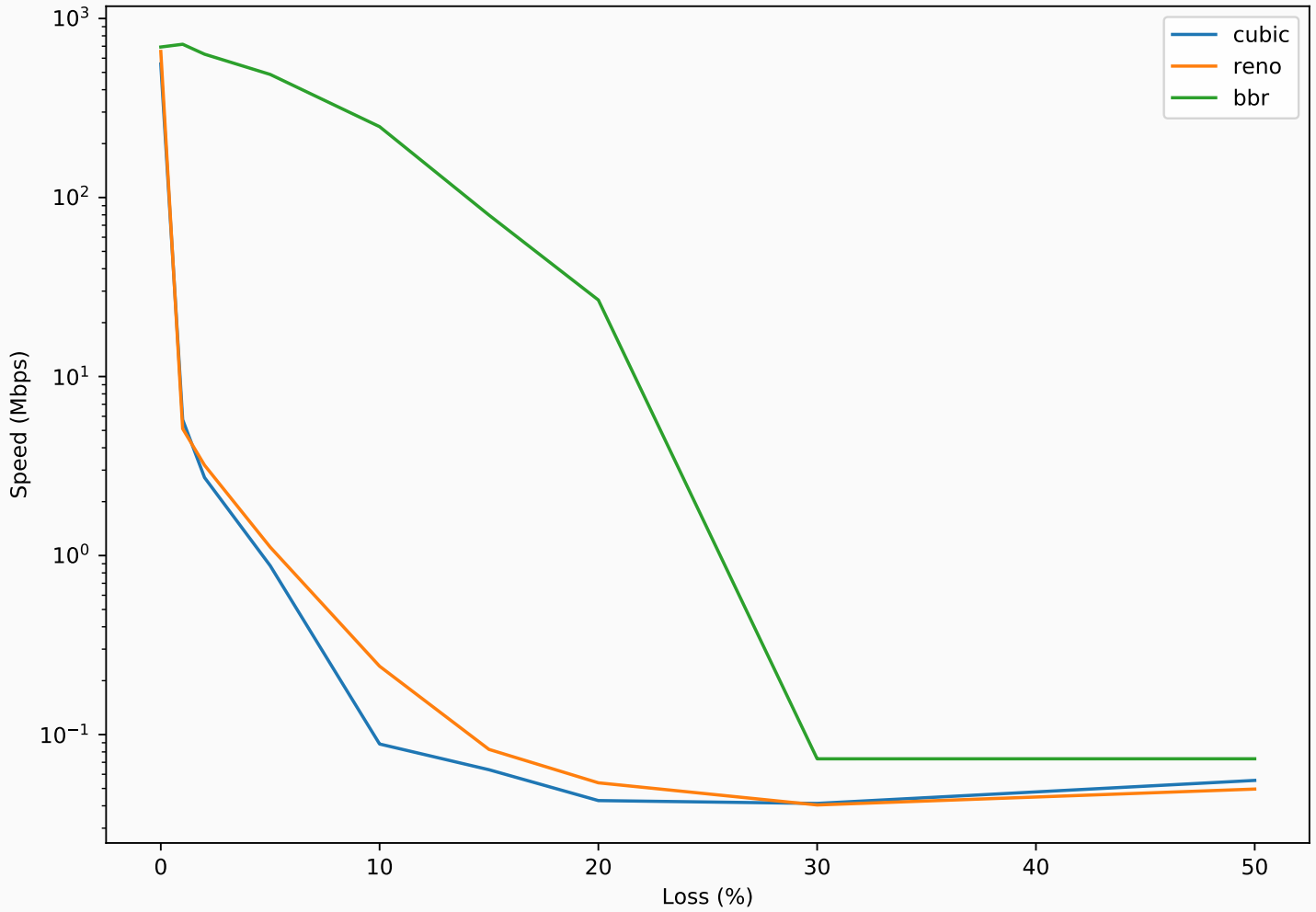
20 seconds Iperf test: description [1/3]

- Variable loss rate on the server's link, then variable delay
- `h1` as server
- `h2` as client
- Client options: `-f k -c IP -t 20`

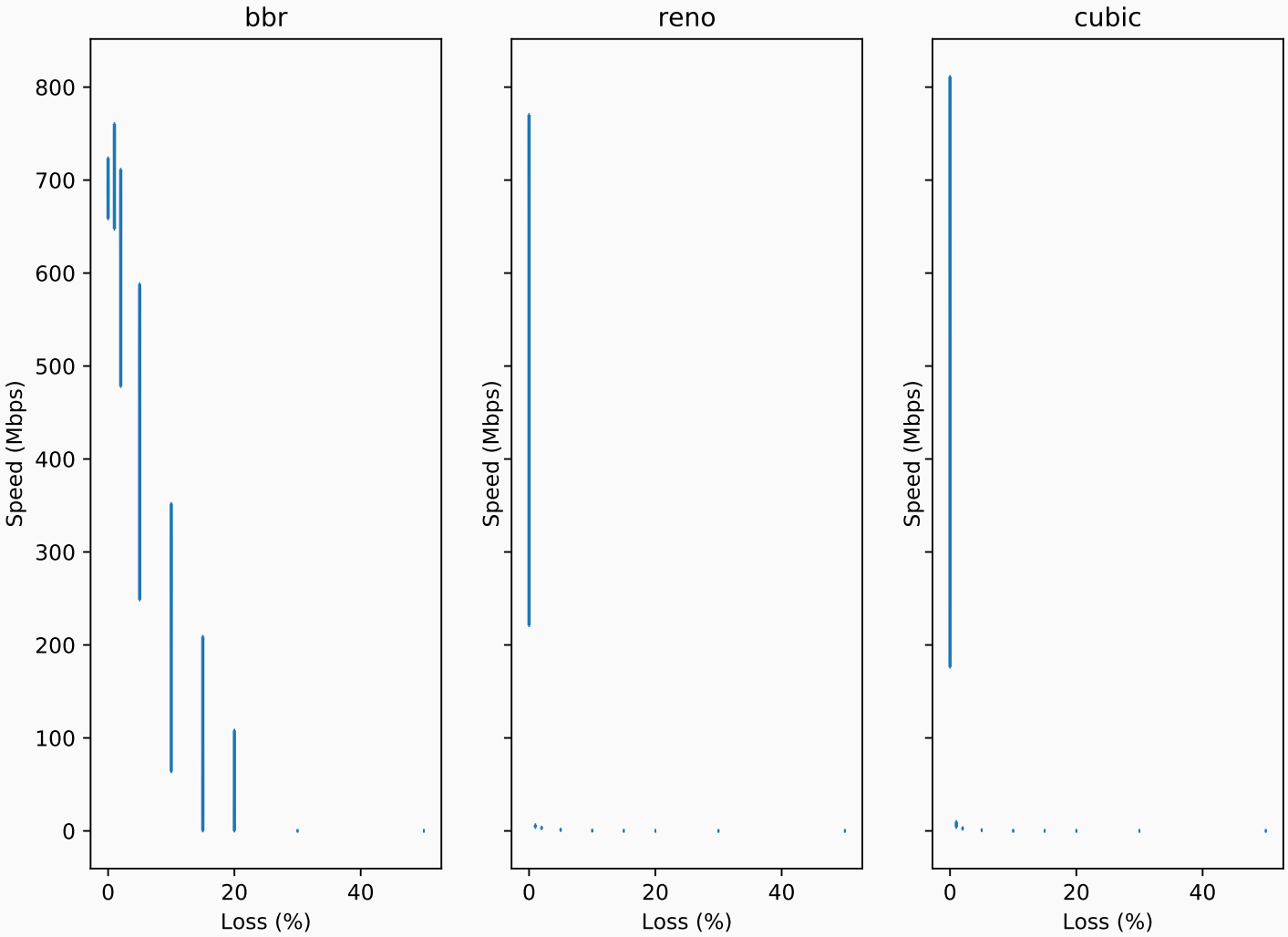
  -f k kbps as output format

  -t 20 Run for 20 seconds

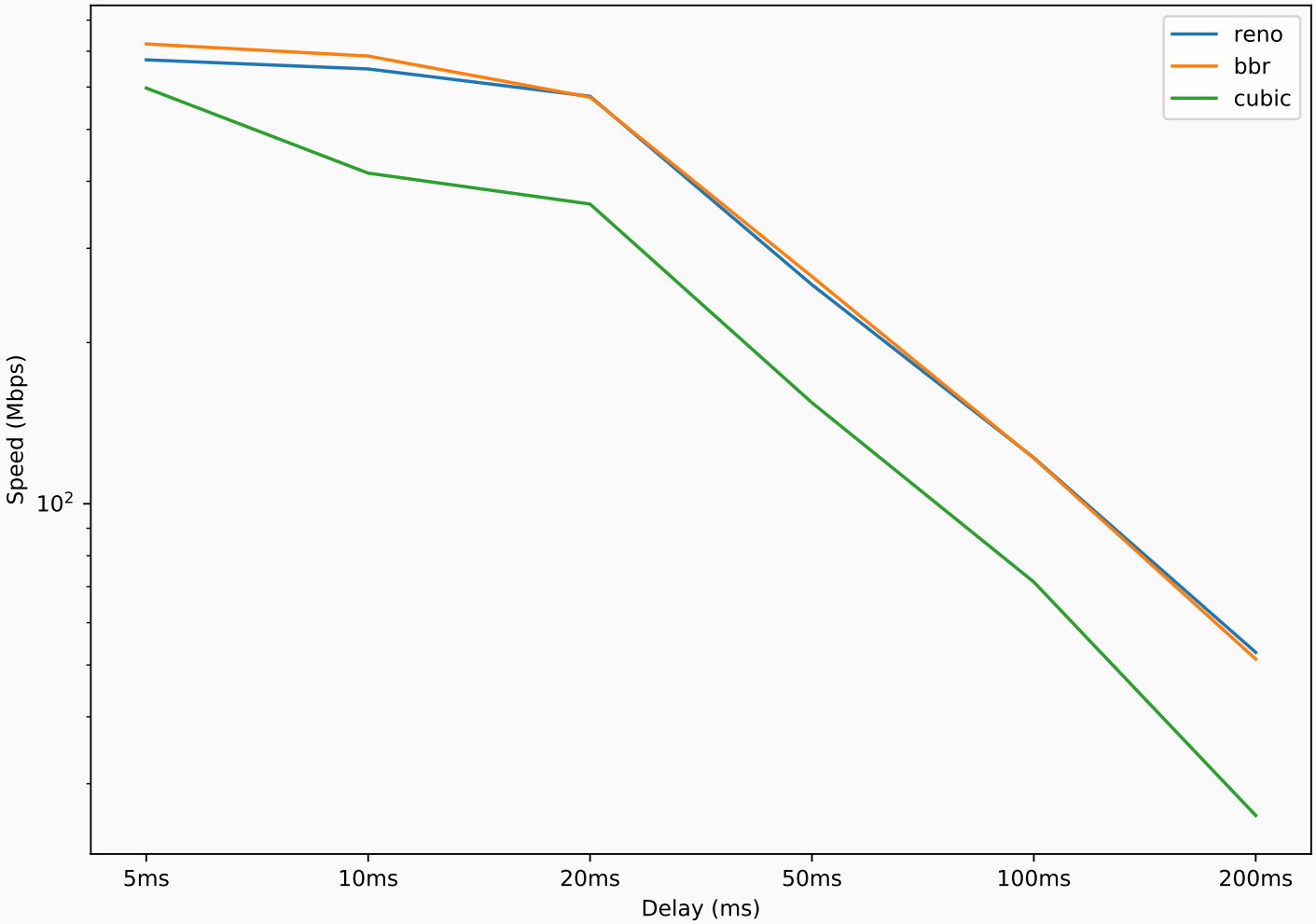
20 seconds Iperf test: speed vs loss (log) [2/5]



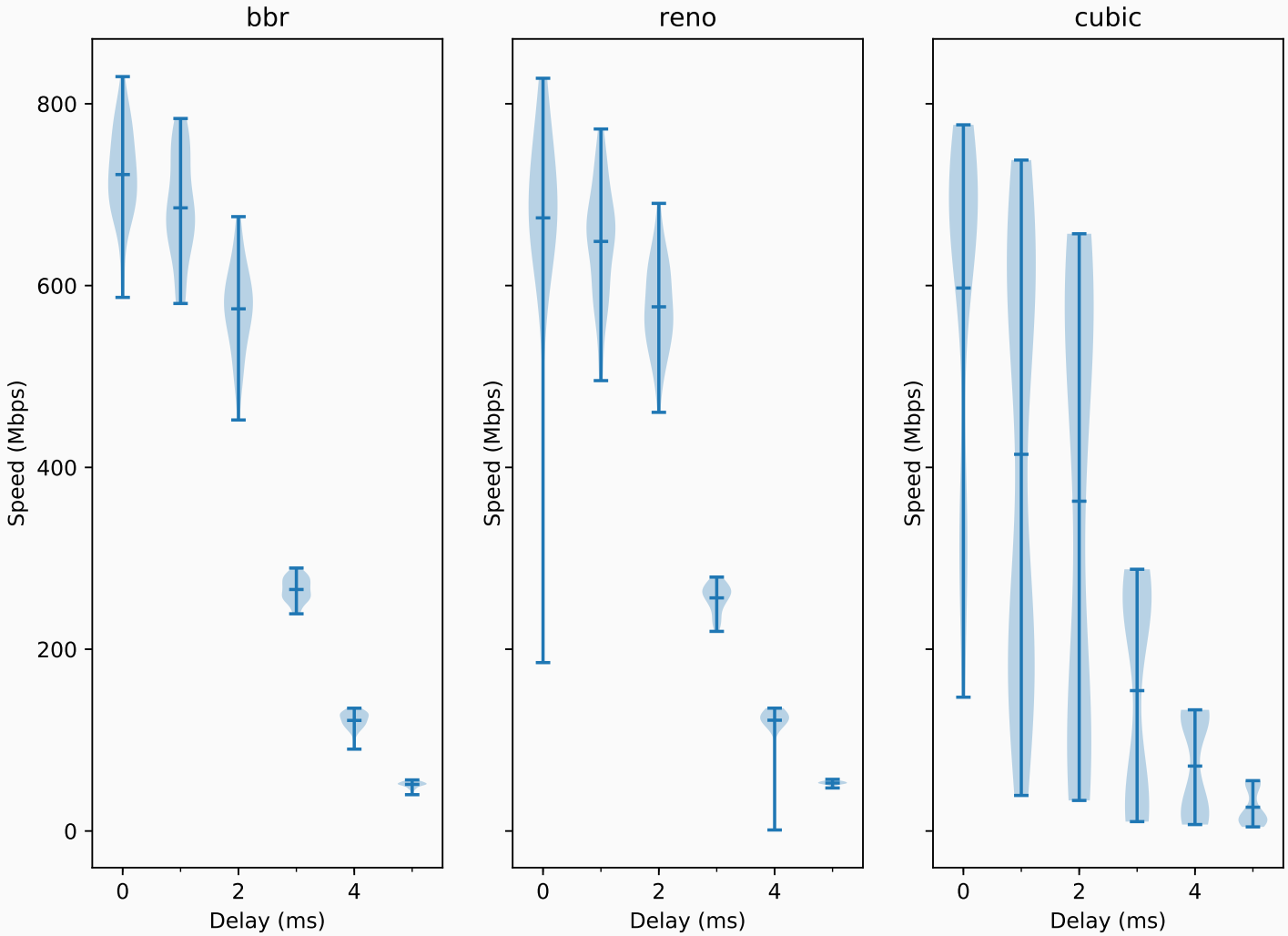
20 seconds Iperf test: speed vs loss distribution [3/5]



20 seconds Iperf test: speed vs delay (log) [4/5]



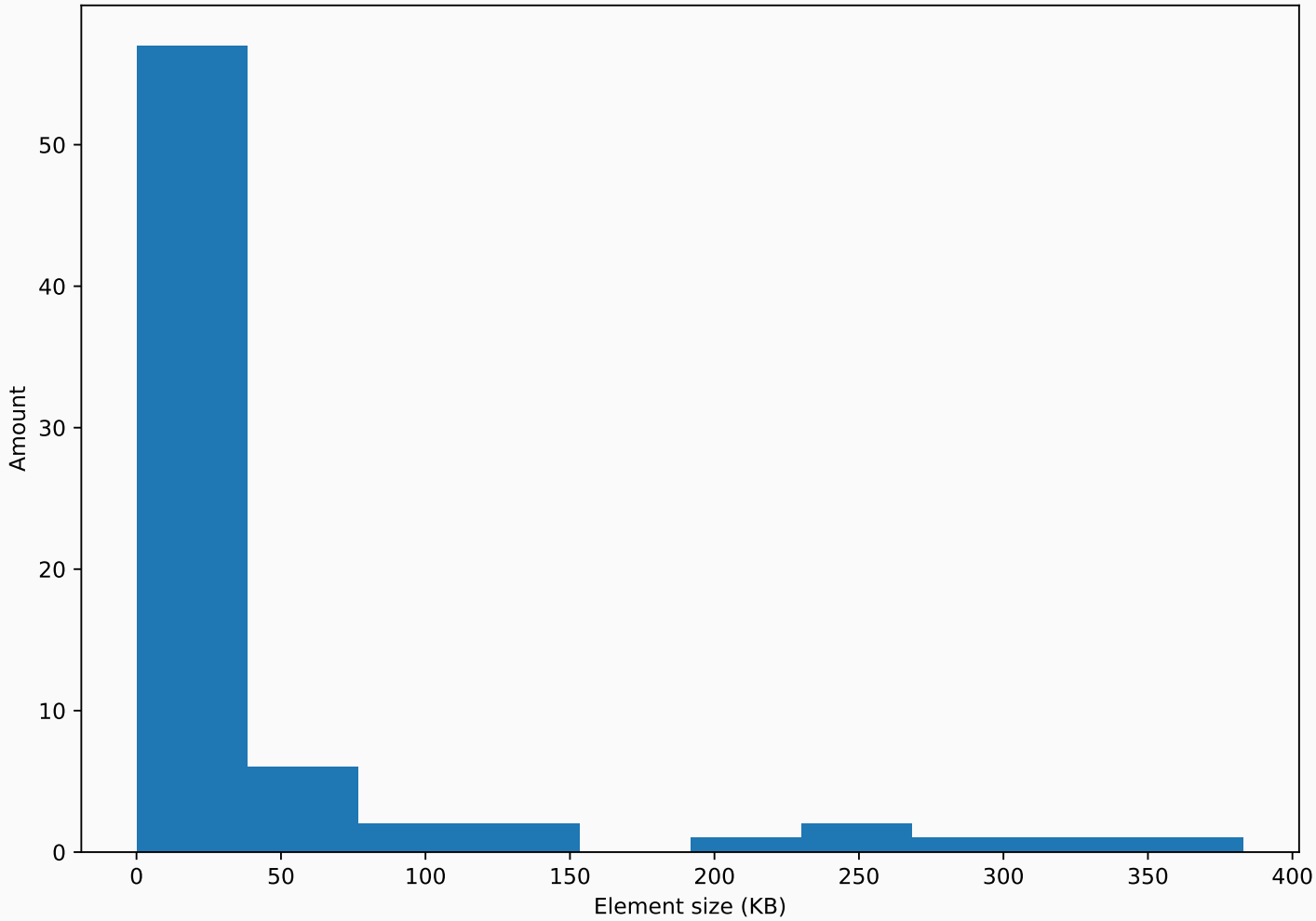
20 seconds Iperf test: speed vs delay distribution [5/5]



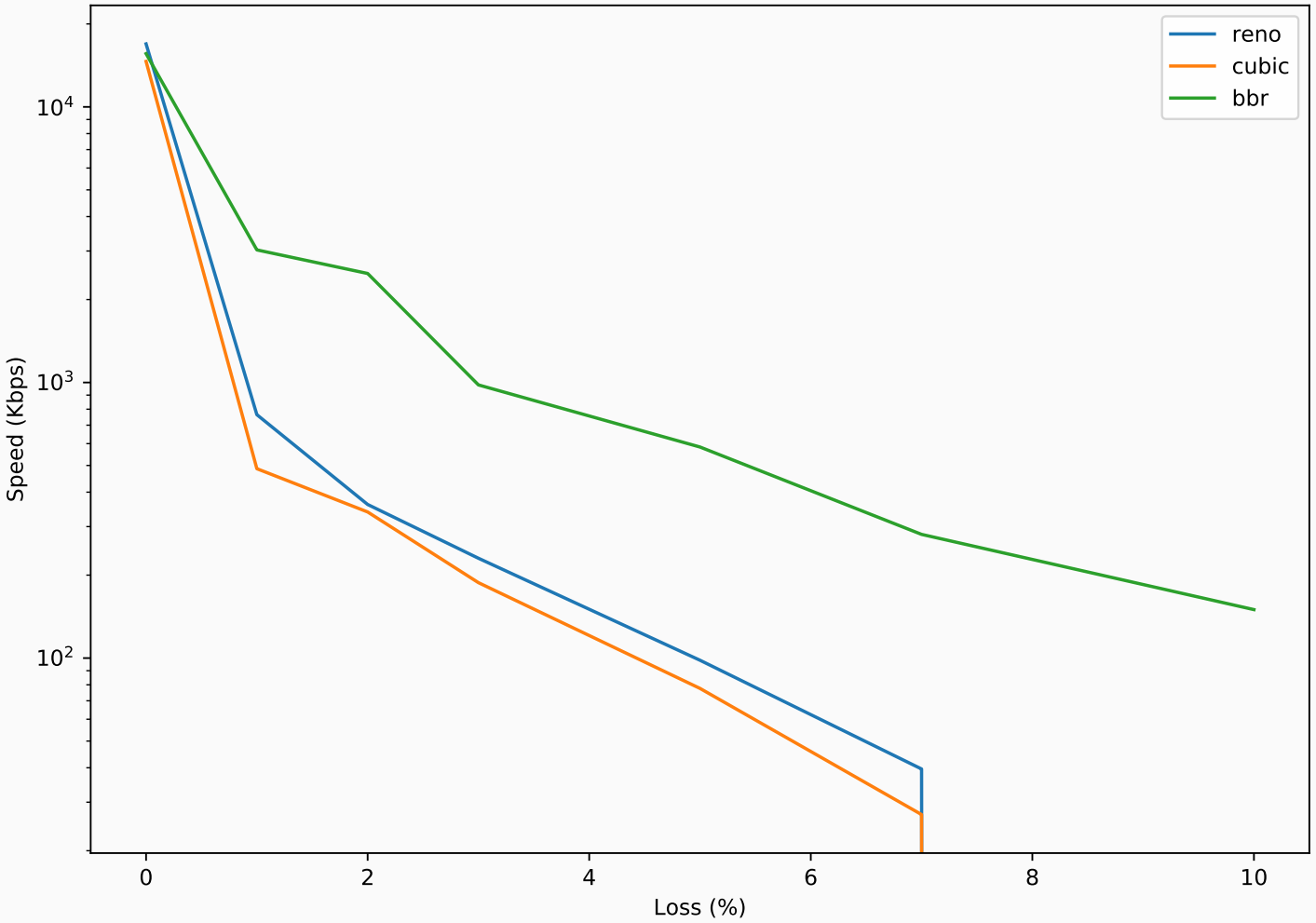
Complex web page simulation: description [1/4]

- Variable loss rate on the server's link
- `h1` as server
- `h2` as client
- `nginx` as server since Python's `SimpleHTTPServer` is too slow
- `wget -r` as client, simulating the web page browsing
- `bbc.co.uk` index page as an example: elements has same size but random data
- 72 elements with a total size of 2.902 MB
- Speed is obviously slower than a single big file download
- Elements included as images into a simple HTML file

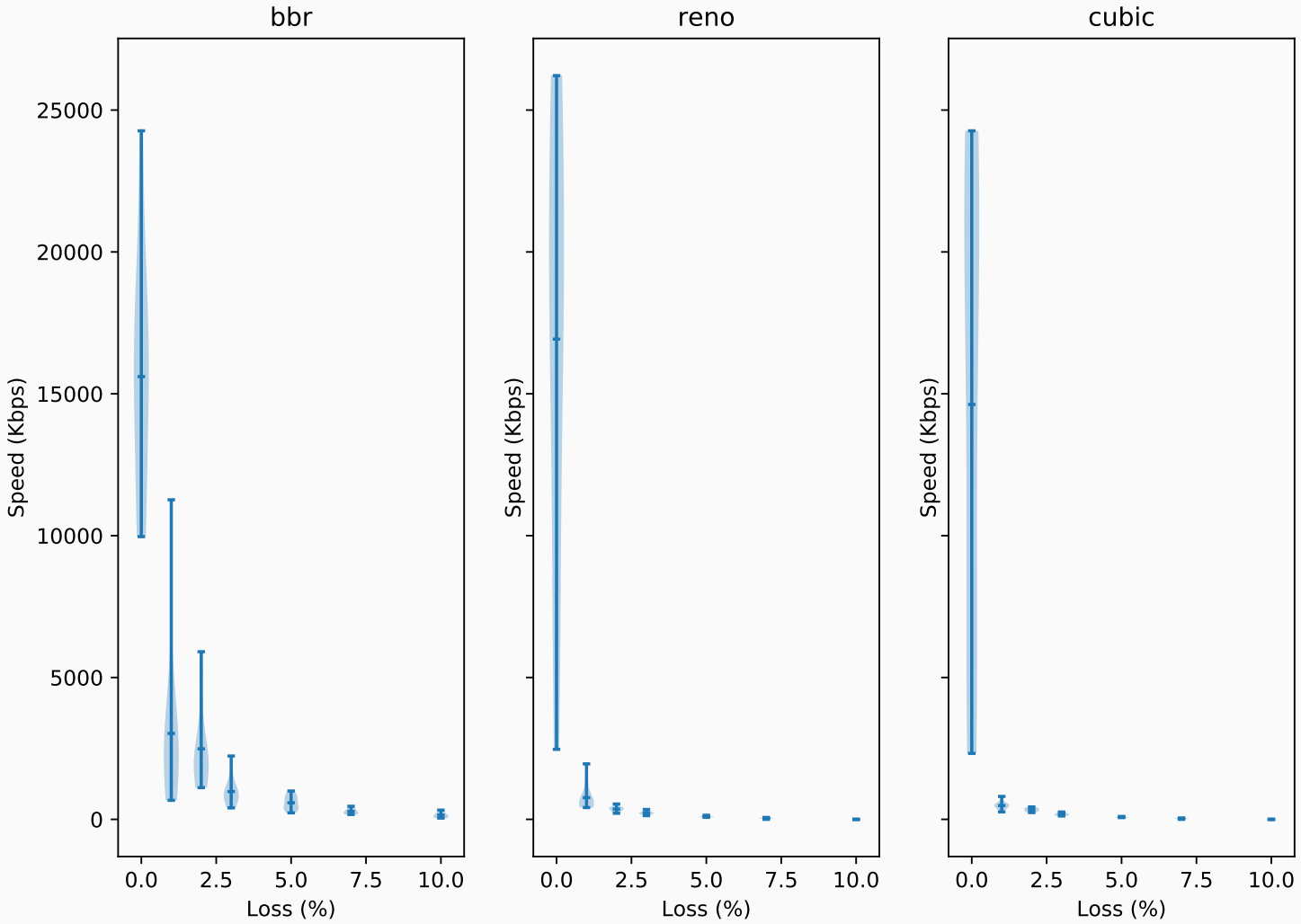
Complex web page simulation: page structure [2/4]



Complex web page simulation: speed (log) [3/4]



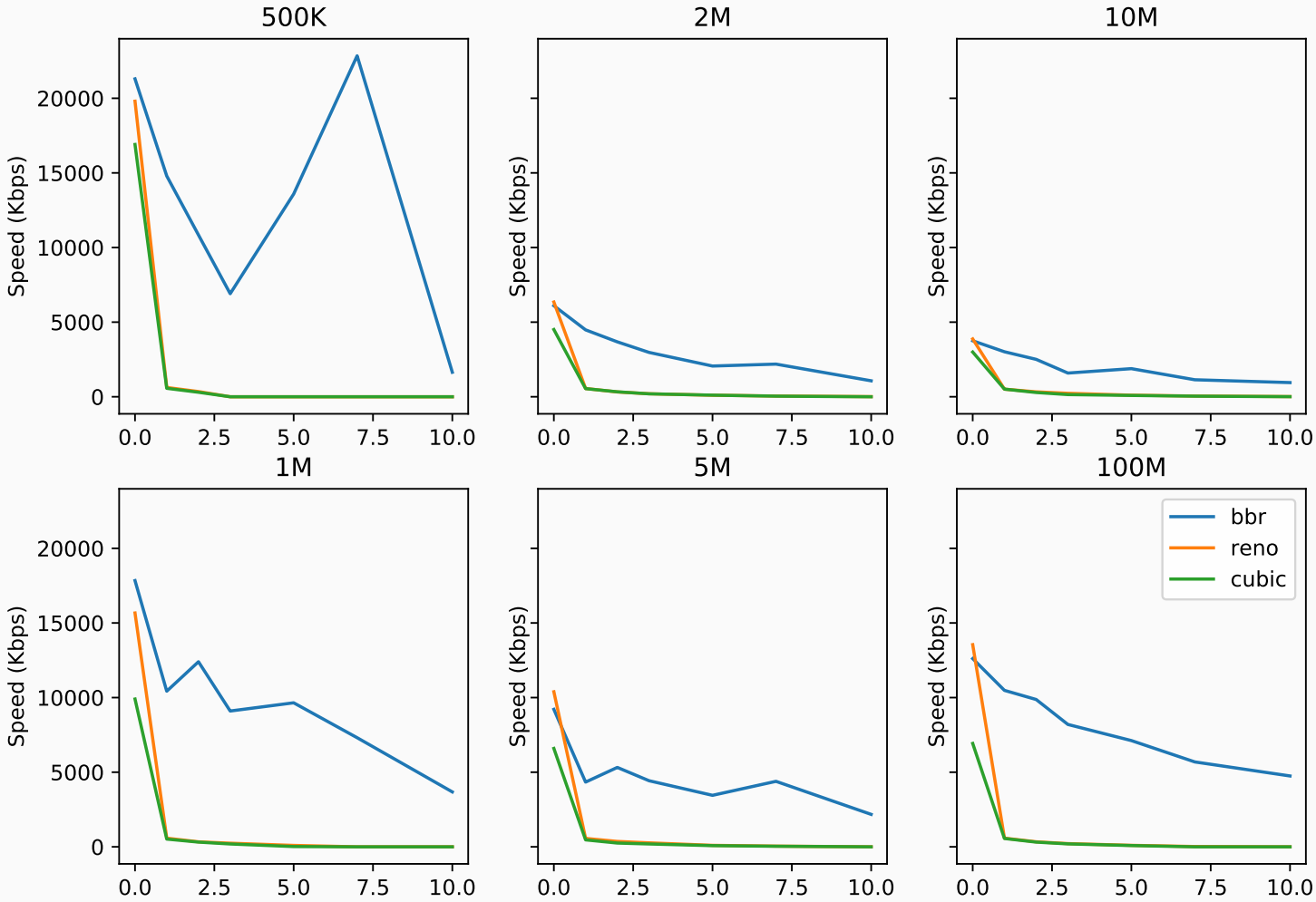
Complex web page simulation: speed distribution [4/4]



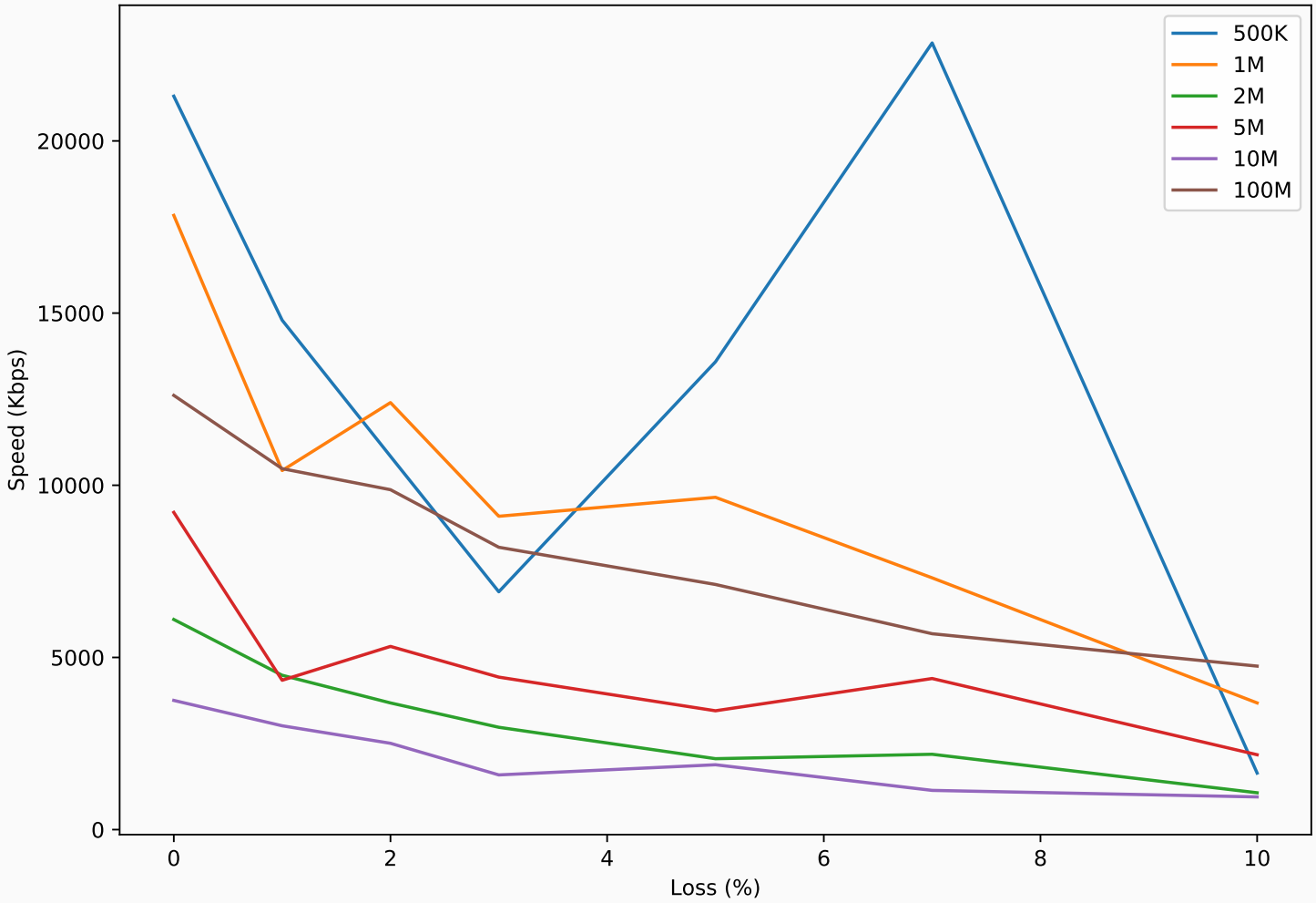
Single file download: description

- Variable loss rate on the server's link
- `h1` as server
- `h2` as client
- As before, `nginx` as server
- Simple `wget` as client, 120 seconds timeout
- 500KB, 1MB, 2MB, 5MB, 10MB and 100MB files
- Random content in files

Single file download: speed for every size and protocol [1/2]



Single file download: BBR performances [2/2]



New Reno faster than Cubic!? [1/2]

From the previous tests, it appears that New Reno is faster than Cubic in some cases. I run some test on a real network, just to verify.

- 1 Gbit links, adequate hardware
- less than 1 ms ping time
- Debian with 4.18 kernel on same machines
- Overall, New Reno appeared up to 5 Mbps faster than Cubic
- Limited tests and environment variables: longer tests should be performed
- Of course, it can depend by specific version or machine's characteristics

New Reno faster than Cubic!? [2/2]

Condition	New Reno	Cubic
No delay, no loss	947	942
100 ms delay	201	196
5% loss	944	942
10% loss	943	913

Conclusions

- Tests showed the better performances of BBR over Cubic and New Reno
- BBR achieves better speed both across loss and delays
- Congestion Window cannot be inspected without rebuilding the kernel (to get `tcp_probe`)
- Fairness to other protocols needs to be investigated using multiple machines (the kernel is shared)
- All the code is available on `github.com/MassimoGirondi/DNCS_BBR`.