

CCRG ISP B Term

-- Week 7 / Final Week Report

Yongcheng Liu, Mingxi Liu,
Mona Ma, Zachary Porter

Agenda

- Data Collection - Yongcheng
 - Trial Setup
 - New Trial System
- Data Analysis - Mona, Mingxi
 - Single trial
 - Best & worst round for each congestion control algorithms
 - Throughput
 - Average over time, Boxplot, RTT/Lose over time, CDF of mean
 - Throughput during steady state
 - Average over time, Boxplot, RTT/Lose over time, CDF of mean
 - Average difference & standard deviation

Data Collection – Test Setup

- Trials:
 - 1 on weekday (Dec 7)
 - 24 hours
 - Main Experiment Loop:
 - 47 rounds
 - 4 TCP Flows (Cubic, BBR, Hybla, PCC)
 - 2 Proxy Status (on, off)
 - = 376 Data Sets
 - Raw Data Dir: /home/yliu31/clayTrials/WEEKEND_TEST_2 (Though in fact it is experimented at weekday)

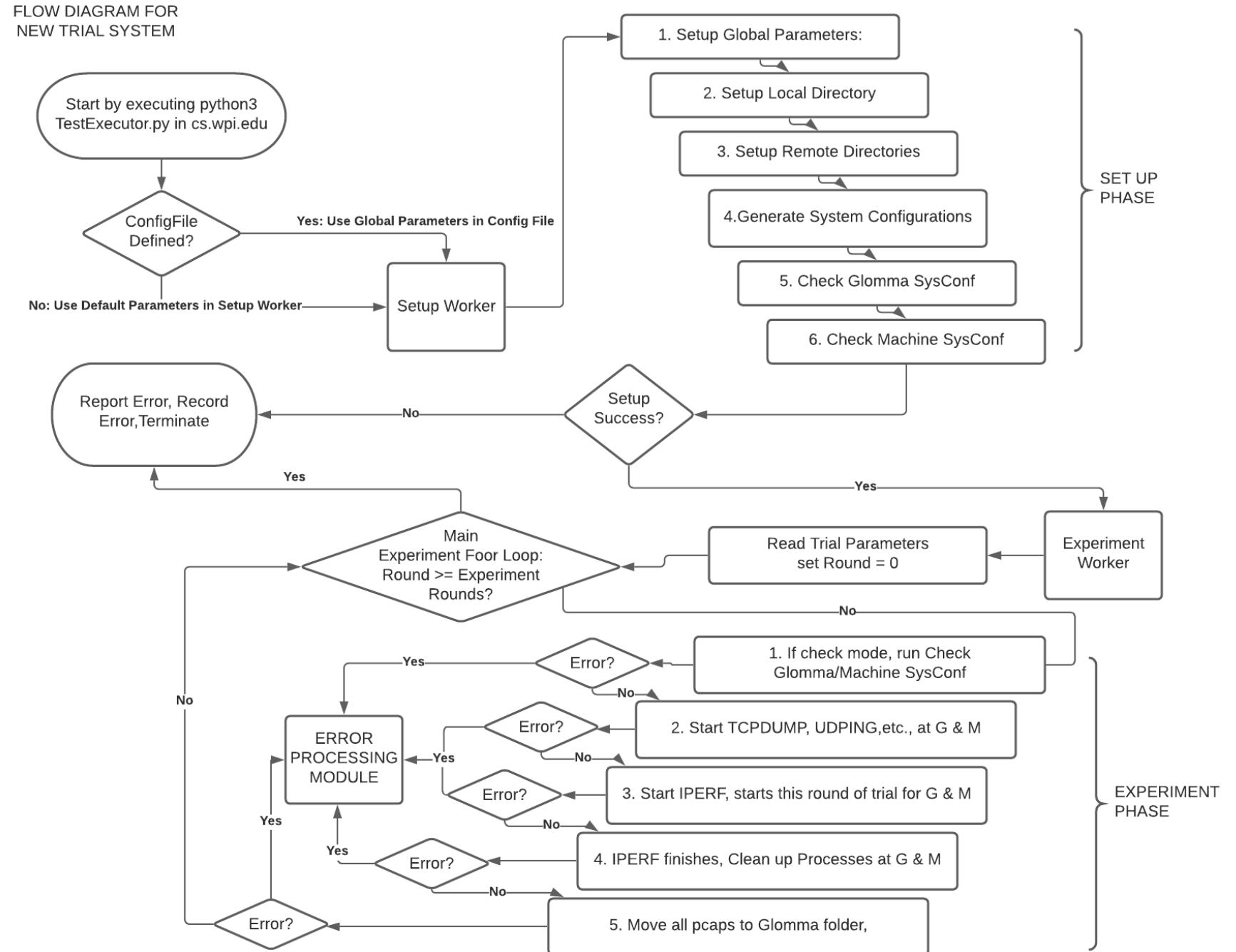
Data Collection – New Trial System

- Old Trials:
 - Discarded lots of tests, due to
 - 1. Glomma Resetting Issues. After glomma resets, glomma's ip may change. Thus, due to the firewall rule, Glomma may fail to scp/ssh to MLCs. Also, we need to send our key files to satellite server after reset.
 - 2. Route Resetting Issues (Collisions?). While we are running our experiments, for some reasons our routes reset many times. Even in some trials, it happened in the middle of the trial.
 - Other Issues:
 - SSH Overheads: We tried with 50M tests, which will cost around 1 minute per trial on 1 machine. Majorly caused by using ssh to issue commands at MLCs.
 - No Auto Error Detection and Correction:
 - In many times, we had to discard the whole test if any error occurred. Also, we can only know errors might happened after data is analyzed in old settings.
 - Like SSH issues in the middle of the test. Test will not auto-terminate or auto-correct

Data Collection – New Trial System

- New Trial System:
 - We proposed a new trial system to solve found issues
 - Features:
 - Auto checking, before the experiment and during the experiment
 - Auto correcting for some checks
 - Lower SSH overheads
 - Extensible

New Trial System Flow Diagram



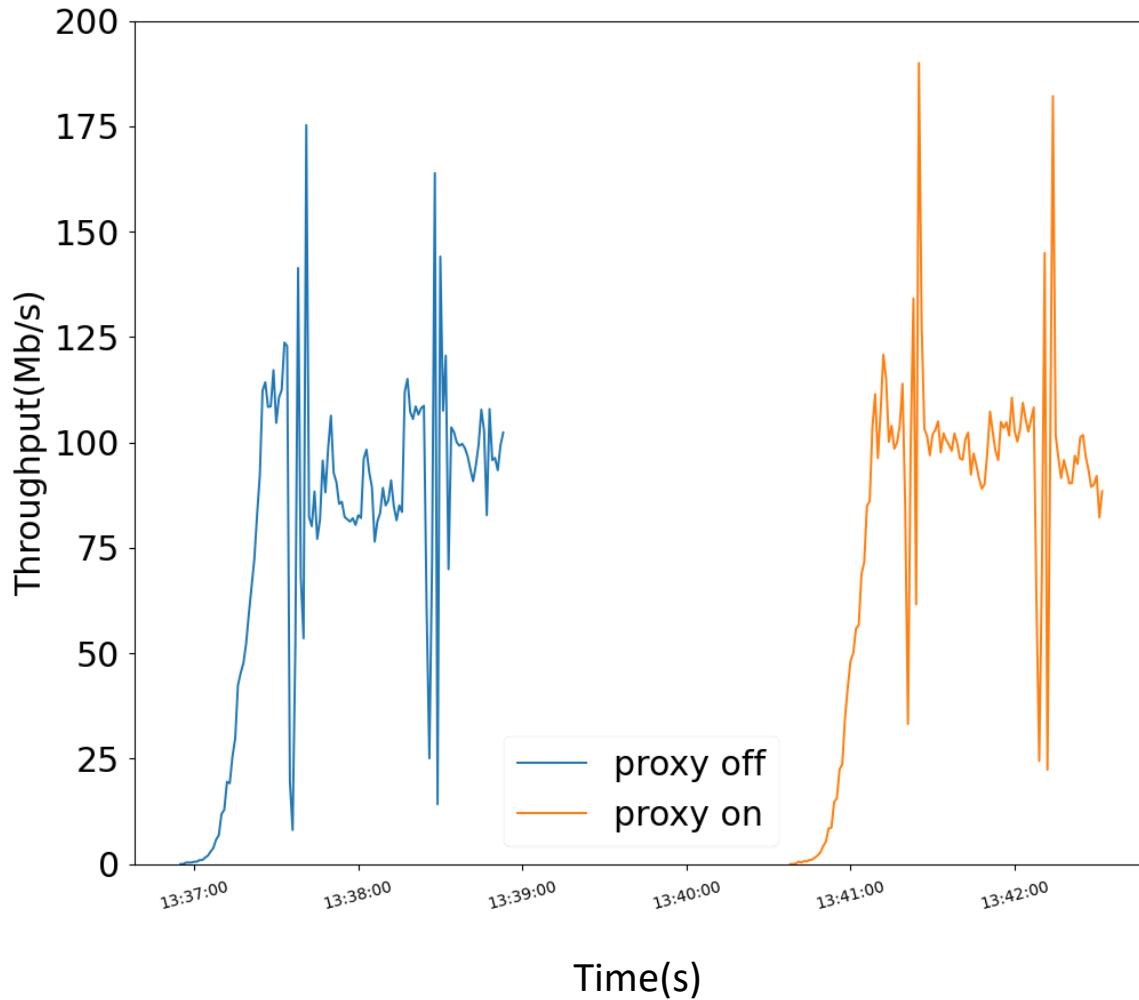
New Trial System Structure – Error Detection And Handling

- Configuration Based Checking
 - Checks if system field equals those in configuration files
 - Glomma:
 - [IP] section: MLC IPs
 - MLCs:
 - [TCP] section: tcp_mem, wmem, rmem, protocol
 - [IP] section: Glomma IP (get at cs server)
- SSH Based Checking
 - Connection Testing:
 - Basically call “ssh host ‘echo ‘TEST’ or ‘\$HOME’ ’”. Verify by comparing output from ssh calls
 - In Experiment, if “ssh: / scp: ... err” received at Glomma, report.
 - In Experiment, cs server use screens to TCPDUMP, etc.
 - If screens closed before it should (cannot find screen instance at cleanup()), report.
 - In Experiment, if “scp pcap < 1KB”, report

New Trial System Structure – Error Detection And Handling

- ConfigChecker:
 - @TEST Decorator, and will run all test methods with @TEST in defined CheckerClass
 - Return False if any error, else True. If –message flag, return error method names also.
 - Glomma Checks:
 - SSH – Connection to satellite and permission? -> If no, in output report this and terminates
 - ip route show – [IP] section exists (specific to eno2 via 192.168.1.1)? -> If not add missing one
 - SSH Connection Reset By Peer -> Ask CS to check Machine Conf
 - Machine Checks:
 - [IP] section Glomma ip in hosts.allow?
 - Check and set [TCP] section
- Working:
 - Move all error handling to a module in cs server / executor server (not implemented yet). On error, report an error code if knows type, else error message.

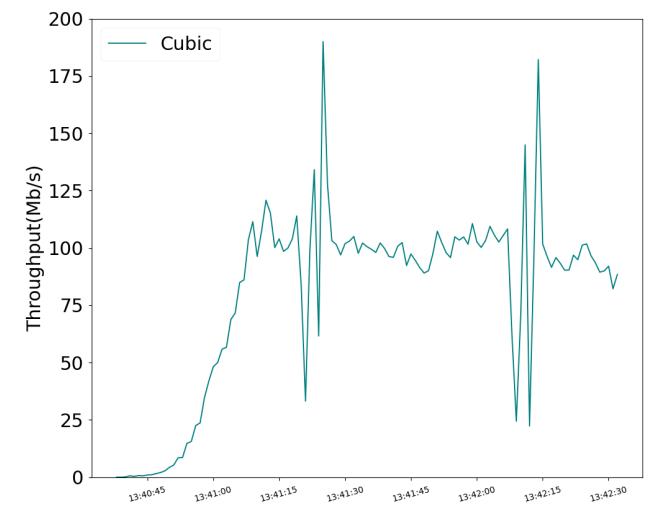
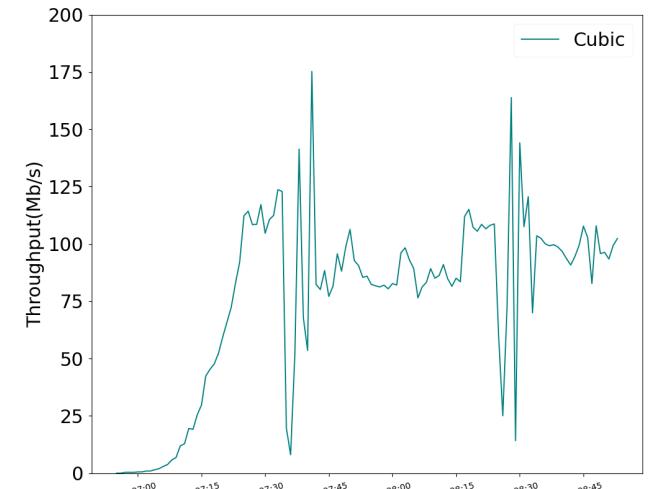
Single trial setup



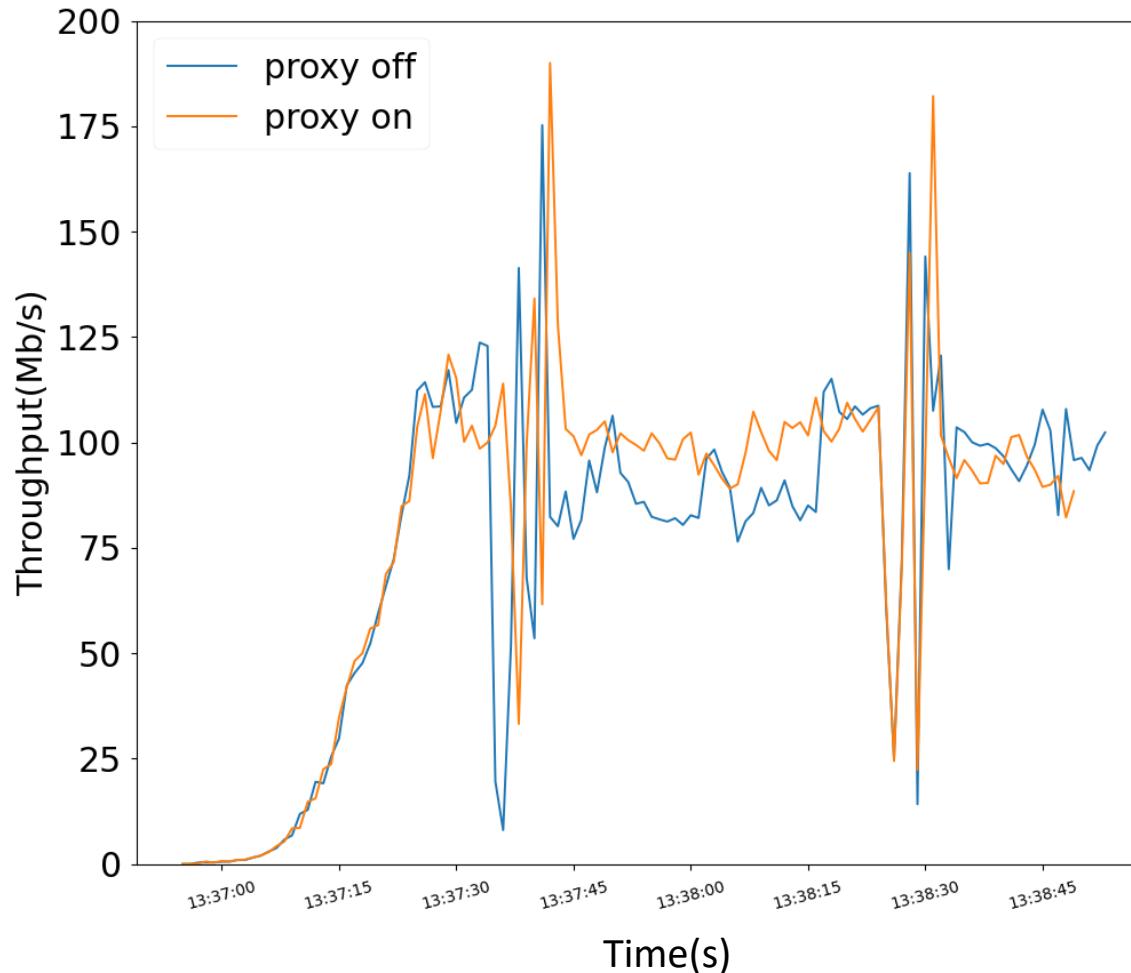
Proxy off



Proxy on



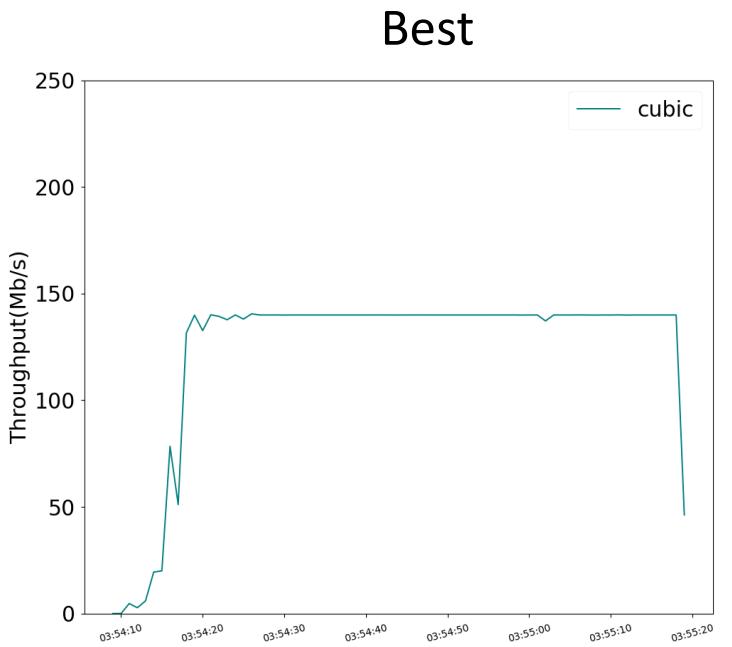
Direct comparison



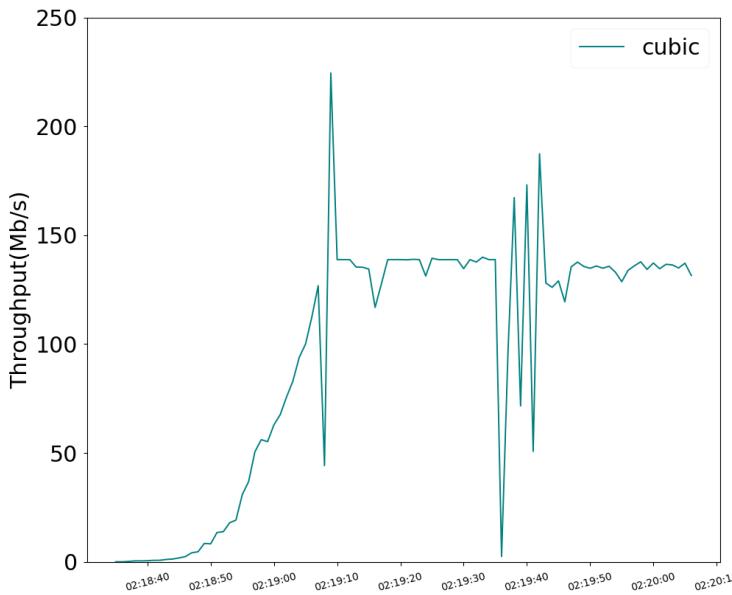
cubic	proxy off	proxy on
average	75.6806083	79.4155727
standard dev	40.1686941	40.9763953
max	175.26592	190.023936
median	85.432	96.274624

Cubic

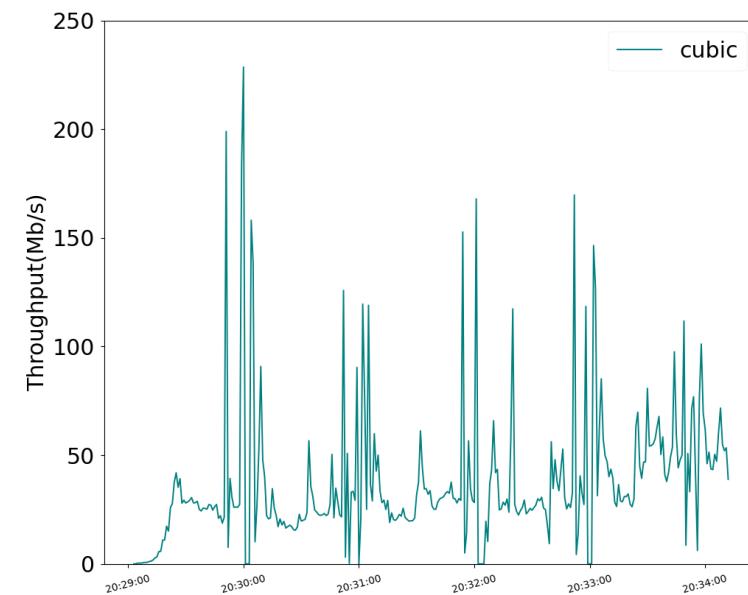
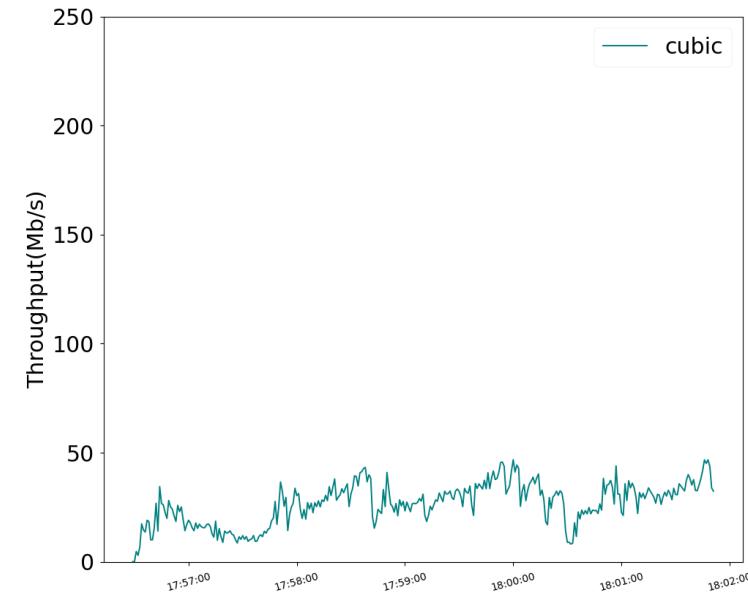
Proxy on



Proxy off



Worst

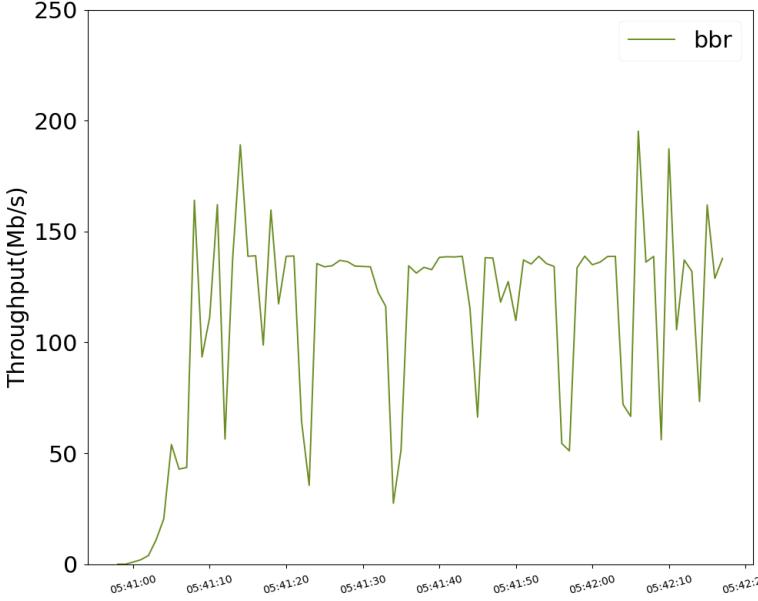


BBR

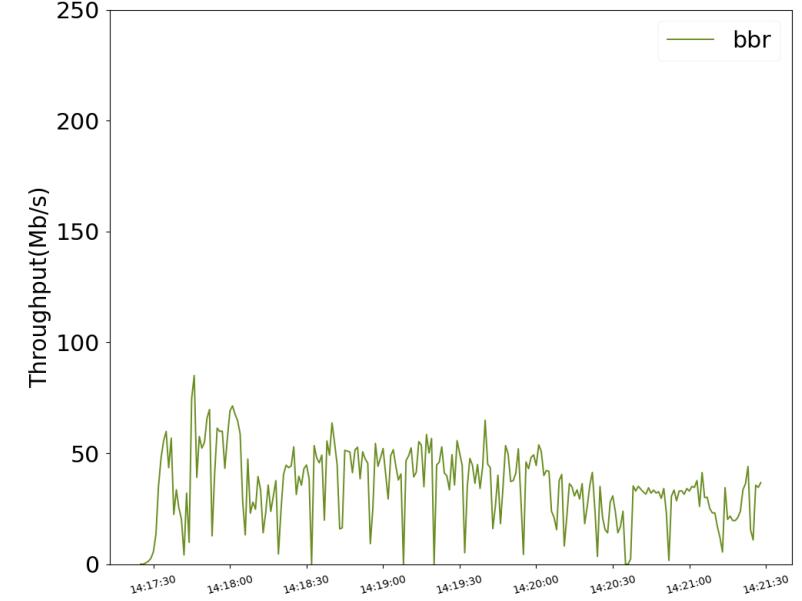
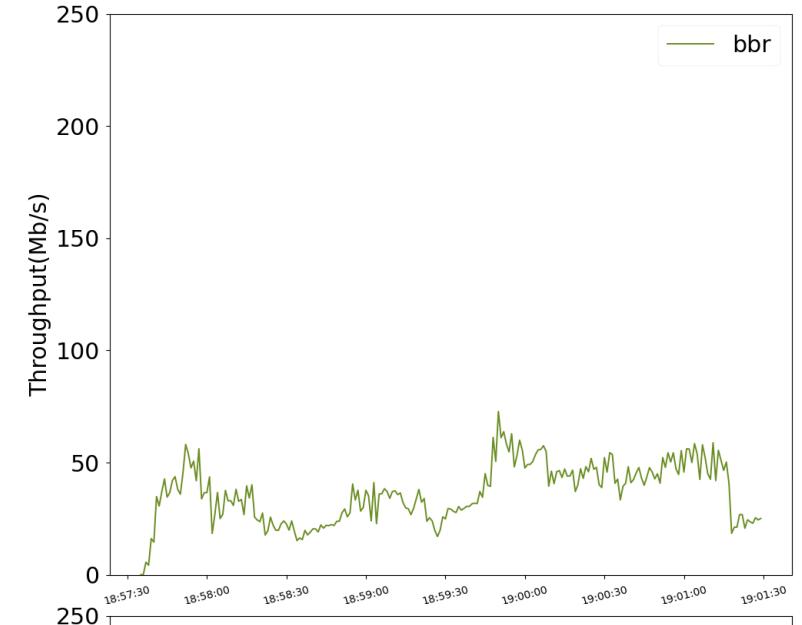
Proxy on



Proxy off

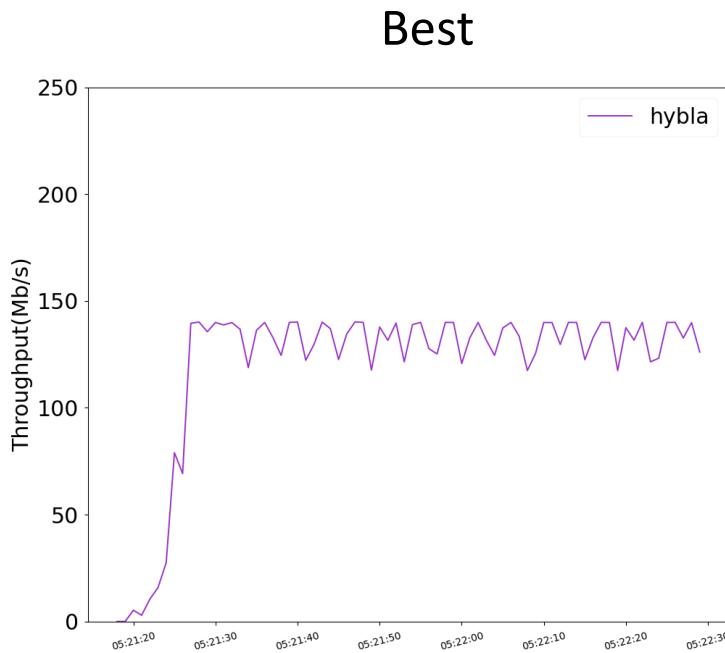


Worst

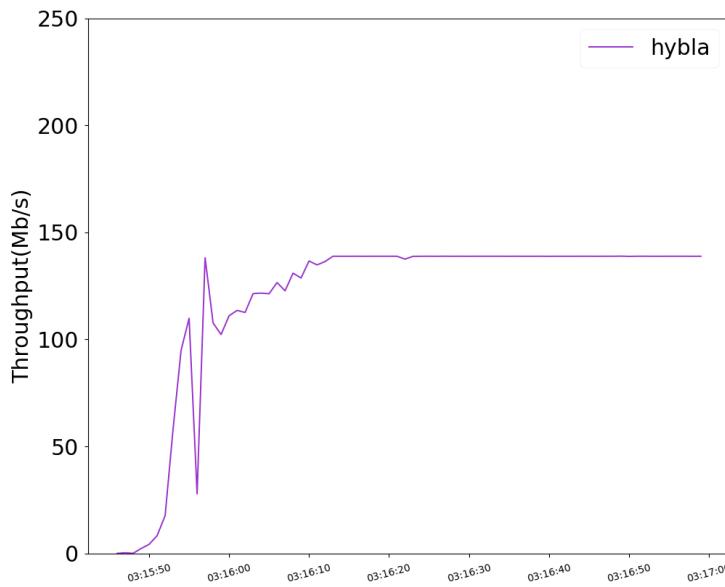


Hybla

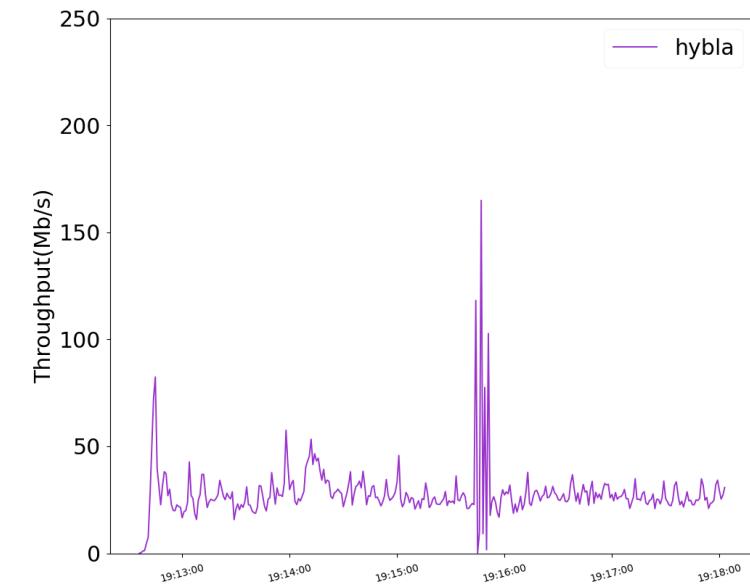
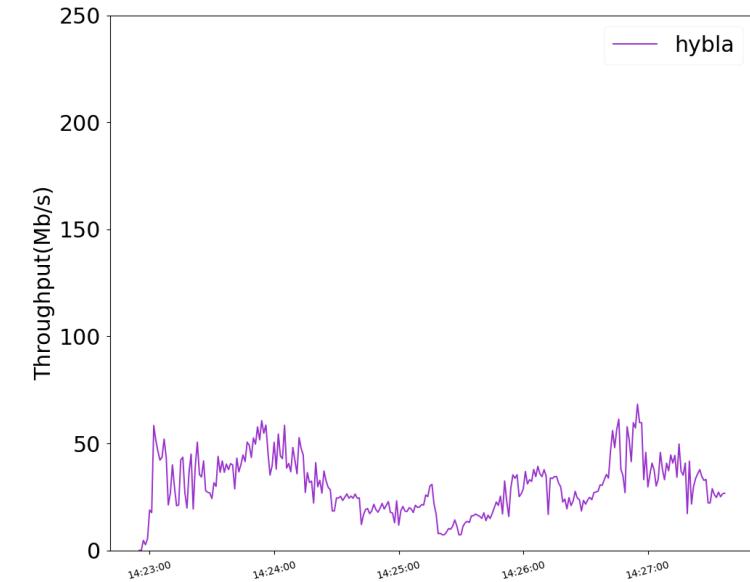
Proxy on



Proxy off

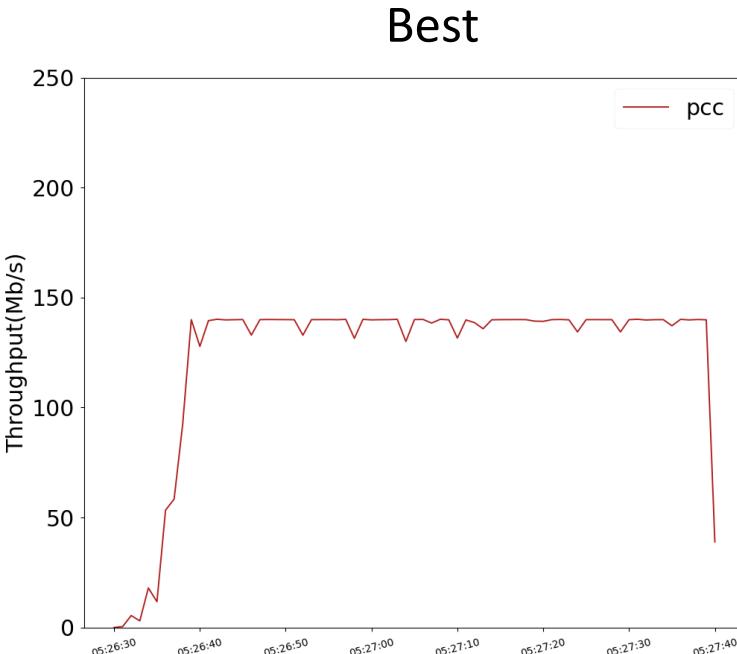


Worst

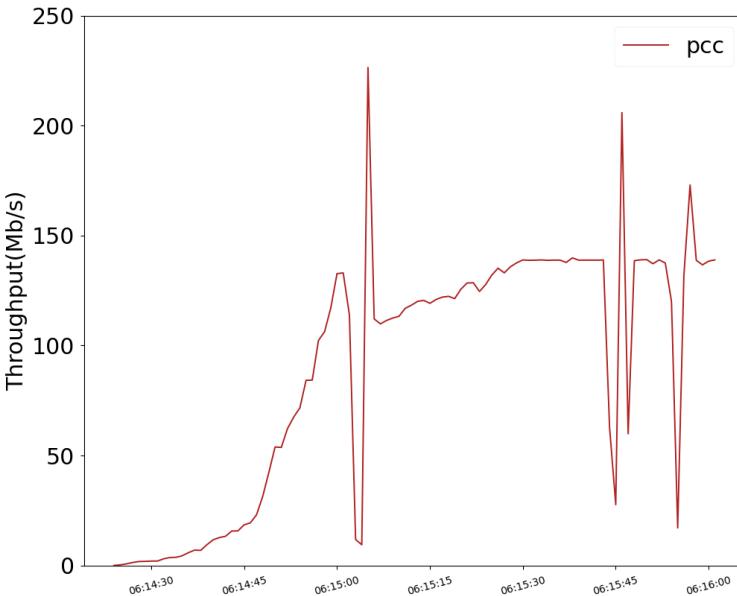


PCC

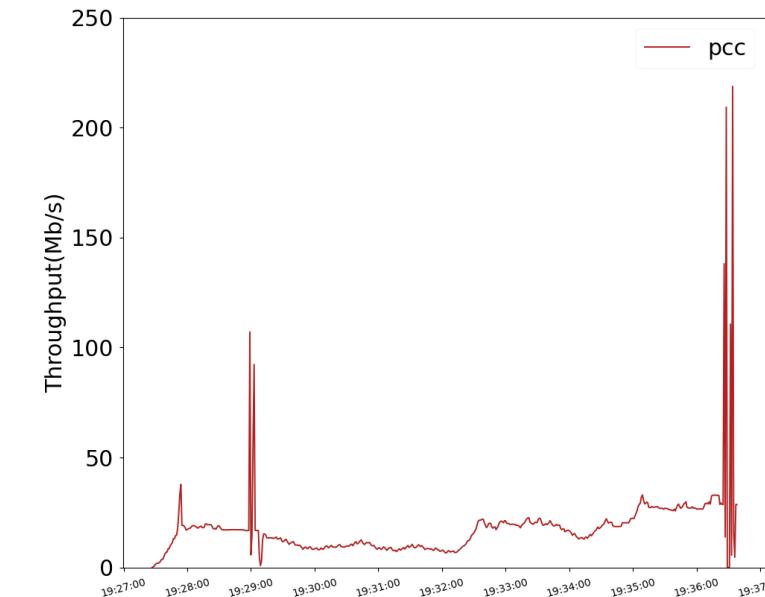
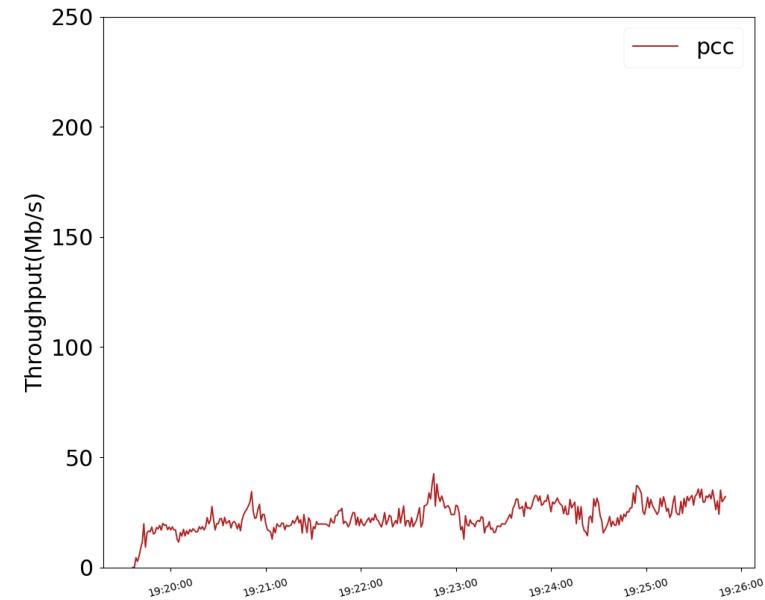
Proxy on



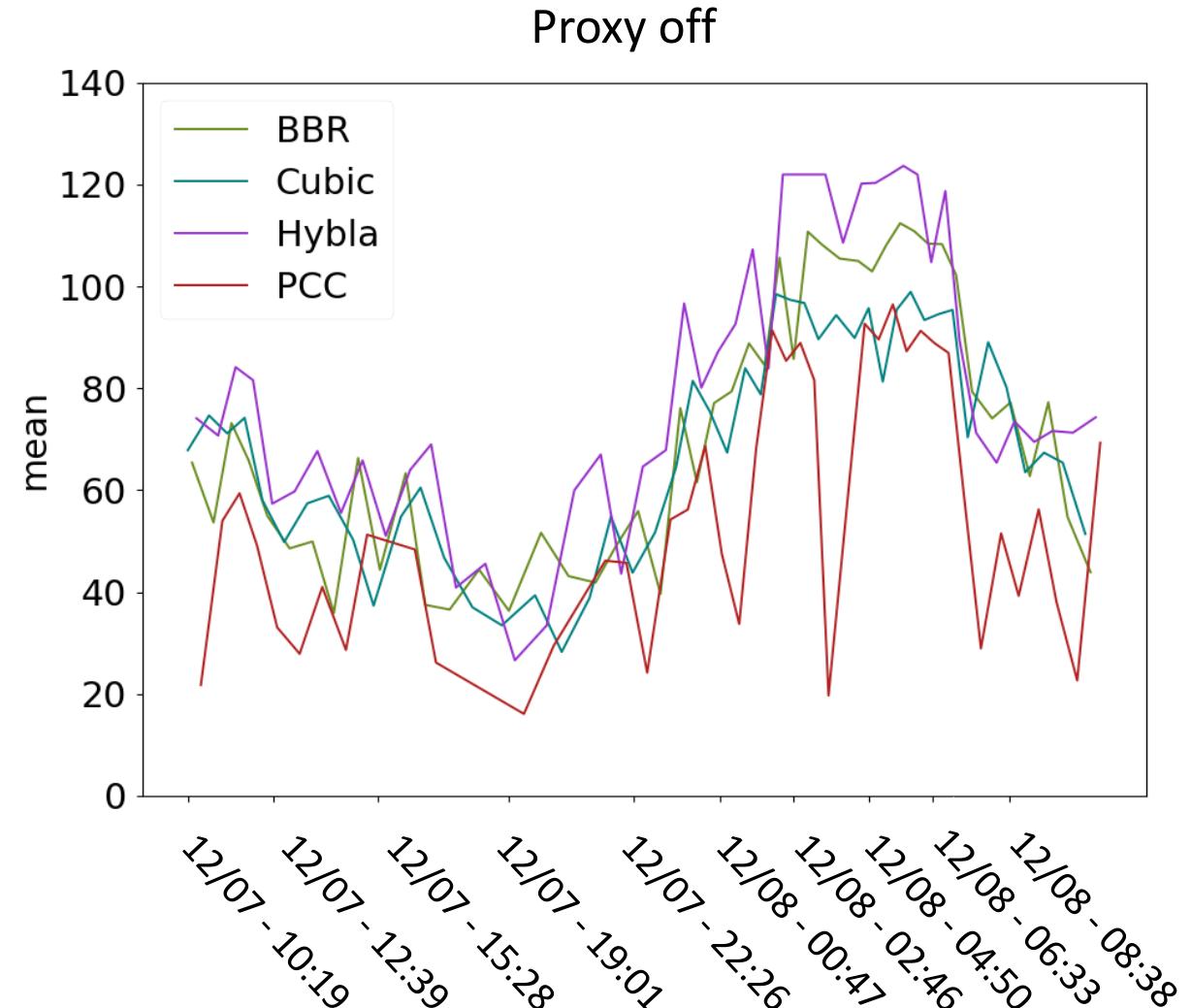
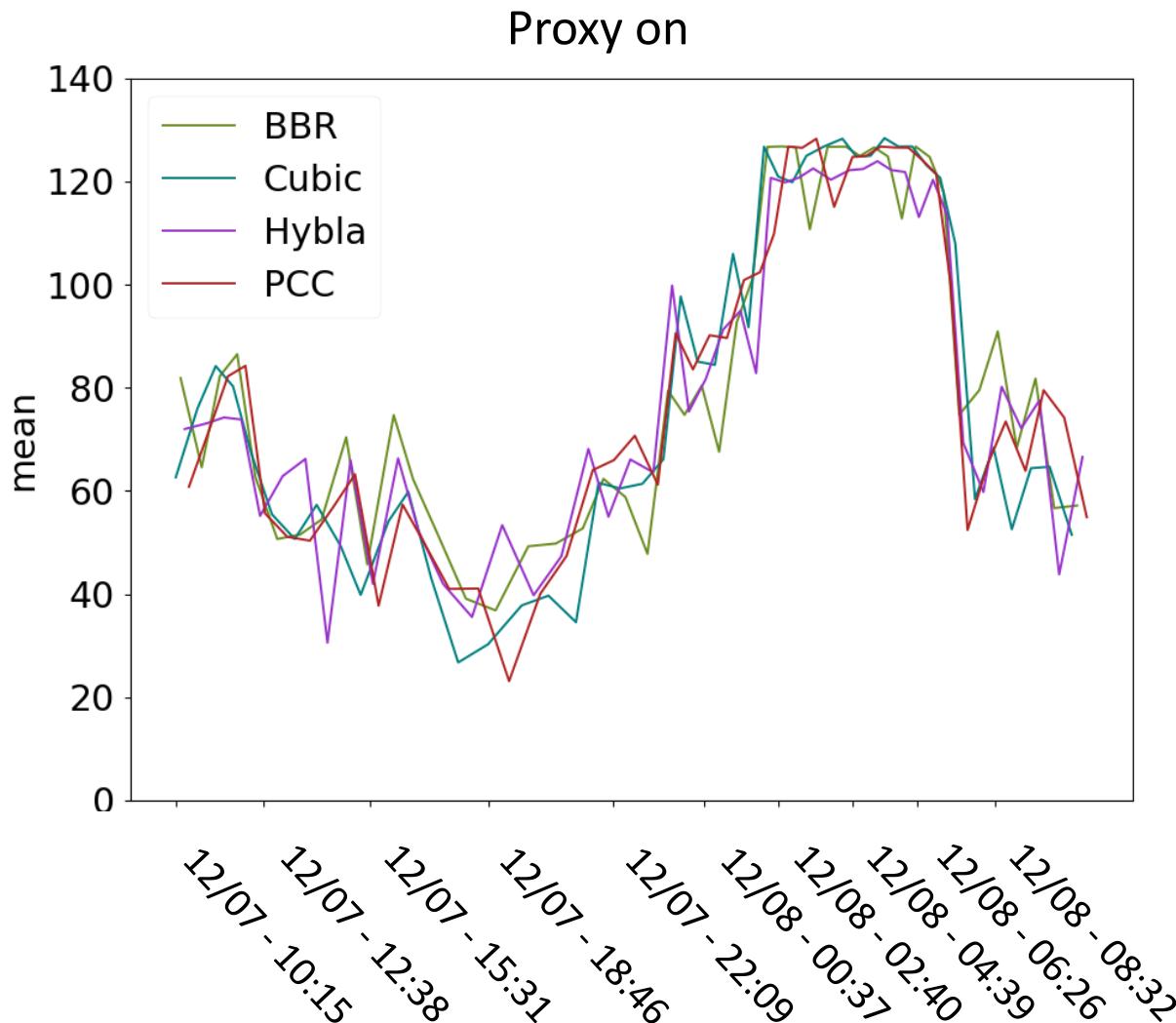
Proxy off



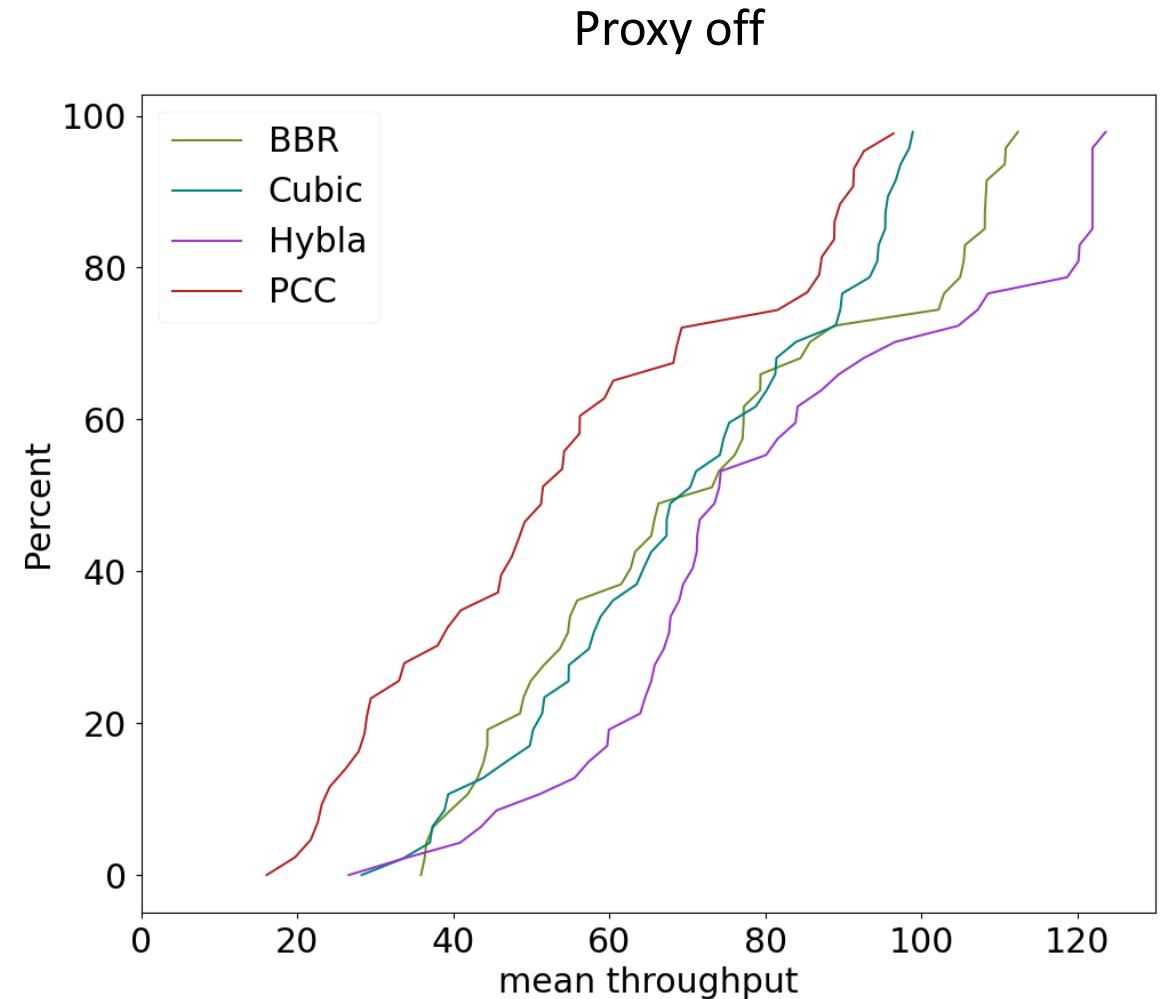
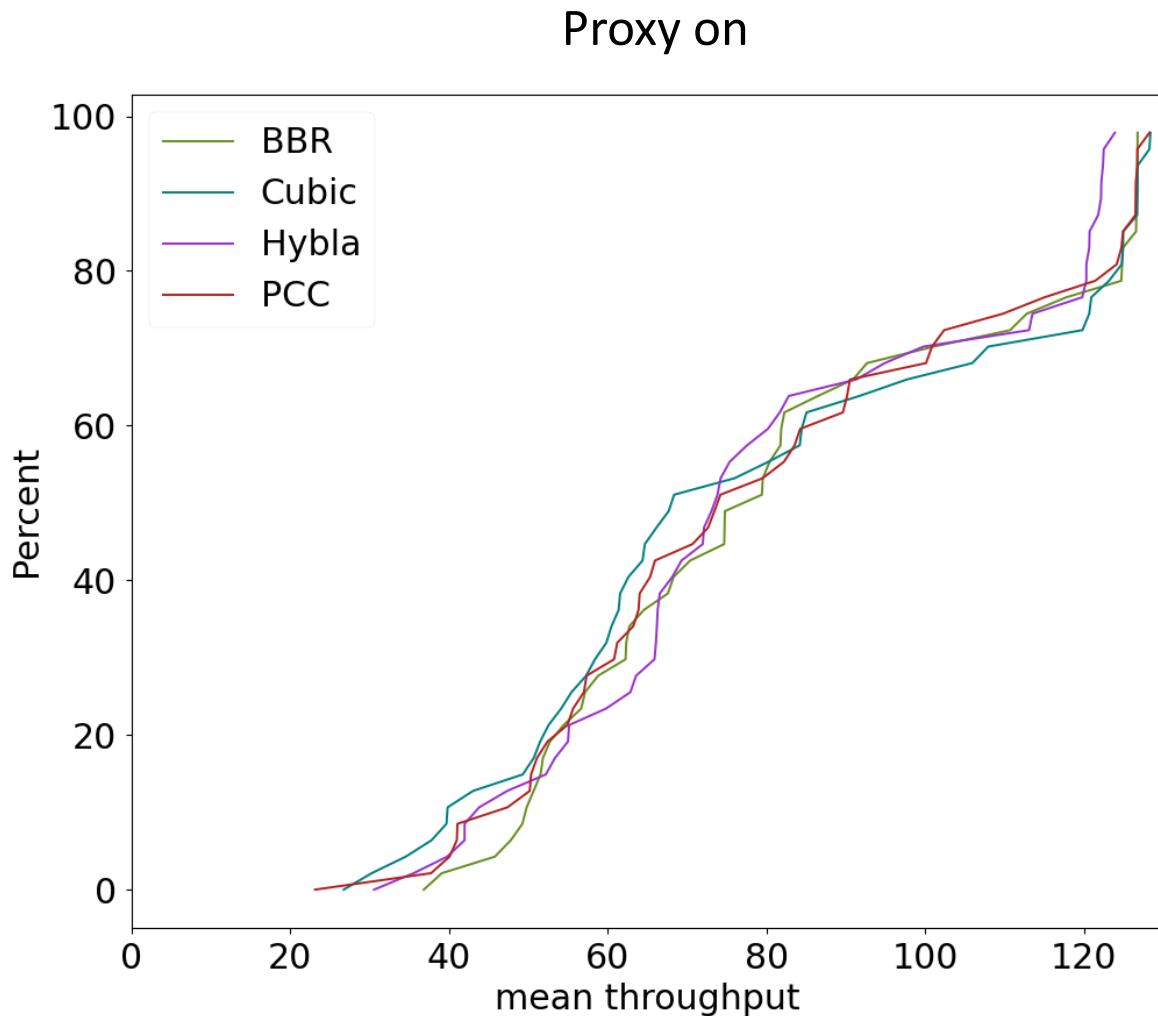
Worst



Weekday trial (24 hours)

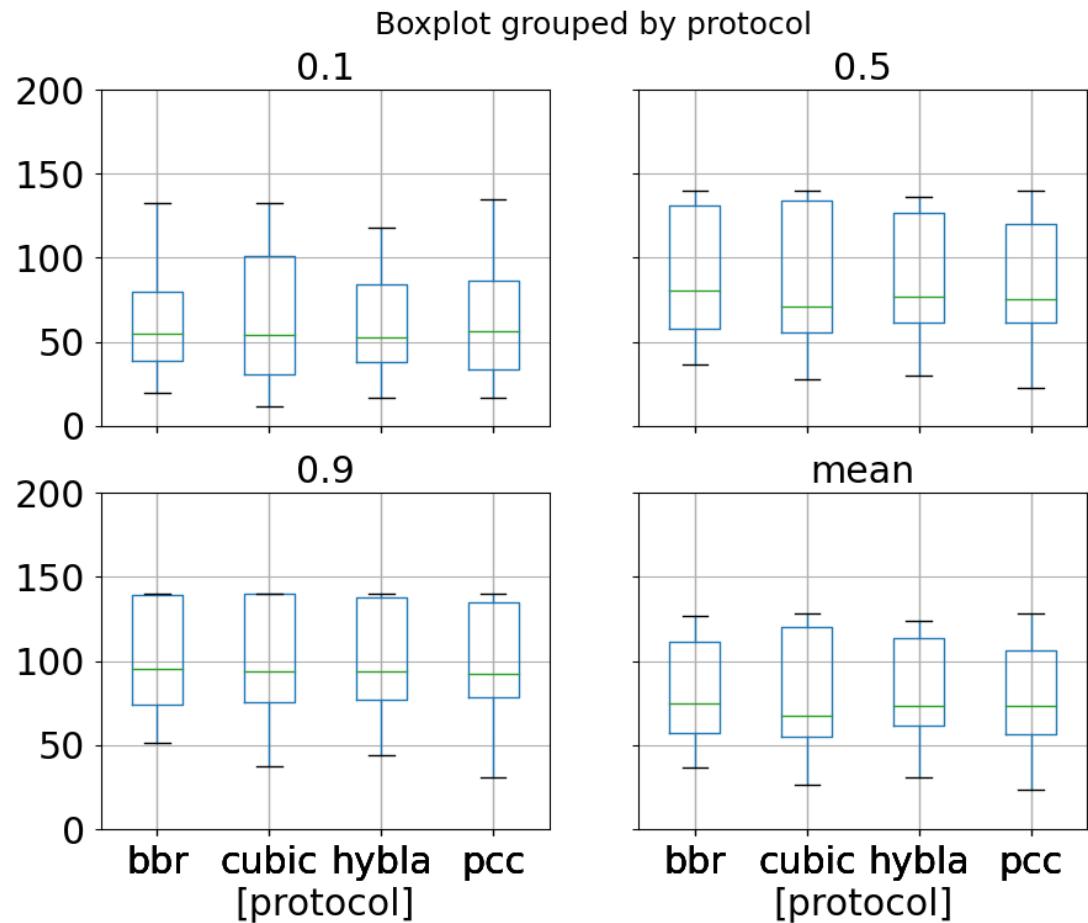


CDF (mean)

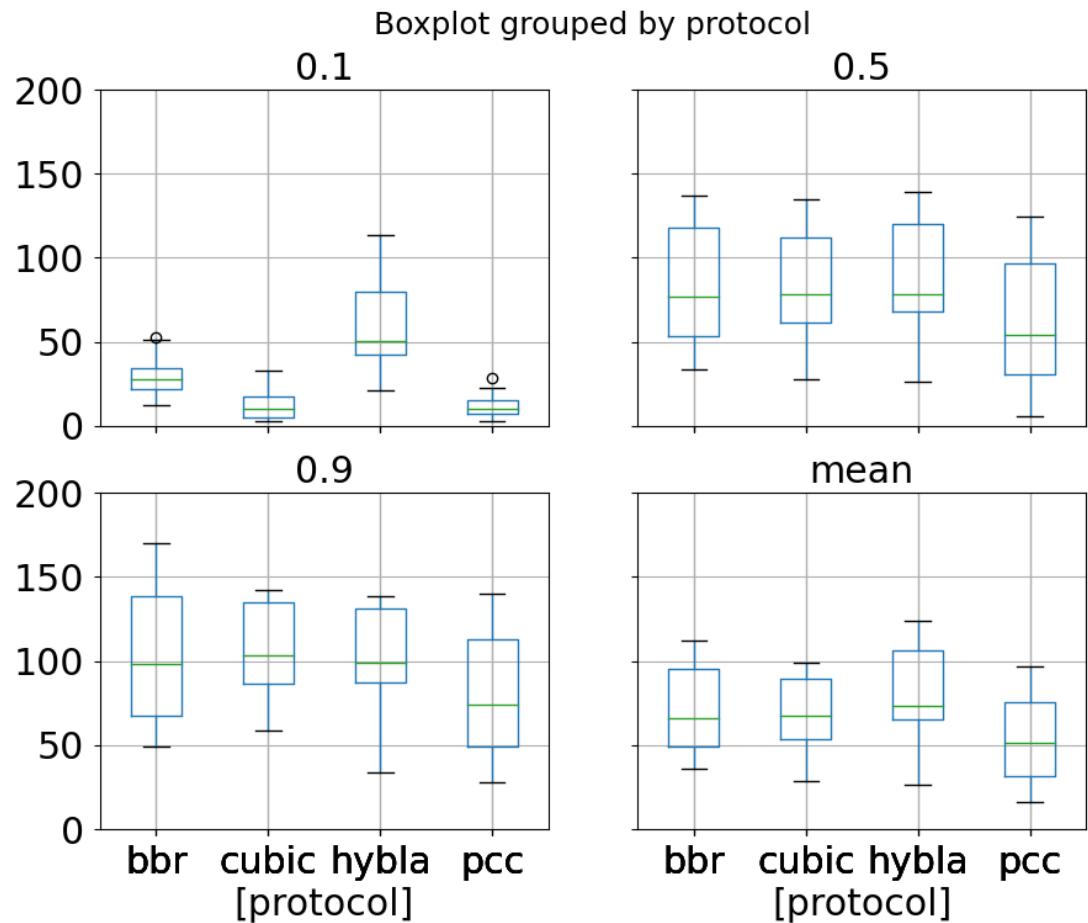


Boxplot

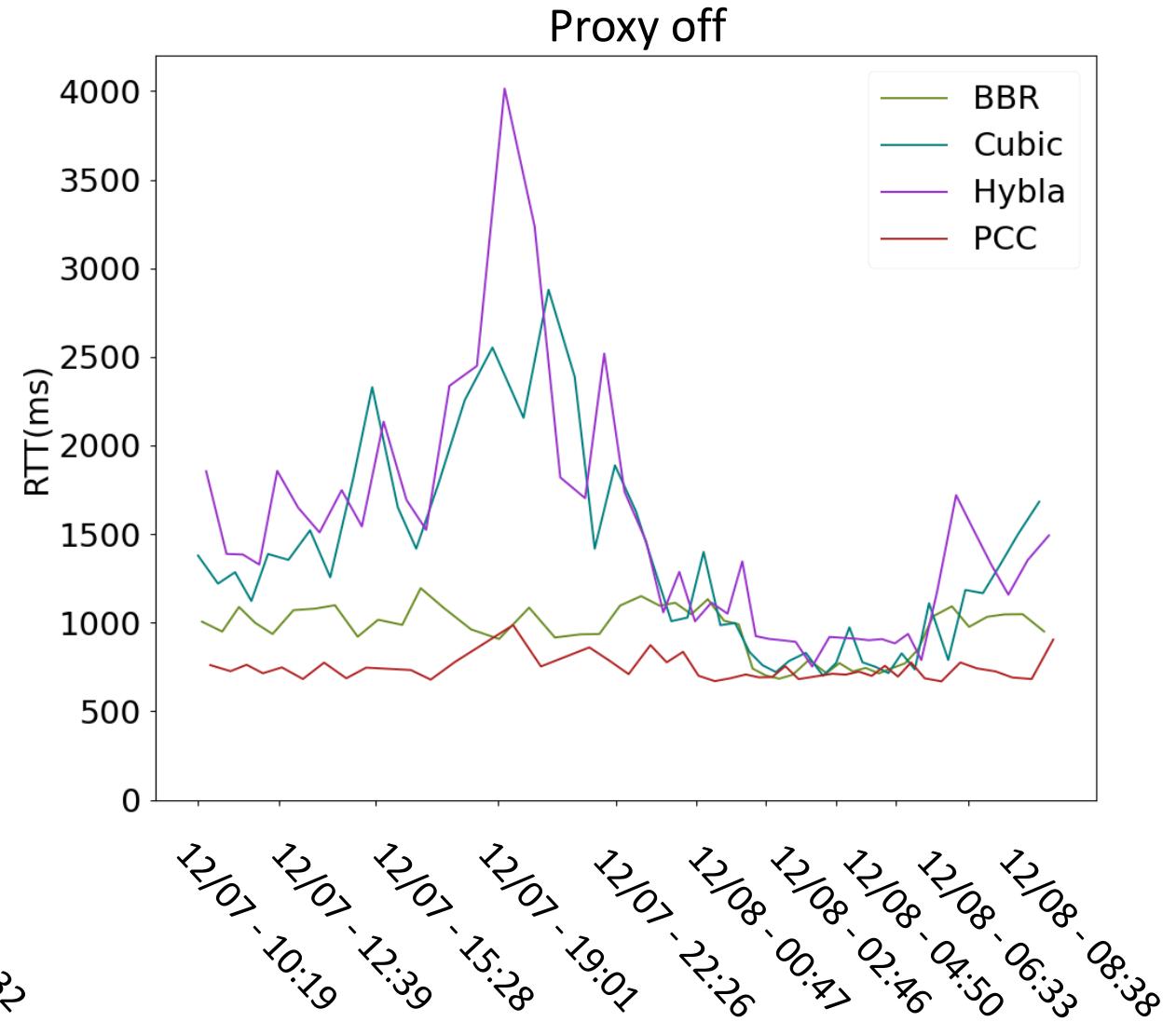
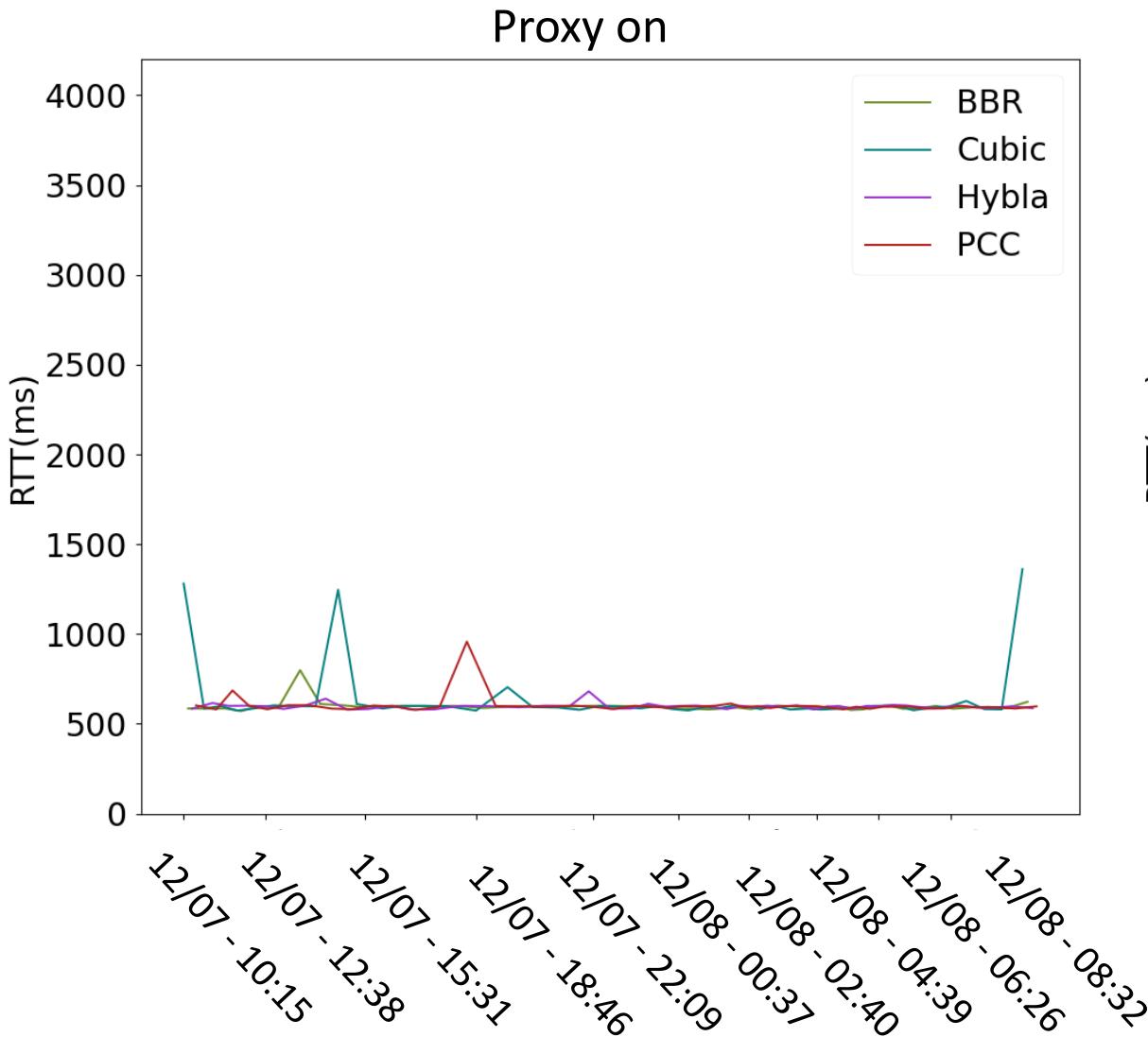
Proxy on



Proxy off

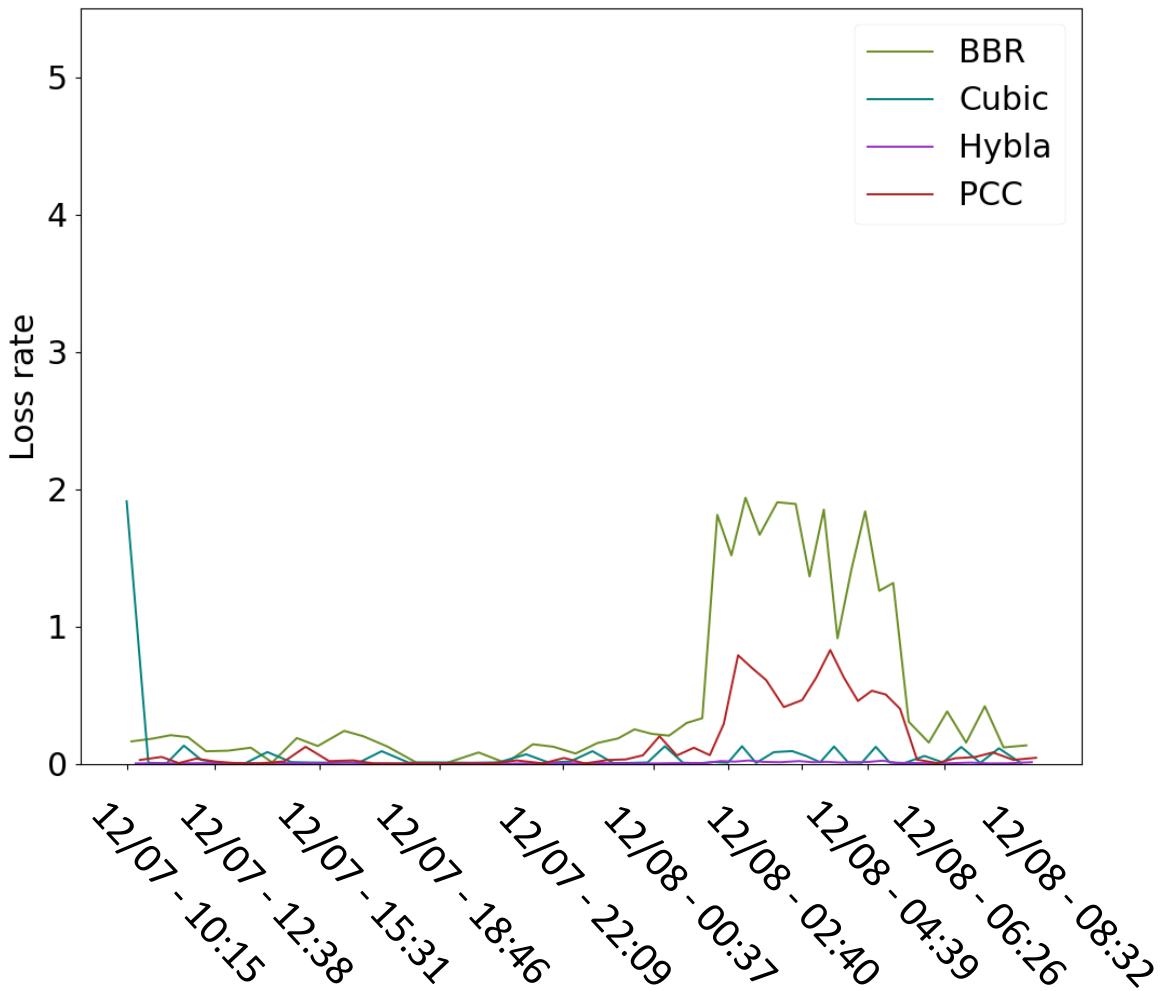


RTT over time

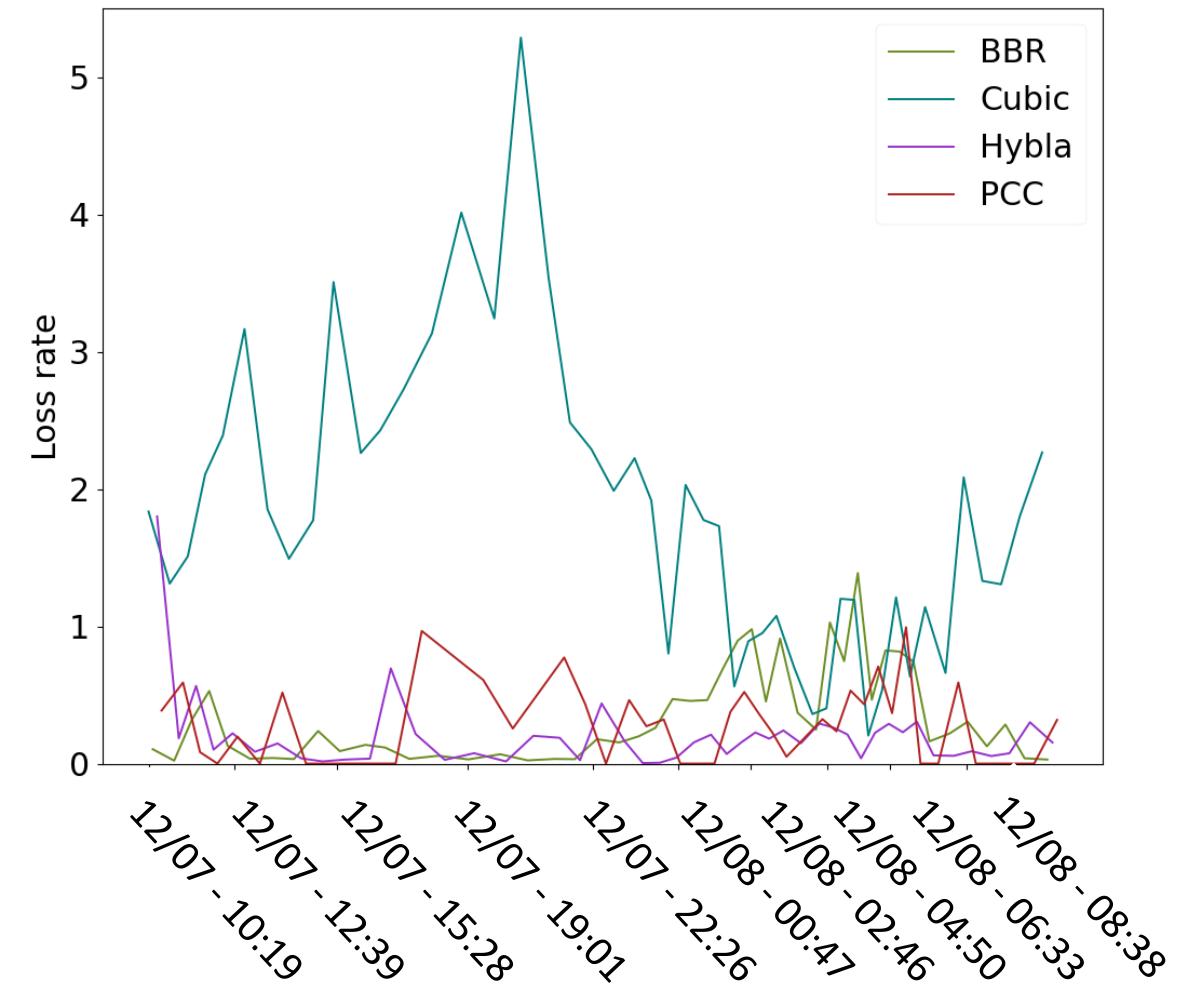


Loss over time

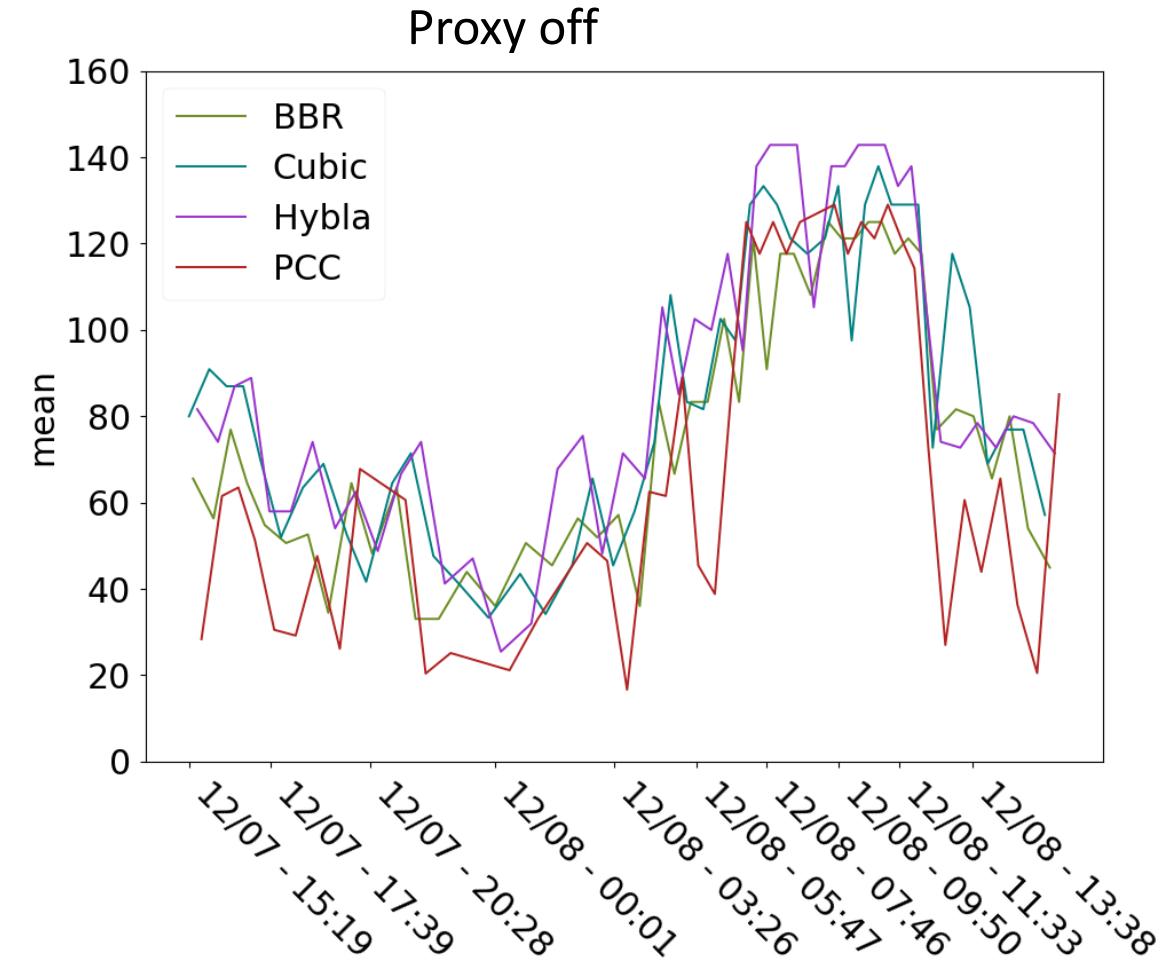
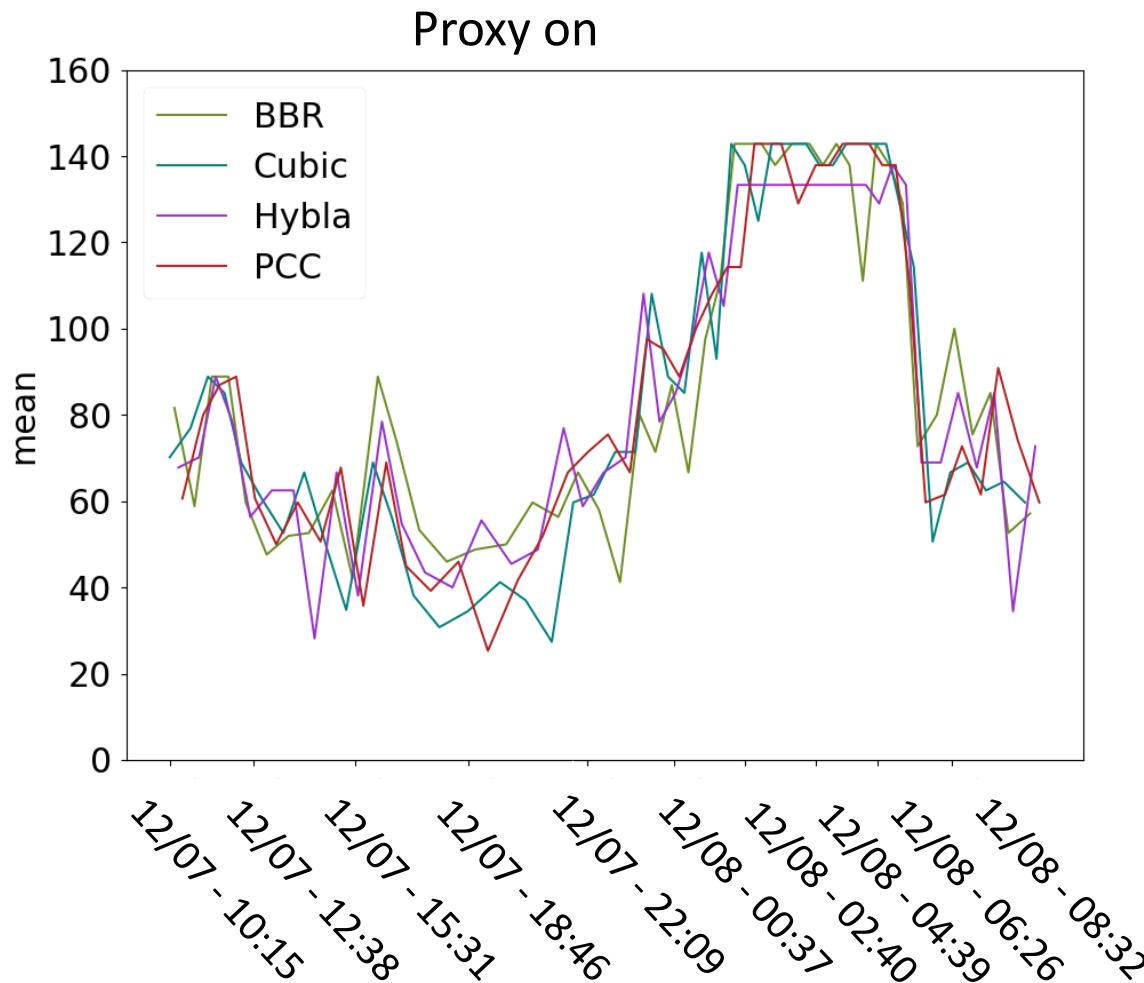
Proxy on



Proxy off

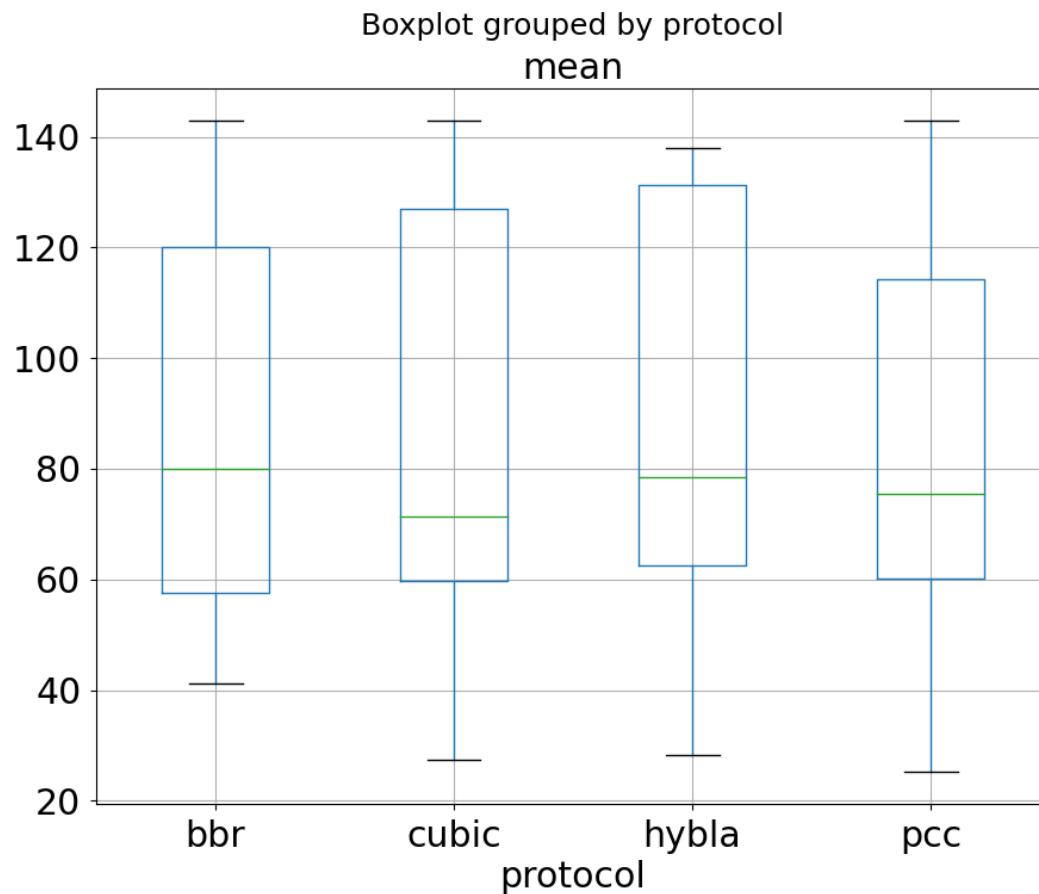


Weekday trial (steady state)

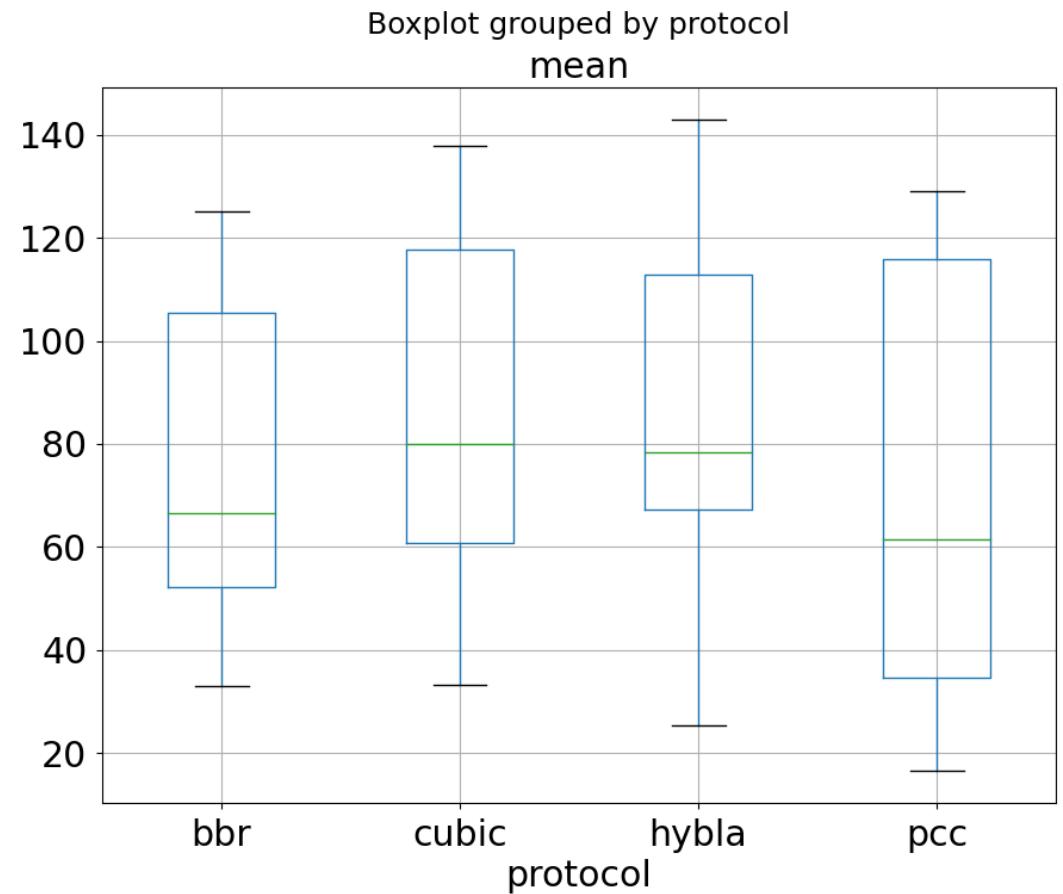


Boxplot (steady state)

Proxy on

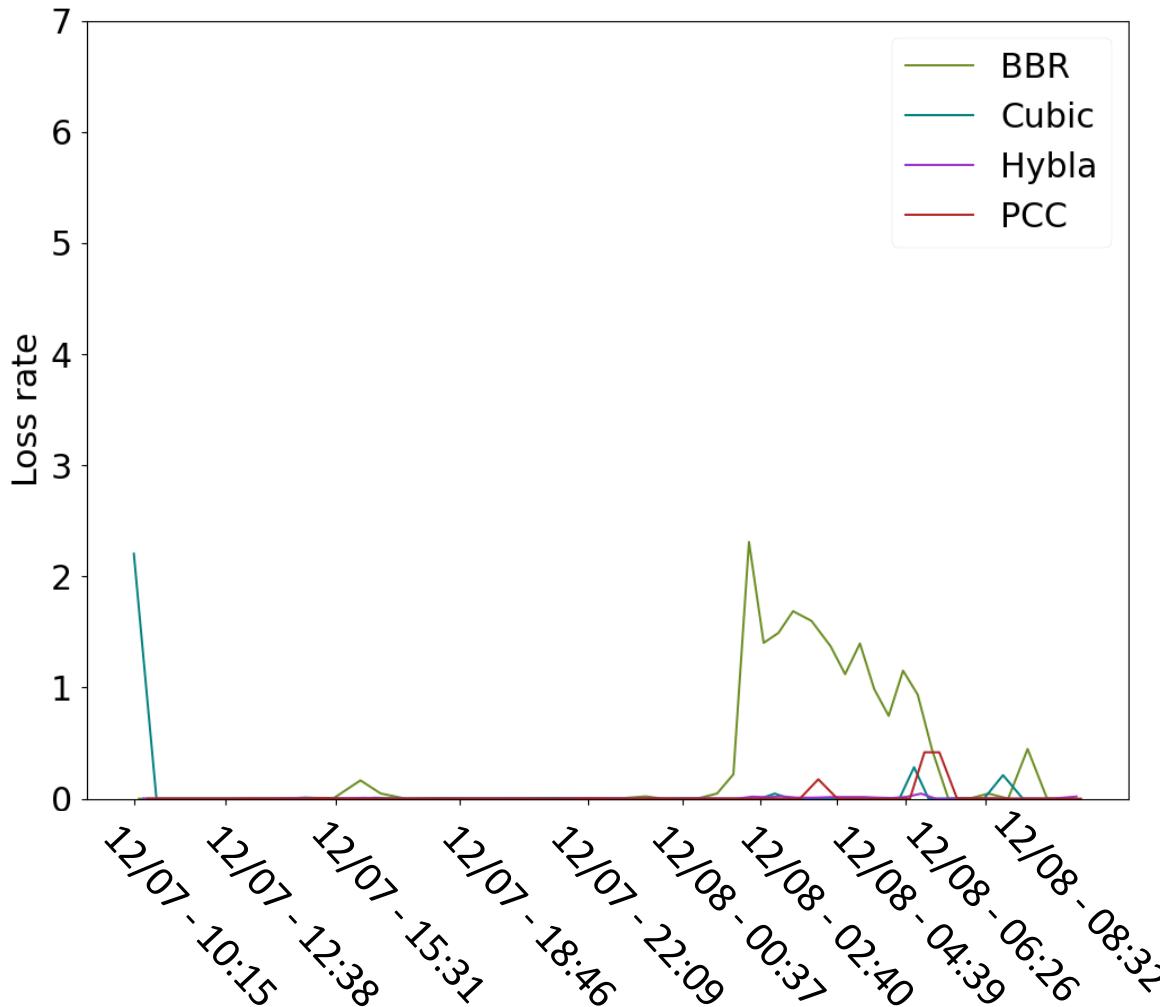


Proxy off

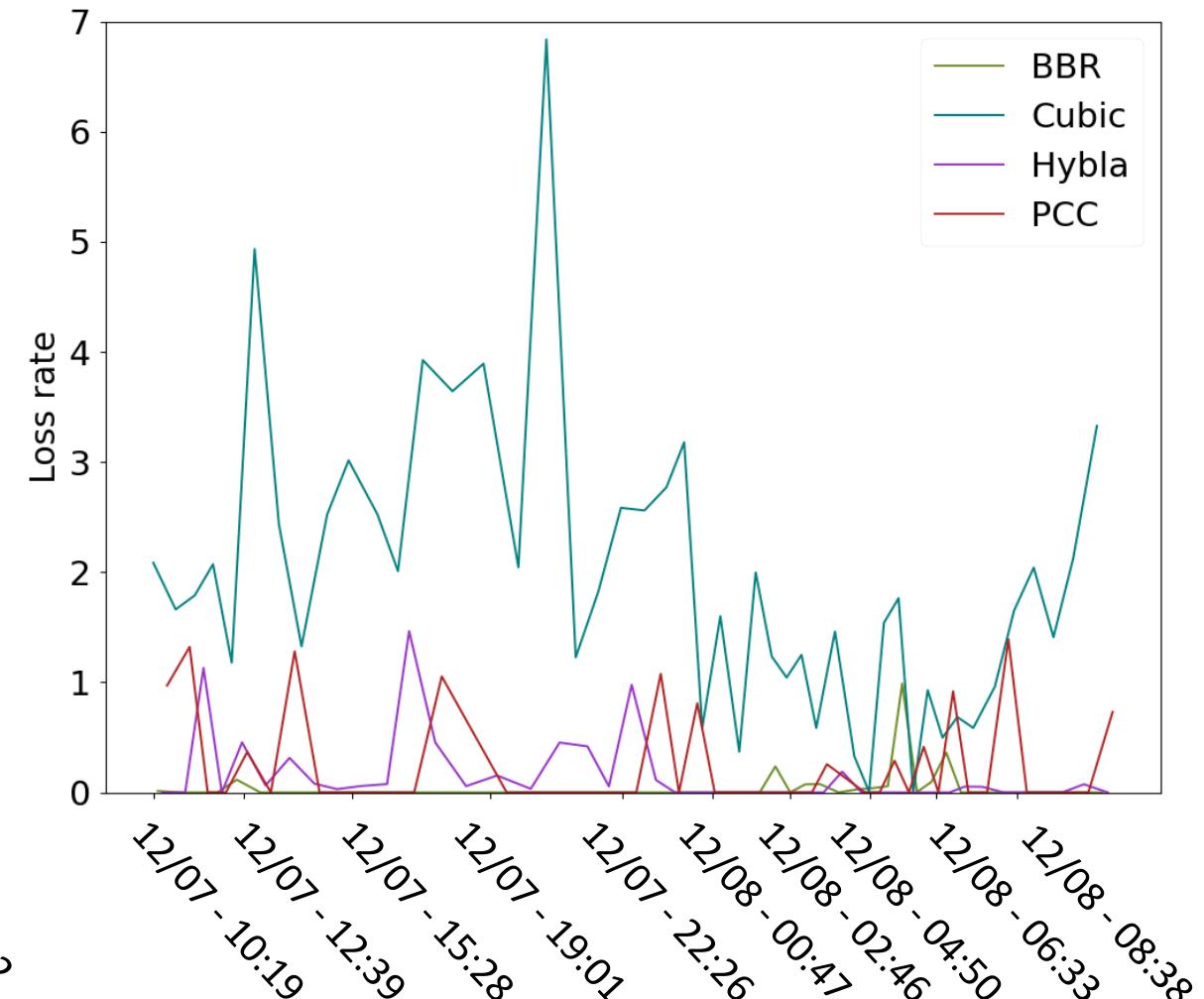


Loss over time (steady state)

Proxy on

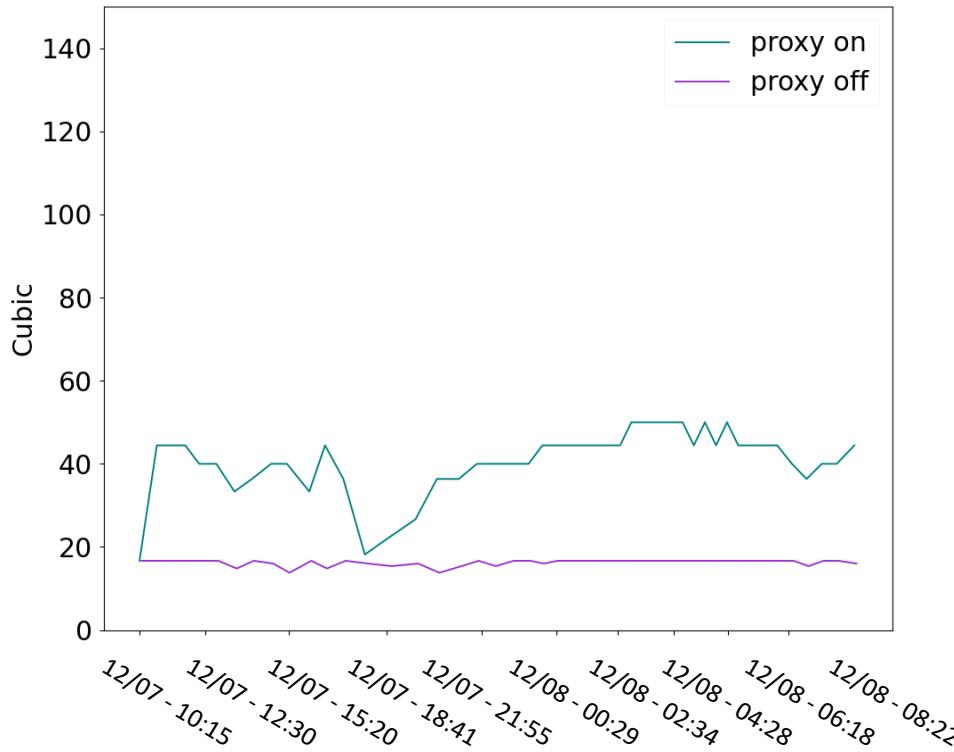


Proxy off



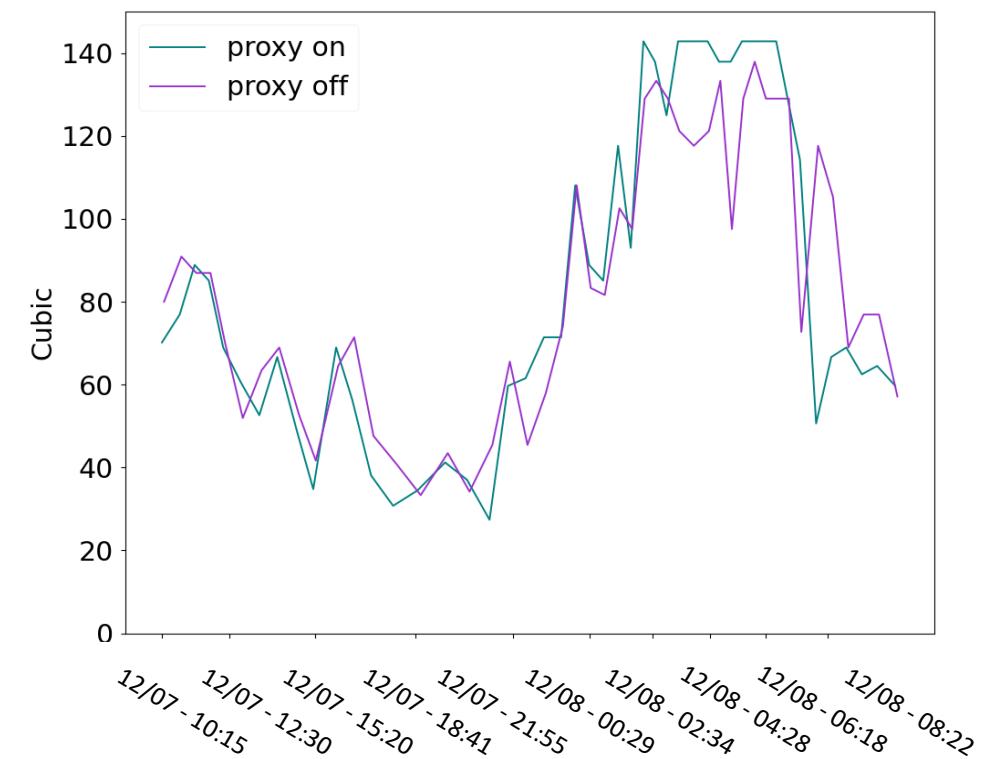
Average Diff & Standard Dev

Startup state



Cubic	Mbytes/sec
Average difference	24.6108975
Standard deviation	7.4474517

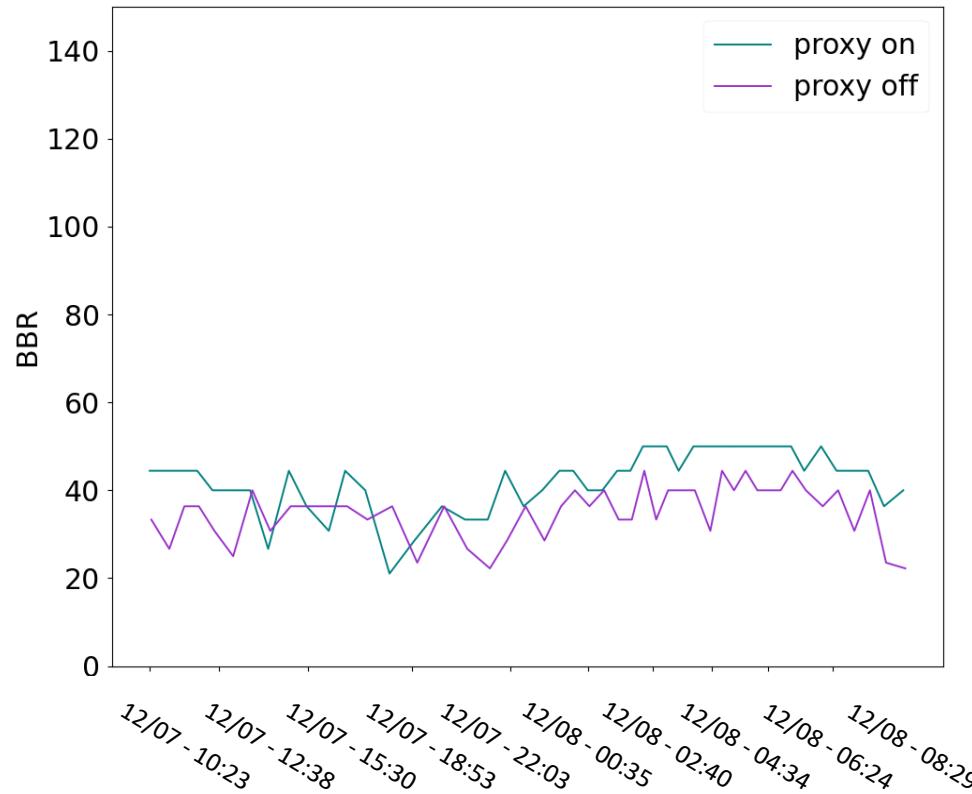
Steady state



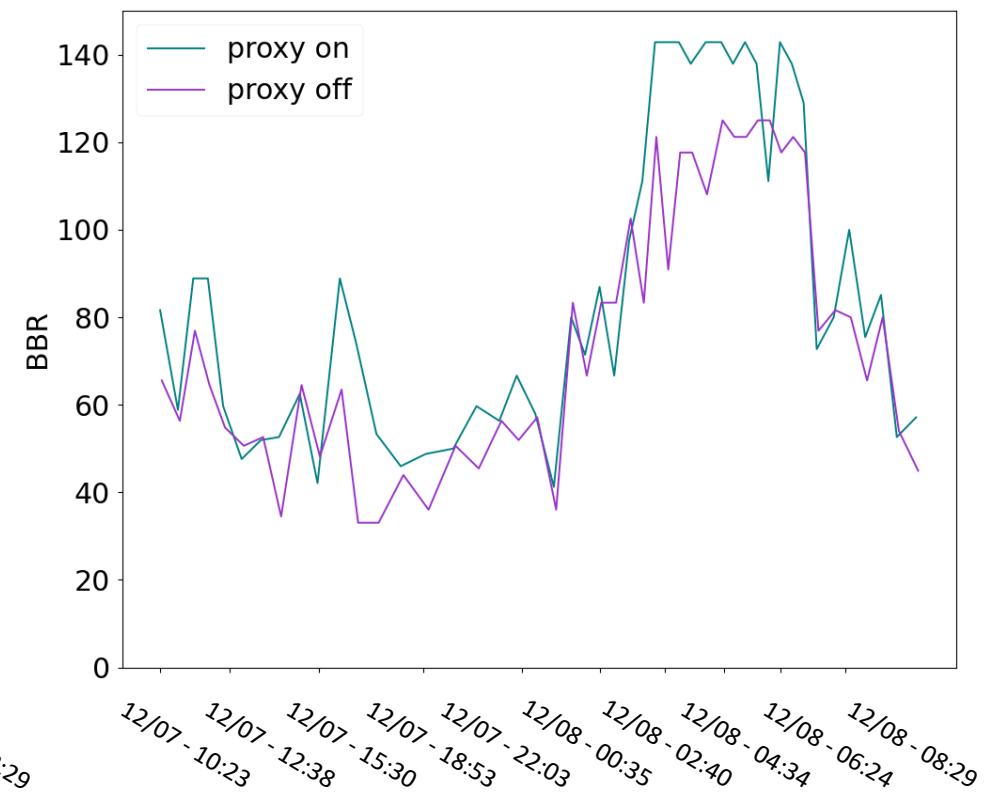
Cubic(steady state)	Mbytes/sec
Average difference	0.8478034
Standard deviation	17.5729021

Average Diff & Standard Dev

Startup state



Steady state

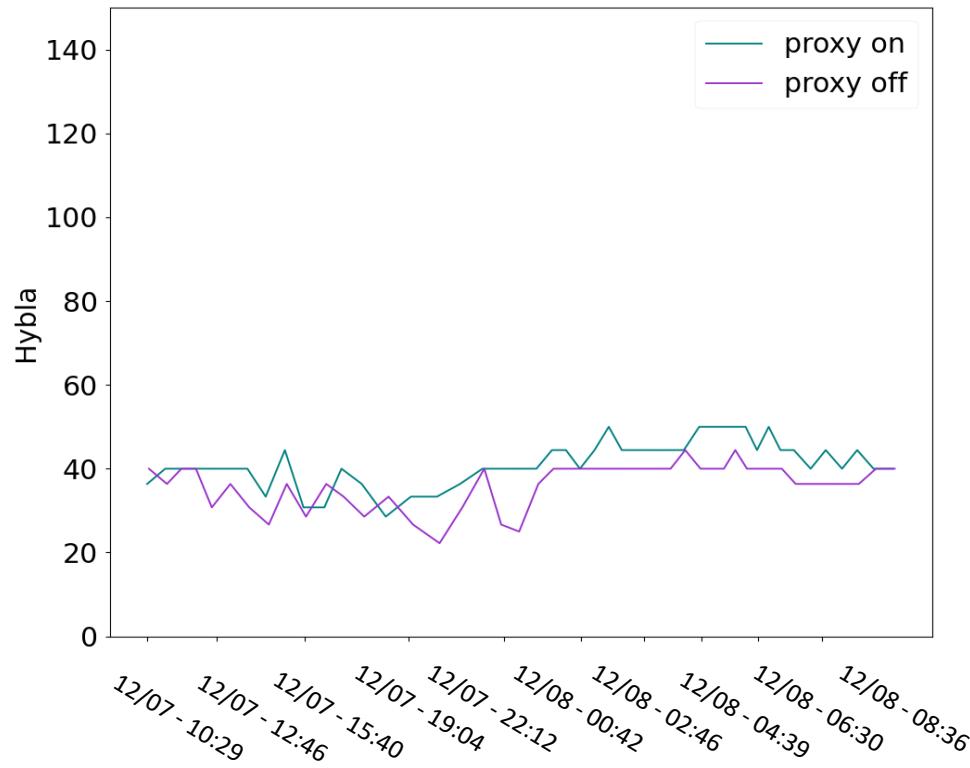


BBR	Mbytes/sec
Average difference	7.433451
Standard deviation	6.535414

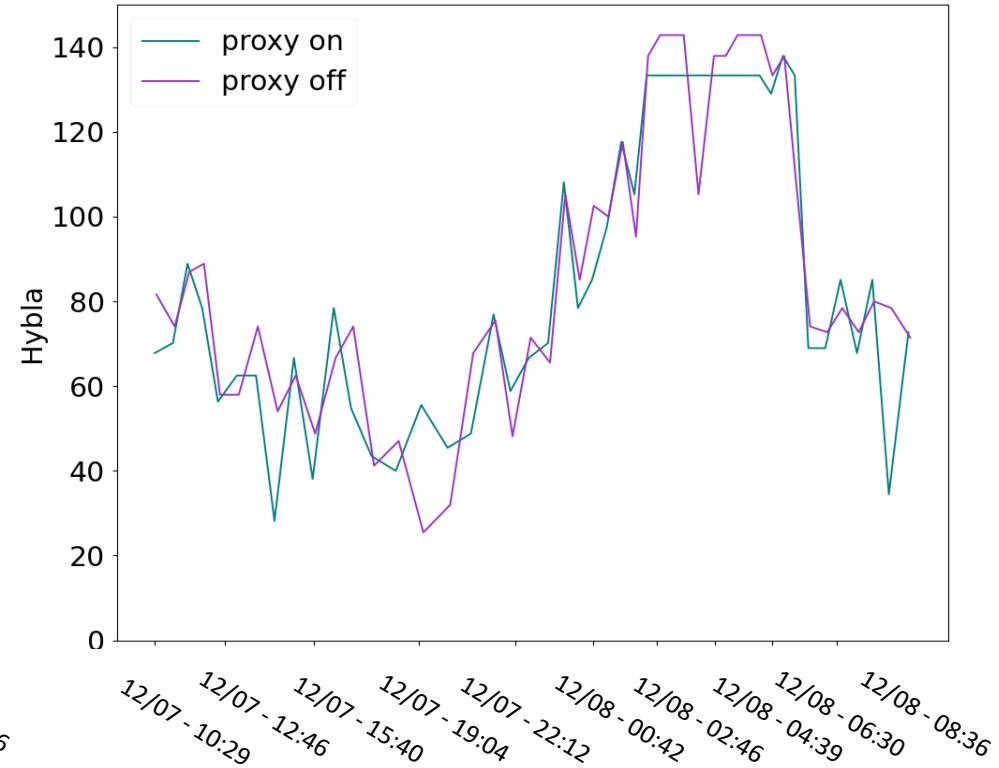
BBR(steady state)	Mbytes/sec
Average difference	10.9672840
Standard deviation	13.8420893

Average Diff & Standard Dev

Startup state



Steady state

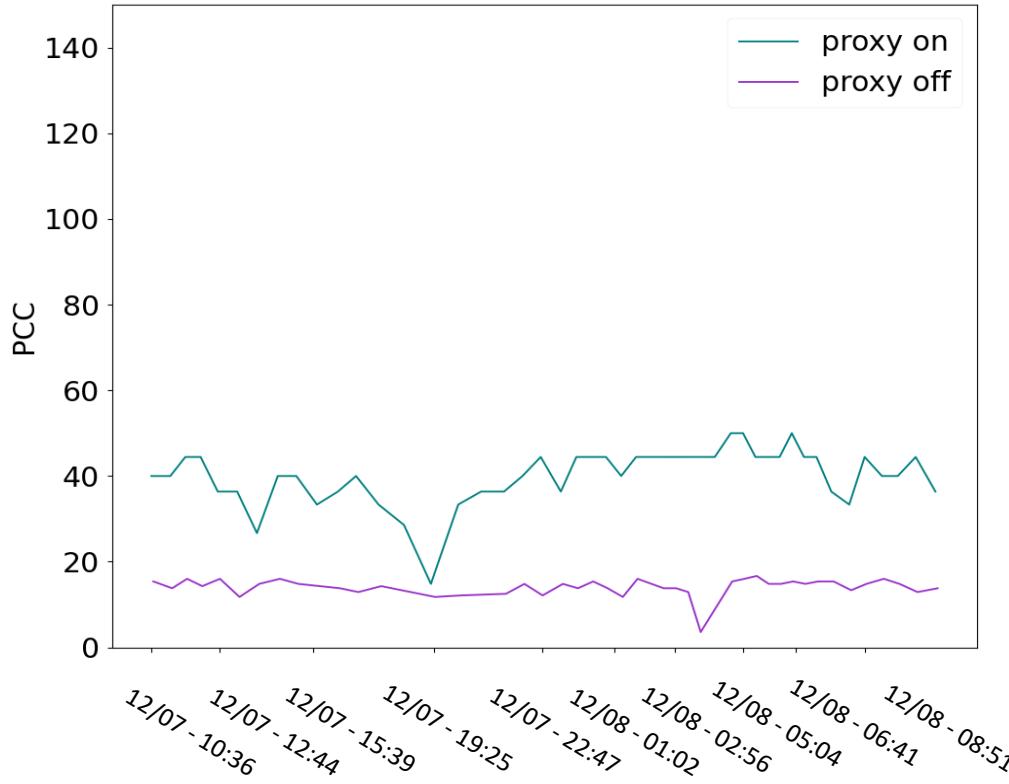


Hybla	Mbytes/sec
Average difference	5.106114
Standard deviation	4.482034

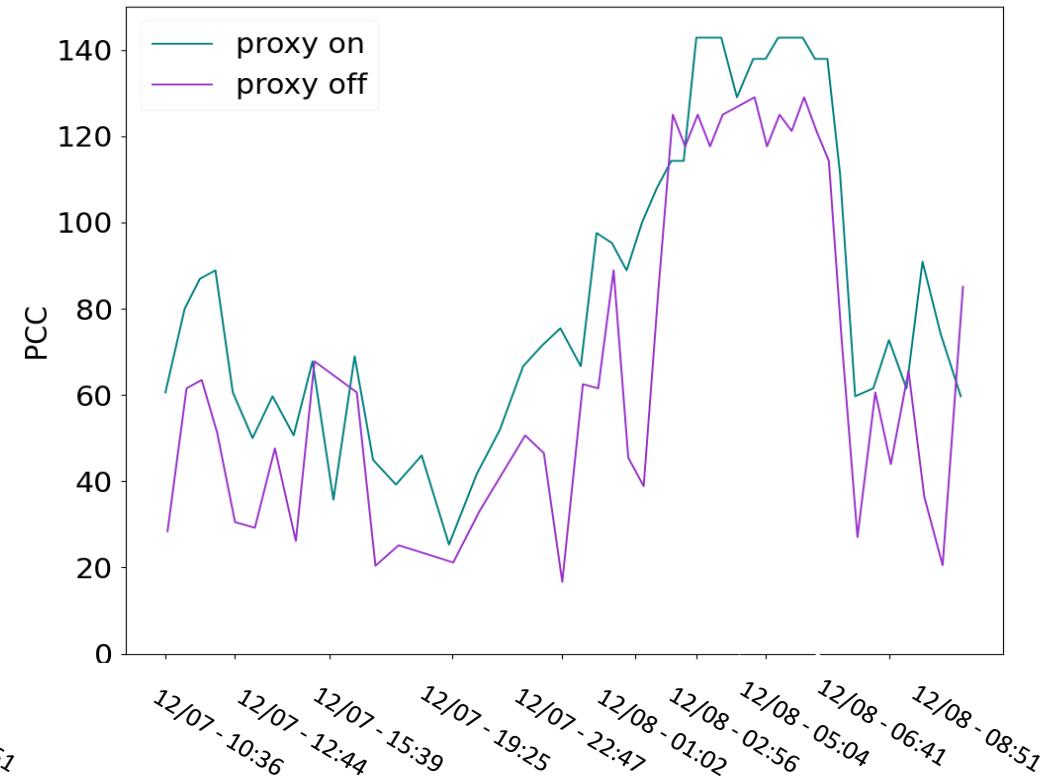
Hybla(steady state)	Mbytes/sec
Average difference	-2.6286000
Standard deviation	12.9442221

Average Diff & Standard Dev

Startup state



Steady state



PCC	Mbytes/sec
Average difference	26.195413
Standard deviation	6.631563

PCC(steady state)	Mbytes/sec
Average difference	19.9629806
Standard deviation	31.0013881

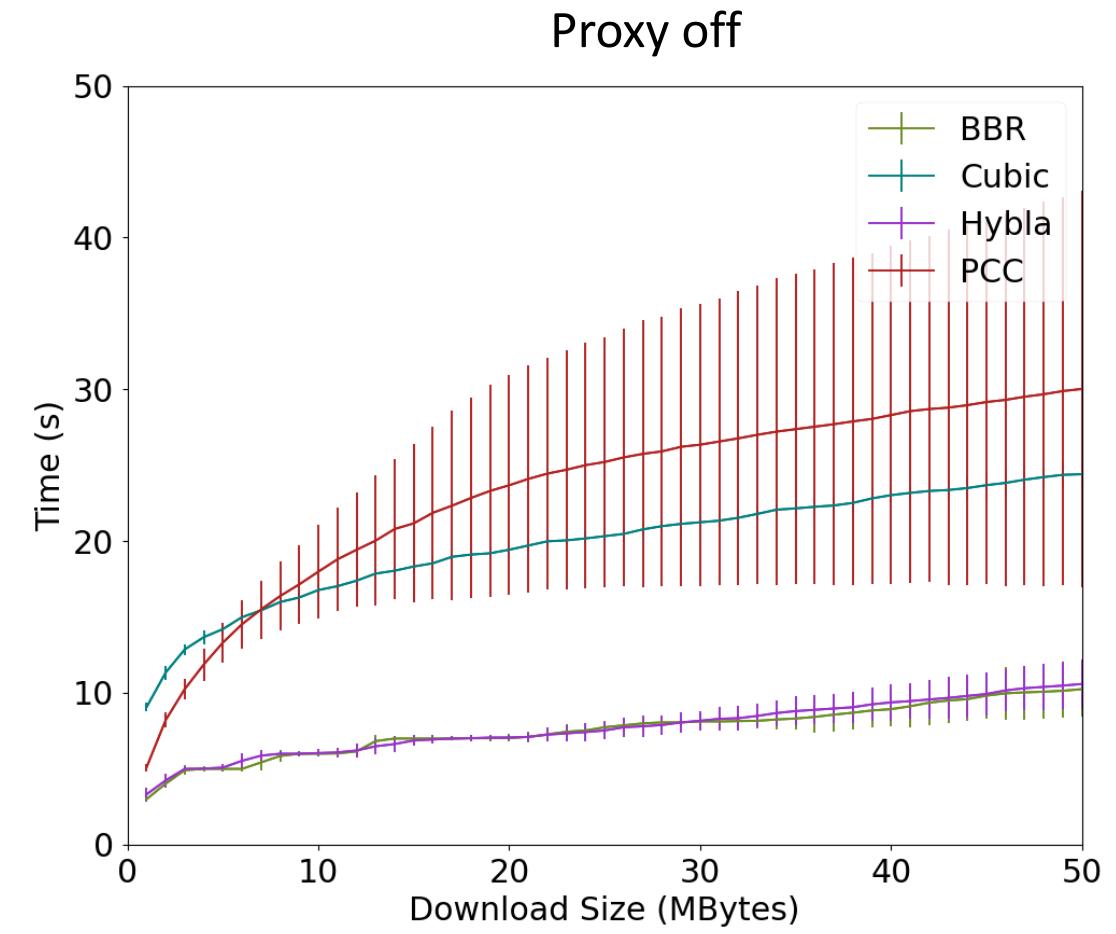
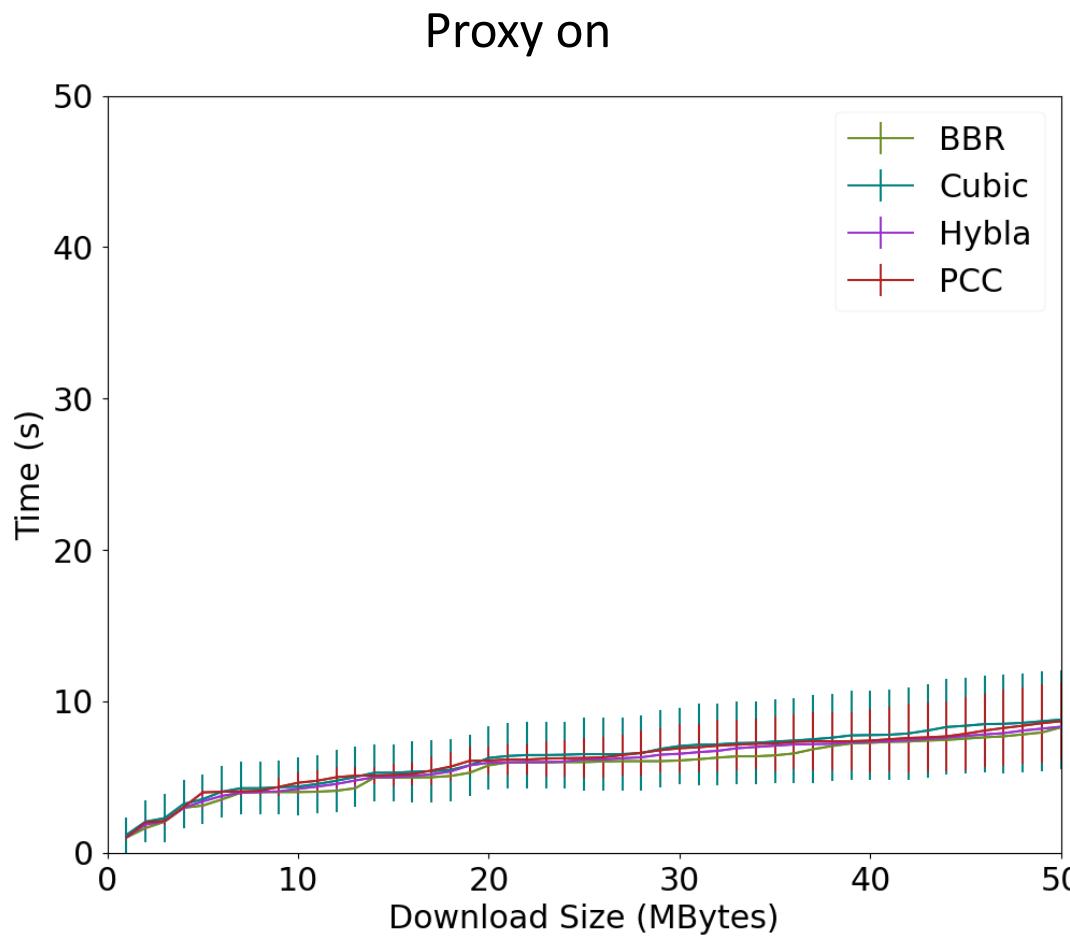
Summary

Complete(Mbytes/sec)	Cubic	BBR	Hybla	PCC
Average difference	10.7816896	10.1398785	-1.013994	27.6126552
Standard deviation	16.4409405	9.5258862	9.62518649	25.5394993

Startup state(Mbytes/sec)	Cubic	BBR	Hybla	PCC
Average difference	24.6108975	7.433451	5.106114	26.195413
Standard deviation	7.4474517	6.535414	4.482034	6.631563

Steady state(Mbytes/sec)	Cubic	BBR	Hybla	PCC
Average difference	0.8478034	10.9672840	-2.6286000	19.9629806
Standard deviation	17.5729021	13.8420893	12.9442221	31.0013881

Download time vs. Download object size



Linux Cubic Observations

- During initial growth, an additional 5% growth per RTT should be observed

```
/*
 * The initial growth of cubic function may be too conservative
 * when the available bandwidth is still unknown.
 */
if (ca->last_max_cwnd == 0 && ca->cnt > 20)
    ca->cnt = 20; /* increase cwnd 5% per RTT */
```

- Ideal center for baseline growth observation lies at 100ms RTT

```
/* 1/c * 2^2*bictcp_HZ * srtt */
cube_factor = 1ull << (10+3*BICTCP_HZ); /* 2^40 */

/* divide by bic_scale and by constant Srtt (100ms) */
do_div(cube_factor, bic_scale * 10);
```

- The default `tcp_cubic` refresh rate is 2^{10} updates/sec. I hypothesize a performance barrier exists at $\text{RTT} < 1/1024$ sec
- Hystart off `initial_ssthresh` is $\sim 10^9$ bytes

Linux Cubic Observations

- Possible that deviations from the expected 100ms lead to premature linear growth (As mentioned in "TCP startup performance in Large Bandwidth Delay Networks") (Observed on our connection RTT~0.5ms).
- I'm unsure if this is related to our lower-than-expected growth speed over the satellite connection with respect to `init_cwnd`. A miscalculation of the `cubic_power` variable with respect to the RTT.
- Testing setup:
 - Artificially imposed delay ranging from 0ms to 1024ms including tests at 100ms and 600 ms (near Satellite RTT).
 - Comparisons between `cwnd` impact in LAN vs link.
 - Might be possible to modify kernel to include new hardcoded value of `cubic_power` with standard RTT=600ms.