Pantheon Programming Assignment 3 – Report

Yeshwanth Akula

Banner Id: 001378540

Github: https://github.com/yesh-is-here/pantheon-pa3

1. Overview

This report details the experimentation and evaluation of three congestion control algorithms using the Pantheon framework: CUBIC, BBR, and COPA. We performed controlled local tests using Mahimahi-emulated network conditions. The evaluation was conducted to observe and compare uplink throughput, average RTT, and packet loss rate across all schemes.

2. Selected Algorithms

CUBIC: A widely adopted loss-based congestion control algorithm used as the Linux default.

BBR: A model-based algorithm by Google that targets high throughput with low latency.

COPA: A delay-sensitive congestion control algorithm that reacts swiftly to network congestion without bufferbloat.

3. Network Environment and Test Methodology

We used the following standardized setup:

Network Trace: 12mbps_data.trace and 12mbps_ack.trace from Mahimahi.

Duration: Each test was run for 60 seconds.

Testbed: Pantheon on Ubuntu 22.04 LTS with Python 3.12 and Mahimahi installed.

The test pipeline:

1. Pantheon scripts were invoked using test.py.
2. Mahimahi simulated the network using mm-link.

3. Resulting logs were generated in the results/ folder.
4. A custom Python script analyze_logs.py was developed to parse and extract metrics.
5. Results

4.1 Uplink Throughput (Mbps)

All schemes performed comparably:

CUBIC: 12.478 Mbps

BBR: 12.318 Mbps

COPA: 12.495 Mbps

4.2 Average RTT (ms)

All schemes achieved ~1 ms RTT.

4.3 Packet Loss Rate (%)

Loss was negligible across CUBIC and BBR.

COPA did not log loss explicitly, which we documented accordingly.

6. Problems Encountered and Fixes
    a. Python 3 Compatibility

Problem: Pantheon used Python 2-style syntax (e.g., print without parentheses, xrange, unicode).

Fix: Modified all relevant scripts to Python 3.12-compatible syntax.

    b. YAML Loader Error

Problem: yaml.load() threw TypeError due to missing loader.

Fix: Updated to yaml.load(config, Loader=yaml.FullLoader).

    c. Type Errors in JSON Metadata

Problem: Pantheon wrote bytes into json.dump().

Fix: Decoded byte strings using .decode() before dumping.

d. System-Wide Package Restrictions

Problem: Ubuntu disallowed pip install due to external environment protections.

Fix: Created and activated a venv virtual environment, then installed dependencies locally.

e. COPA Wrapper Parsing

Problem: COPA's wrapper script failed due to indentation and syntax errors.

Fix: Rewrote copa.py with corrected argument parsing and added fallback default.

f. Missing Dependencies

Problem: COPA required libprotobuf-dev, libboost-dev, and other libraries.

Fix: Captured these dependencies in the script's deps handler and installed manually.

7. Custom Script – analyze_logs.py

A custom Python script was developed to:

Parse mm_datalink_run1.log files.

Compute average throughput.

(Optionally) estimate RTT and loss using heuristics.

Output a formatted Pandas dataframe.

Generate bar graphs using matplotlib. Graphs are in This script is included in the submission and can be run via python3 analyze_logs.py in the virtual environment.

8. How to Reproduce

1. Set up virtual environment

Sudo apt install python3.12-venv

Python3 -m venv venv

Source venv/bin/activate

Pip install pandas matplotlib

2. Run tests

```
PYTHONPATH=src:src/helpers python3 src/experiments/test.py local \
 --schemes “cubic bbr copa” \
 --uplink-trace tests/12mbps_data.trace \
 --downlink-trace tests/12mbps_ack.trace \
 --data-dir results \
 --runtime 60
```

3. Run analysis

```
Python3 analyze_logs.py
```

9. Submission Contents

All following Contents are pushed to GitHub RepoResults/ folder with all experiment logs

Analyze_logs.py (Python 3 script for analysis)

README.md with setup instructions

Report.pdf (this document)

Throughput.png, rtt.png, loss.png

10. Final Notes

Despite numerous hurdles — from deprecated Python syntax to undocumented library errors — we successfully modernized the Pantheon framework, conducted fair experiments, and delivered replicable, visual results. This submission reflects a deep engagement with low-level network emulation and scripting.

Thank you for this comprehensive and rewarding assignment.