Project 1

▼ Hide Assignment Information

Instructions

Project 1 will be about using *traceroute*, parsing its output, and performing a statistical analysis of the traceroute results.

This project will require using Python to create a command line tool that automatically executes *traceroute* multiple times towards a target domain name or IP address specified as command line parameter. Based on multiple *traceroute* executions, the program will need to derive latency statistics for each hop between the *traceroute* client and the target machine.

To allow for repeatable tests, the program should also allow reading pre-generated *traceroute* output traces stored on multiple text files (one text output trace per file). Based on this pre-generated output, the program will need to compute the latency statistics as for the case of live *traceroute* execution.

Additional details about Project 1 will be provided in class.

Submission instructions
Make a directory called *trstats* containing a single *Python3* file called *trstats.py*. Create a *.tar.gz* archive containing the *trstats* directory, name the resulting file as *Project1-LASTNAME_FIRSTNAME.tar.gz*, and submit it via eLC under this assignment.

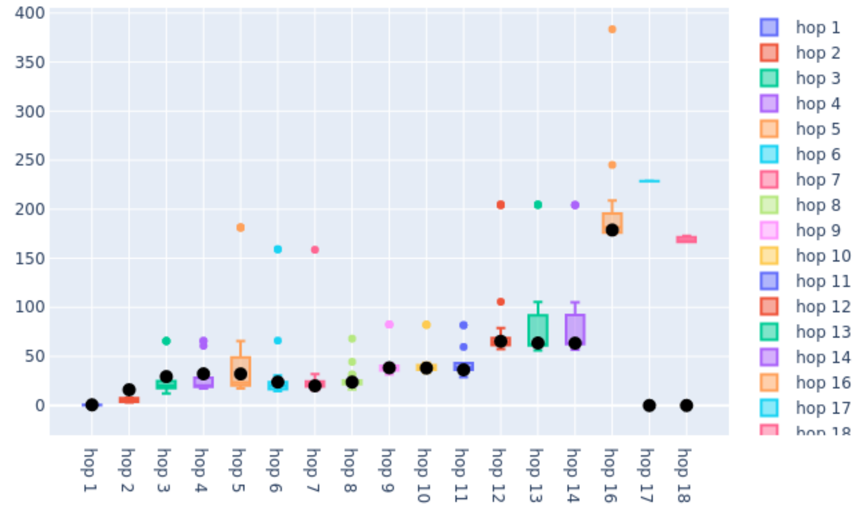
Detail instructions
Your command-line tool will need to support the following CLI arguments

```
usage: tr_stats.py [-h] [-n NUM_RUNS] [-d RUN_DELAY] [-m MAX_HOPS]
                   -o OUTPUT -g GRAPH [-t TARGET] [--test TEST_DIR]

Run traceroute multiple times towards a given target host

optional arguments:
  -h, --help       show this help message and exit
  -n NUM_RUNS      Number of times traceroute will run
  -d RUN_DELAY     Number of seconds to wait between two consecutive runs
  -m MAX_HOPS      Number of times traceroute will run
  -o OUTPUT        Path and name of output JSON file containing the stats
  -g GRAPH         Path and name of output PDF file containing stats graph
  -t TARGET        A target domain name or IP address
  --test TEST_DIR  Directory containing num_runs text files, each of which
                   contains the output of a traceroute run.If present, this
                   will override all other options and tcpdump will not be
                   invoked. Stats will be computed over the traceroute output
                   stored in the text files
```

Essentially, the main task in this project is to write a *Python3* wrapper arround *traceroute*, so that you can programmatically run *traceroute* multiple times, read the latency statistics output by every run, and build a distribution of latency values over which to compute the required statistics. For instance, the main output of your program should be a file in JSON format that looks like this example:

```
[{'avg': 0.645,
  'hop': 1,
  'hosts': [('172.17.0.1', '(172.17.0.1)')],
  'max': 2.441,
  'med': 0.556,
  'min': 0.013},
 {'avg': 6.386,
  'hop': 2,
  'hosts': [('testwifi.here', '(192.168.86.1)')],
  'max': 16.085,
  'med': 5.385,
  'min': 3.108},
 {'avg': 26.045,
  'hop': 3,
  'hosts': [('96.120.4.5', '(96.120.4.5)')],
  'max': 65.753,
  'med': 20.298,
  'min': 12.287},
 {'avg': 26.819,
  'hop': 4,
  'hosts': [('96.110.205.9', '(96.110.205.9)')],
  'max': 65.847,
  'med': 20.51,
  'min': 17.444},
...
 {'avg': 168.84,
  'hop': 18,
  'hosts': [('124.83.228.222', '(124.83.228.222)')],
  'max': 172.869,
  'med': 166.869,
  'min': 166.781}]
```

Also, the program should output a boxplot graph showing the latency distribution per each hop, similar to this one:



Due Date

Sep 10, 2021 11:59 PM

Submit Assignment

Files to submit *

(0) file(s) to submit

After uploading, you must click Submit to complete the submission.

Add a File Record Audio

Comments

Paragraph B *I* U Lato (Recomm...

Submit Cancel