CSE 5299 – Fall 2024 Programming Assignment #2

Ariannah Black. (November 12, 2024)

Ping Program Design Document

Note: The code and (in-line) documentation for this project can also be found at the following GitHub link: https://github.com/ariblack17/ping-program.

Description

The Program: The Ping program for this project is intended to be a manual implementation of the ping function and a few of its features. The user, from the command line, calls the *ping* method for some host server. As a result, their system sends an ICMP packet (echo request) to the server, containing the current time. The server sends a response ICMP (echo response) back to the client, and the client is then able to calculate the RTT from the time difference between requests and responses, if the packet received was an expected ICMP. Packets will be "timed out" if they have not been received before their TTL expires (specified by the *-t* command line option). This function repeats until the number of sent packets (specified by the *-c* command line option) is reached, and some relevant statistics (average RTT, min RTT, max RTT, …) about the session is output to the client's console.

The command to run the *ping* method is as follows:

sudo python3 ICMP.py ping <host_ip> <options>

Trade Offs

Parsing ICMP Error Codes: In order to parse the error codes received by ICMP messages from the server, I had to manually map each possible error type and code to its respective description, for both IPv4 and IPv6 addresses. While this solution is functional, it is also rather inefficient.

Catching Errors: While the code design meets the specifications required by the assignment, there could be more done to improve the project's robustness and make it more similar to the actual built-in Ping function.

User Interface: Given the starter code, I felt that it was most natural to have the user interact with the client-side program directly via the command line. In previous versions of the program, I had developed a client-side driver program that the user would run instead of using the command line interface; this, however, seemed to add some unnecessary complexity to the code.

Extensions

Increasing Similarity to Ping: The most obvious extension to this project would be to implement more of the functionalities present in the built-in ping function into the manually-implemented version. Some of these may include the *-i* (wait some number of seconds before sending each packet), *-o* (stop just after sending one packet), or the *-q* (quiet, so display nothing except the post-ping summary) options.

Streamlining Code Structure: Currently, there is a large amount of logic located under an *if __name__ == "__main___":* block in the main code file. Again, this part of the assignment is functional but perhaps not optimal. Rewriting and refactoring the code here to better fit the structure of the rest of the code (especially the starter code and helper functions) would improve readability.

Test Cases

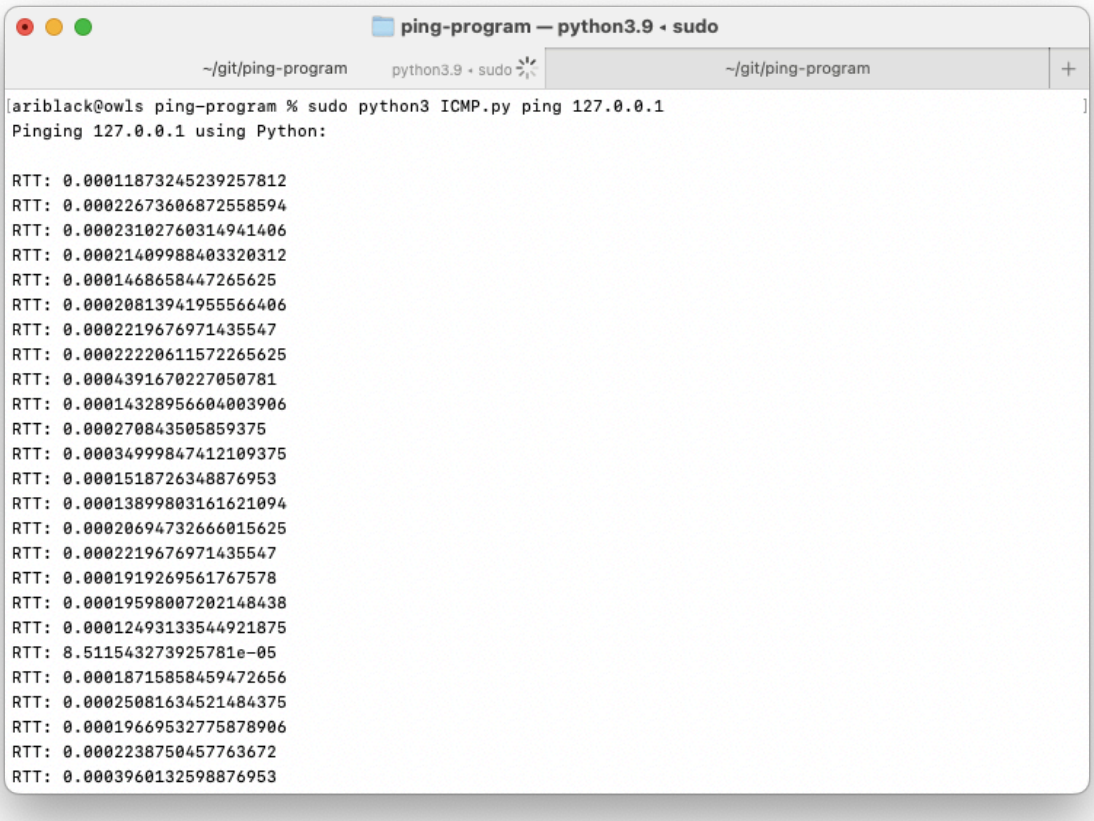
Structure: The following section has a simple structure – first (in bold/italics) is a line containing the relative client-side command used to run the relevant function in *ICMP.py*. (That is, each command is truncated for better readability. In order to actually run, each command needs to begin with *sudo python3 ICMP.py*, which is omitted below. The full commands are visible in their associated screenshots.) Note that commands are separated by |.

Below the command line is a brief description of the given test cases' functionalities, and below this descriptor is the screenshot showing the program's outputs for the given commands.

Commands:

ping 127.0.0.1

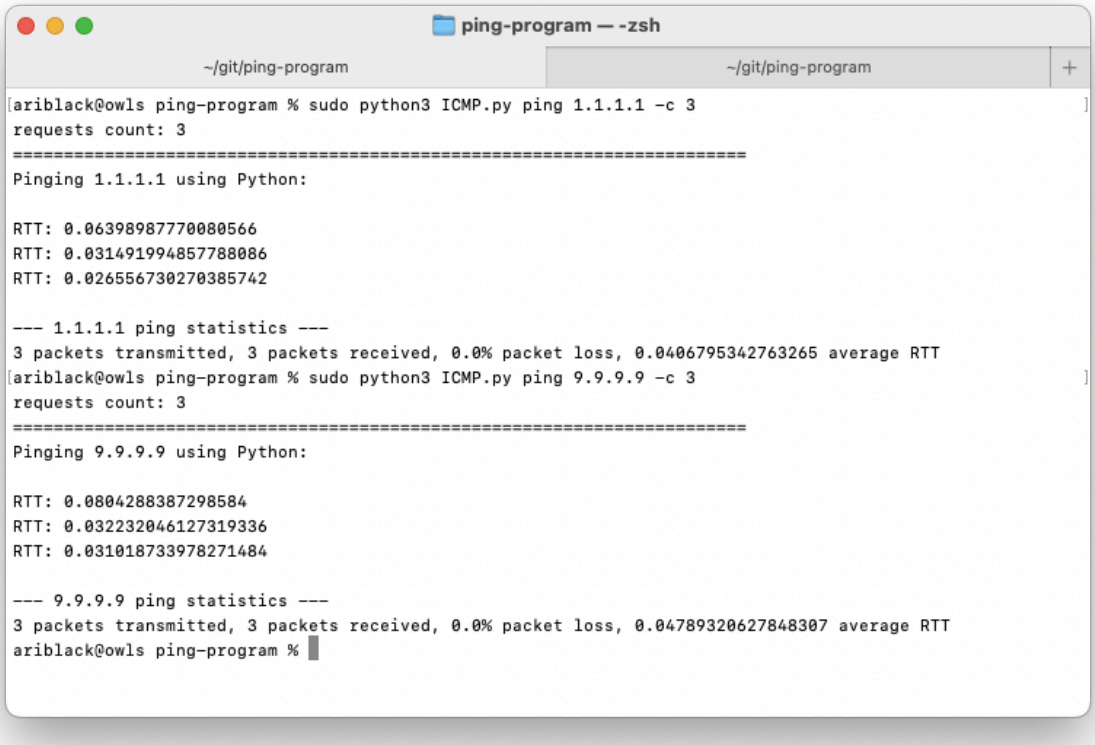
Test ping to localhost with no arguments. Shows that ping works.



ping 1.1.1.1 -c 3 | ping 9.9.9.9 -c 3

Test ping to servers on different continents, specifying an argument for -c. 1.1.1.1 belongs to Cloudflare (located in Australia) and 9.9.9.9 belongs to Quad9 (located in Switzerland).

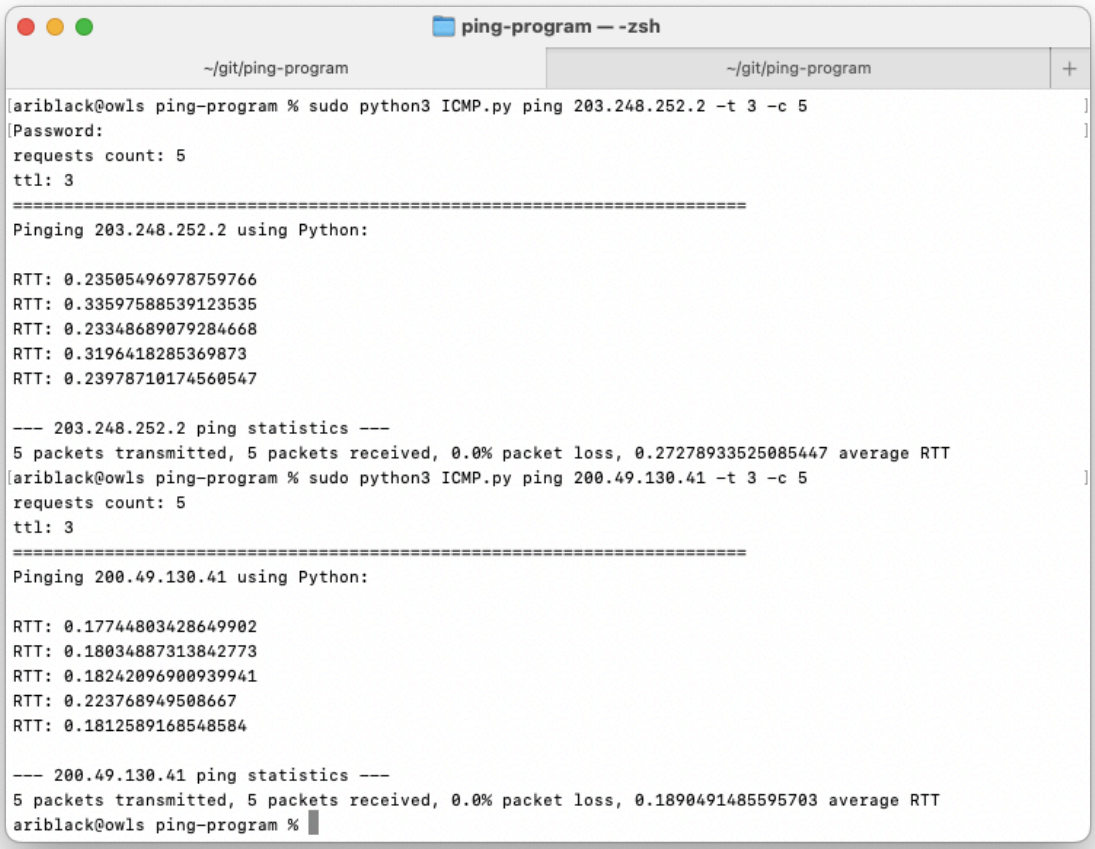
Shows that ping -c works for servers on different continents, and shows that session statistics are calculated and output.



ping 203.248.252.2 -t 3 -c 5 | ping 200.49.130.41 -t 3 -c 5

Test ping to servers on different continents, specifying arguments for -t and (a different) -c. 203.248.252.2 belongs to Korea Telecom (located in Korea) and 200.49.130.41 belongs to Telefonica (located in Argentina).

Shows that ping -t -c works for servers on different continents, and that -c evaluates correctly for various values (5 here, versus 3 above).

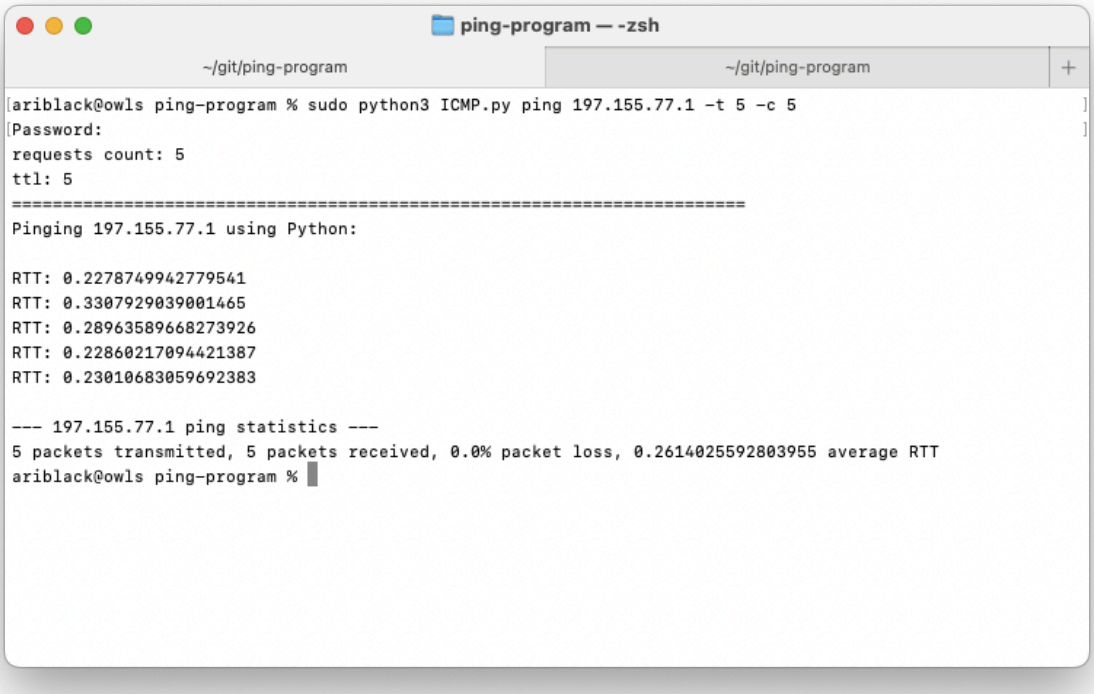



```
ariblack@owls ping-program % sudo python3 ICMP.py ping 203.248.252.2 -t 3 -c 5
Password:
requests count: 5
ttl: 3
=========================================================================
Pinging 203.248.252.2 using Python:

RTT: 0.23505496978759766
RTT: 0.33597588539123535
RTT: 0.23348689079284668
RTT: 0.3196418285369873
RTT: 0.23978710174560547

--- 203.248.252.2 ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss, 0.27278933525085447 average RTT
ariblack@owls ping-program % sudo python3 ICMP.py ping 200.49.130.41 -t 3 -c 5
requests count: 5
ttl: 3
=========================================================================
Pinging 200.49.130.41 using Python:

RTT: 0.17744803428649902
RTT: 0.18034887313842773
RTT: 0.18242096900939941
RTT: 0.223768949508667
RTT: 0.1812589168548584

--- 200.49.130.41 ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss, 0.1890491485595703 average RTT
ariblack@owls ping-program %
```

ping 197.155.77.1 -t 5 -c 5

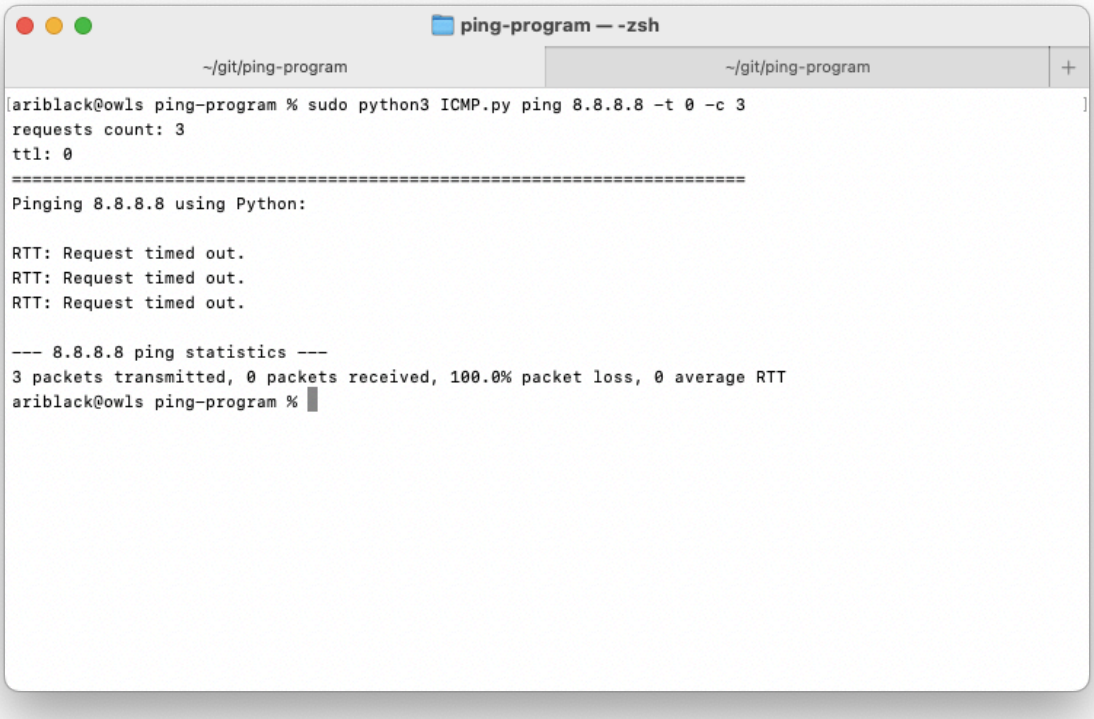
Test ping to servers on all remaining continents (besides Antarctica), specifying arguments for (a different) -t and -c. The IP belongs to Liquid Telecom (located in Kenya).

Shows that ping -t -c works for servers on each continent, and that -t works for various values (5 here, versus 3 above).



ping 8.8.8.8 -t 0 -c 3

Test ping to server setting 0 as the value for -t. Shows that -t correctly changes the timeout value for the ping function.



Known Issues: The code works as expected for all test cases, though this is not to say that there are no bugs or other minor issues present at the time of submission.

Resources

No specific sources were referenced for the development of the Ping program.