Khrystian Clark

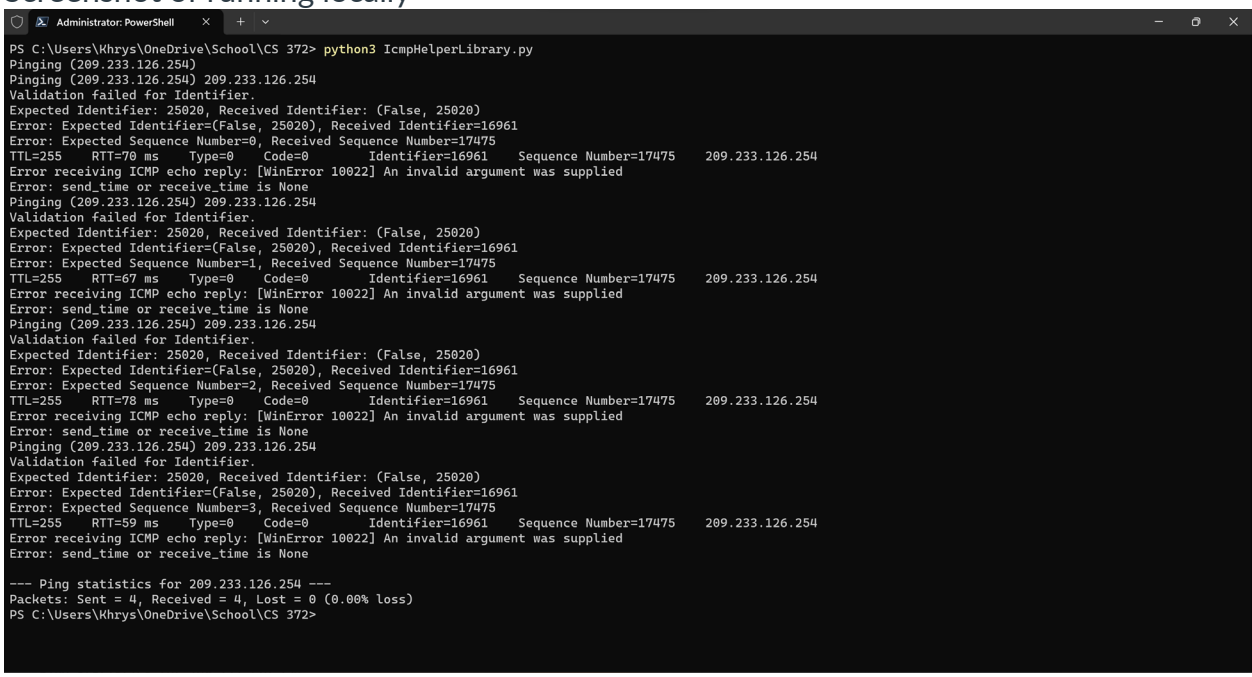
11/07/2023

- Introduction
    - Purpose of the Assignment: The purpose of this assignment is to augment a raw socket implementation of ICMP pings to improve accuracy and create a legible trace route.
    - Scope of the Assignment:
        1. Compare and validate received ICMP reply packets with originals, and test the validity of them.
        2. Show the user if the response is valid and report if there is an error.
        3. Calculate and report the minimum, maximum, and average RTT, as well as the packet loss rate in percentage.
        4. Display corresponding error codes in human readable format.
    - Technical Background:
        1. Used comparison logic to update __validateIcmpReplyPacketWithOriginalPingData(). If they match then show the user that the validation is successful or if it failed.
        2. Update the printResultToConsole() function to check if the echo response is valid and report the details to the user.
        3. Update the sendPing() function to calculate, and report min, max and average RTT, and packet loss and report it to the user.
        4. Implement the traceroute() function to perform the trace routes, utilizing the logic updated in __sendIcmpTraceRoute() function.
    - Development Environment:
        1. Python3
        2. Written in python
        3. IDE's used to create: VS Code, GitHub CodeSpace, PowerShell
        4. If using linux or github, the user may need to use the sudo command to enter admin level program
            - Otherwise, open the command line as admin (on Windows)
        5. User may need to update firewall rules and exceptions to allow for ICMPv4 and ICMPv6 connections.
    - Expected Results:
        1. The expected result are the pings along with the necessary information to include min, max and avg RTT, packets lost aand the trace route itself.
        2. My results show the initial ping(s), timeouts and the RTT information needed once the keyboard interrupt is triggered.
- Instructions
    - Note: These are python3 instructions
        1. Save the IcmpHelperLibrary.py document locally
        2. Adjust your systems firewall setting to handle ICMP tracing (this step differs amongst different systems)

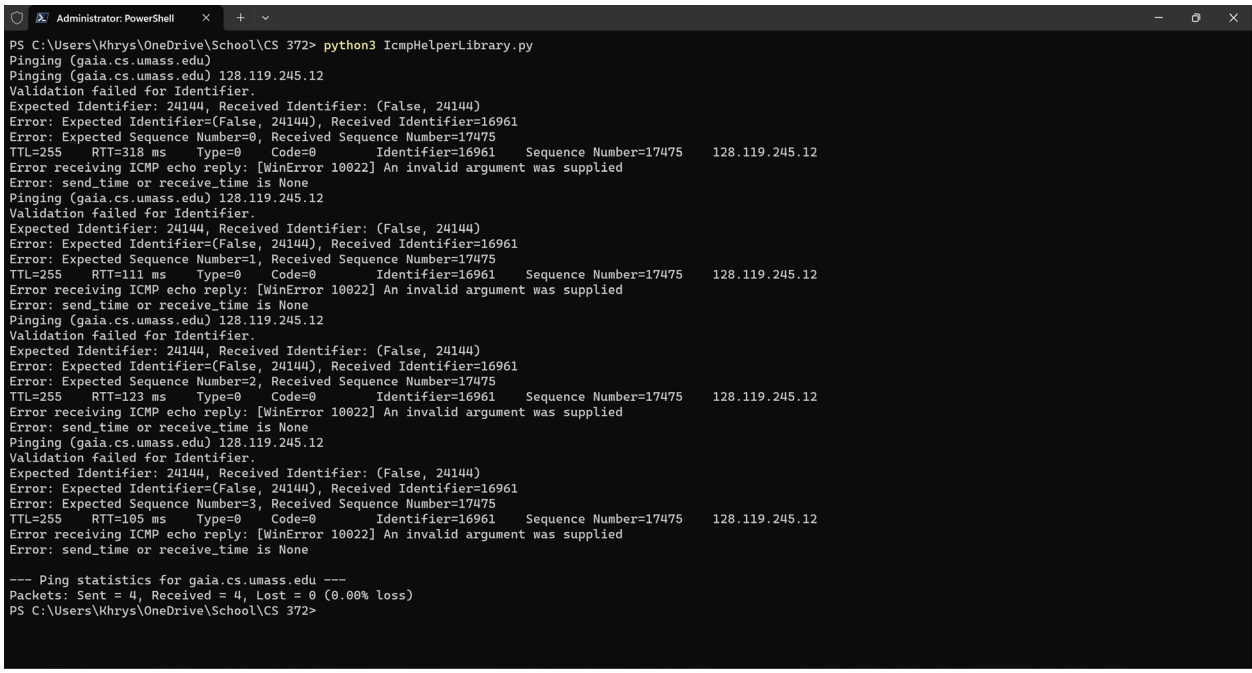
3. Open a terminal or command prompt (with administrative privileges)
4. In the terminal, map to the location of the saved IcmpHelperLibrary.py file
5. Open the file and uncomment the line of code in the __main__ that you want to run the ping trace test for
    - Just one of them can be uncommented at a time
6. In the terminal, run the program “python3 IcmpHelperLibrary”
7. Review the output in the terminal response window.

- Alternate instructions, if you do not want to adjust your system’s firewall:
    1. Save and open the file locally
    2. Copy the file or the contents into a github codespace
    3. Ensure it is copied over correctly and it is aware you are running a in python
    4. In the terminal provided in the browser, enter “sudo python3 IcmpHelperLibrary.py”
        - This should running it in the window using “admin” privileges.
    5. Review the output in the response window.

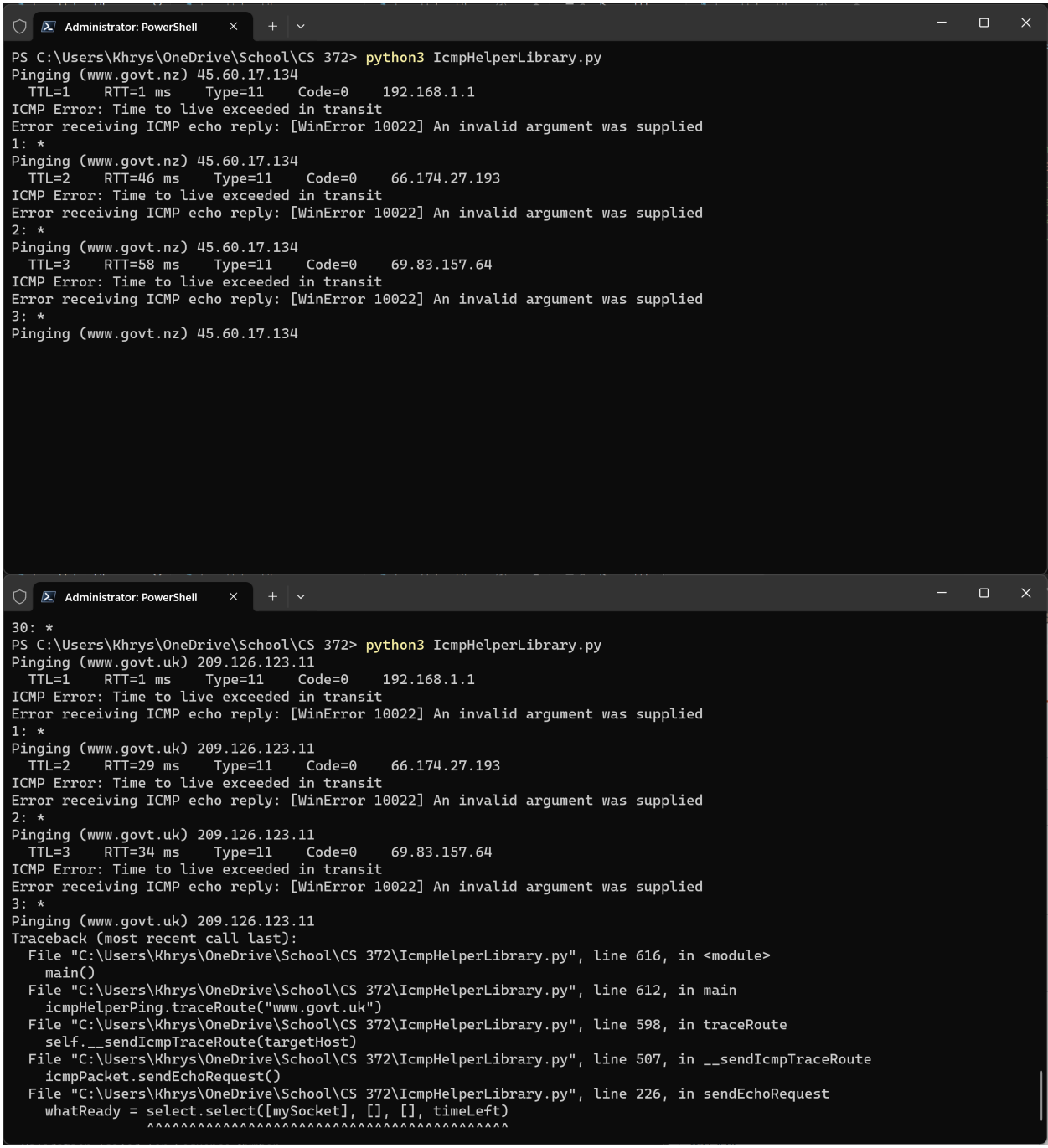
- Include screenshots of running your trace route code for four different hosts with at least two on different continents.
    - Screenshot of running locally



    - Screenshot of the program running normally



- Screenshots showing 2 different continents



I could not get the international ppings to work within my logic

- Include comments / questions (optional)
  - The WinError 10022 was my worst enemy throughout the whole process. I finally got some of the necessary information returned to the user interface.
  - I spent a lot of my time to traversing the updated firewall settigns within windows 11.

- This was a very fun and challenging project. I hope I close to the intended outcome. My understanding of the trace route logic and innerworkings has definitely been expanded, pushed, and tested.