

MSCS-631: Python Lab 4: Lab Report: ICMP

Sandesh Pokharel

University of the Cumberland

Class full name

Dr. Charles Lively

February 2, 2025

ICMP Pinger Lab: Experience and Challenges

Experience

The ICMP Pinger Lab provided me with an opportunity to gain a deeper understanding of the Internet Control Message Protocol (ICMP). The objective was to develop a Python program that sends ICMP Echo Requests (ping) to various hosts and receives ICMP Echo Replies. This allowed me to observe how pings measure network latency, packet loss, and connectivity to remote servers.

I began by reviewing and completing the provided skeleton code. This involved implementing key features, such as extracting the ICMP header from received packets and calculating the round-trip time (RTT). I verified the functionality of my program by first pinging localhost (127.0.0.1). After confirming successful communication, I proceeded to test with servers in different continents, such as google.com (North America), bbc.co.uk (Europe), baidu.com (Asia), and sydney.edu.au (Australia). The program successfully measured RTT and displayed the results for each host.

Testing showed that geographically closer hosts exhibited lower latency, while hosts in distant regions had higher RTT values. These reinforced concepts related to physical distance and network propagation delay.

Additionally, I enhanced the program by handling interruptions (Ctrl + C) gracefully. This change ensured that the program exits with a clean message instead of an error traceback.

Challenges

While completing the assignment, I encountered a few challenges that required troubleshooting and code modifications.

One significant issue was related to permissions. Because the program uses raw sockets, I received a `PermissionError` when trying to run it without elevated privileges. I learned that most operating systems require administrator or root access for raw socket operations to prevent unauthorized network activity. I resolved this issue by running the script with `sudo` on Linux. I also noted that macOS users would need to do the same to avoid permission errors.

Another challenge was related to program termination. By default, stopping the program with `Ctrl + C` triggered a `KeyboardInterrupt` exception, causing the terminal to display an unattractive error message. To address this, I added a `try-except` block to handle the exception gracefully. This allowed the program to display a clean exit message, improving the user experience.

Despite these challenges, the lab was a rewarding experience that enhanced my knowledge of low-level networking, ICMP operations, and cross-platform compatibility considerations.

Screenshot

