

## Program 1 – ICMP Pinger

JJ McCauley  
COSC370  
3/10/2025

### Description

This program utilizes the Python programming language to implement raw sockets to send ICMP Echo Request packets (pings) to a specified host, measuring the RTT for each reply. It computes the internet checksum for error detection, extracts timestamps to calculate RTT and other various metrics (package loss, average RTT, etc.), and allows for large user customization via the command line (hostname, timeout, number of pings). Additionally, the user can stop pinging at any time with Ctrl+C.

### Rough Pseudocode/Design

#### *Global Variables*

ICMP\_ECHO\_REQUEST: The ICMP type for echo requests

rtt\_list: A list to hold all RTTs to calculations at the end

total\_pings: Total number of pings that occurred (in case of user interrupt)

#### *Checksum*

Using the source\_string (header + data of packet), will calculate the 16-bit checksum used for error detection.

- Convert source\_string to bytes if needed
- Convert each 2-byte chunk in source\_string to integer
  - Sum all “chunk integers” up
- If sum has odd length, add the last byte
- Combine overflow bits, compute bitwise complement
- Return the 16-bit checksum

#### *Receiving One Ping*

Using the opened socket, packet ID, timeout number, and destination address (IP), waits for the ICMP Echo Reply (pong), extracting the response header and verified ID, and then calculates the RTT if valid (updating rtt\_list)

- Loop infinitely
  - Wait for incoming packet from socket
    - If over timeout, then return an error message
  - Record time received (using time.perf\_counter() for precision)
  - Retrieve packet from socket
  - Check packet contents, return with error if improper size
  - Extract & Unpack the ICMP header from packet

- Validate the PCMP response, interpreting error code if present
- Extract timestamp (time sent) and calculate RTT
  - Update rtt\_list
- Return success message containing RTT

### ***Sending One Ping***

Constructs and sends an ICMP Echo Request with a timestamp, computing checksum and sending to its destination

- Construct a dummy header
- Obtain the checksum and put it in the header
- Add a timestamp to the header and send it to the socket

### ***Do One Ping***

Creates a raw socket, sends a ping, waits for a response, records delay, and then closes the socket

### ***Ping***

Main function that will be called when testing, resolves a hostname to an IP, pinging every second, and printing the response time (RTT).

- Obtain the destination from user hostname (parameter)
- For a specified number of pings, call the doOnePing function
  - Print it's returned delay
  - Sleep for 1 second, then ping again
- Once the user interrupts the process or the total number of pings has been achieved, showcase numerous statistics such as packet loss, average RTT, and jitter (among others)

Additionally, a functionality that I added was an argument parser, which allows the user to specify the arguments for the ping() function that is called when running the script. This may make it easier for the user, as they can access the entire functionality of the program without having to ever edit it.