

Lab 3. Ping and Traceroute

Due: 11/08/2022

1. Goal

Write the ping and traceroute programs that work like real ones in Linux. We use ping to check network connectivity and to see whether a remote server is up and running. Ping sends an ICMP echo request packet to the server which will in turn reply an ICMP echo reply packet to the sender. The sender then knows the server is up and running, and measures the elapsed time for the communication. Traceroute is a network diagnostic tool often used to find out the routes from the sender to the receiver. In traceroute, the sender generates a sequence of UDP packets destined to the receiver with TTL gradually increasing starting at 1 in order to discover the intermediate routers. An intermediate router issues an ICMP error message when TTL becomes 0, and return the message back to the sender. This is how the sender discovers the intermediate routers on the route to the destination. **IMPORTANT:** To make an intermediate router or the destination to respond, the ICMP packets generated from your ping and traceroute should be acceptable by actual routers or servers. You need to use **raw sockets** to compose ICMP packets. Make your program display the output as the real ping and traceroute programs.

2. Ping

Build your own ping named `your_CS_ID_ping` with the following options:

`-c count`

Stop after sending (and receiving) count ECHO_RESPONSE packets. If this option is not specified, ping will operate until interrupted.

`-i wait`

Wait wait seconds between sending each packet. The default is to wait for one second between each packet.

`-s packetsize`

Specify the number of data bytes to be sent. The default is 56, which translates into 64 ICMP data bytes when combined with the 8 bytes of ICMP header data.

`-t timeout`

Specify a timeout, in seconds, before ping exits regardless of how many packets have been received.

3. Traceroute

Build your own traceroute named `your_CS_ID_traceroute` with the following options:

`-n`

Print hop addresses numerically rather than symbolically and numerically.

`-q nqueries`

Set the number of probes per ``ttl'' to nqueries.

`-S`

Print a summary of how many probes were not answered for each hop.