

.zip contents

- PingClient.java
- A3Manual.pdf

Program Features:

The PingClient.java is the client-side of the UDP Ping program for CPSC441. It works in combination with a PingServer.java file that's running on a remote server. The PingClient requires the IP address of the PingServer machine, as well as the PingServer's listening port number. The PingClient creates a UDP socket and sends 10 data packets (one second between each packet) to the PingServer to simulate an ICMP Ping program. Each packet contains a data payload containing the message "PING", the packet sequence number (0-9), and a timestamp. The PingServer is written to simulate network traffic loss and delay over the Internet, and as such may not return all 10 packets initially received from PingClient. PingClient reports successfully received packets and their round-trip time (RTT), and when the packets are dropped. PingClient then terminates after outputting a statistical summary reporting the average RTT for all 10 packets, the max RTT, and the minimum RTT of the 10 packets.

How to Run:

- 1) Ensure that PingServer.java is running on a remote server. For this exercise, PingServer was tested on UCalgary CPSC Linux Servers. Note the port number the process is running on. For this exercise, PingClient was tested on a local machine.
- 2) Execute PingClient.java locally by compiling "**javac PingClient.java**" and then running "**java PingClient <Server IP Address> <PingServer Port Number>**" as command line arguments. Testing on remote CPSC Linux servers was done using CPSC IP Address: 136.159.5.25 (run command "*ifconfig*" on the PingServer remote server to find this address), and port number 8787.

Known Issues:

To date, no known issues have been found with the application. All assignment requirements have been met.

Screenshots:

```
hechen@csx:~$ java PingServer 8787
Received from 108.173.36.46: PING 0 1605560555671
  Reply sent.
Received from 108.173.36.46: PING 1 1605560555773
  Reply sent.
Received from 108.173.36.46: PING 2 1605560555937
  Reply not sent.
Received from 108.173.36.46: PING 3 1605560556943
  Reply not sent.
Received from 108.173.36.46: PING 4 1605560557948
  Reply not sent.
Received from 108.173.36.46: PING 5 1605560558953
  Reply sent.
Received from 108.173.36.46: PING 6 1605560559170
  Reply sent.
Received from 108.173.36.46: PING 7 1605560559374
  Reply sent.
Received from 108.173.36.46: PING 8 1605560559442
  Reply not sent.
Received from 108.173.36.46: PING 9 1605560560443
  Reply sent.
```



Figure 1: PingServer output response during PingClient execution. Note that we are currently logged into the UCalgary CPSC Linux server.

```
harrisonchen@NightNinja 441A % javac PingClient.java
harrisonchen@NightNinja 441A % java PingClient 136.159.5.25 8787
Sending: PING 0 1605560555671

Successful Return: PING 0 - RTT: 101 ms

Sending: PING 1 1605560555773

Successful Return: PING 1 - RTT: 164 ms

Sending: PING 2 1605560555937

Dropped packet 2

Sending: PING 3 1605560556943

Dropped packet 3

Sending: PING 4 1605560557948

Dropped packet 4

Sending: PING 5 1605560558953

Successful Return: PING 5 - RTT: 216 ms

Sending: PING 6 1605560559170

Successful Return: PING 6 - RTT: 204 ms

Sending: PING 7 1605560559374

Successful Return: PING 7 - RTT: 68 ms

Sending: PING 8 1605560559442

Dropped packet 8

Sending: PING 9 1605560560443

Successful Return: PING 9 - RTT: 168 ms

RTT Analysis Results - Average: 492 ms, Maximum RTT: 216 ms, Minimum RTT: 68 ms
```

Figure 2: PingClient behaviour during local execution. Note that for the dropped packets, we add 1000 ms to the overall RTT average time calculation, since PingClient waits that long before deciding to move on.

```
Sending: PING 7 1605560559374

Successful Return: PING 7 - RTT: 68 ms

Sending: PING 8 1605560559442

Dropped packet 8
```

Figure 3: PingClient packet data sample. Note that outbound PING packets have the “PING <sequence number> <timestamp>” data format as required. A successful PING response outputs the RTT of the packet, while a dropped packet response provides a drop response.