

How I construct the Program:

DoOnePing: I search online to open a socket with icmp and found the code

ReceiveOnePing:

1. I extract the ICMP header. ICMP is piggyback with IP packet therefore ICMP header starts at 20<sup>th</sup> bytes (length of IP header) and ends at 28<sup>th</sup> bytes (length of ICMP header given on assignment pdf).

2. In sendOnePing, it is known about the format of the struct, bbHHh, and time is attach after the header (data section) in a d format. Thus, from the following information, we can unpack the structure easily and retrieve the header and data (time).

3. Finally can compute rtt\_min by using python min function on rtt\_min and rtt, rtt\_max by using python max function on rtt\_max and rtt, and rtt\_sum by adding rtt\_sum + rtt. Also increment rtt\_count to compute the number of packet received

A. Test your client by sending packets to localhost, that is, 127.0.0.1.

```
Pinging 127.0.0.1 using Python:
0.00100016593933
0.0
0.000999927520752
0.0
0.0
0.0
0.0
0.0
0.0
0.000999927520752
0.0
--- localhost ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max: 0.0/0.000300002098083/0.00100016593933

Process finished with exit code 0
```

B. Test your client by sending packets to stonybrook.edu or cs.stonybrook.edu

```
Pinging 129.49.2.176 using Python:
0.00299978256226
0.00399994850159
0.00399994850159
0.00299978256226
0.00300002098083
0.00499987602234
0.00899982452393
0.00499987602234
0.00400018692017
0.0019998550415
--- stonybrook.edu ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max: 0.0019998550415/0.00419991016388/0.00899982452393

Process finished with exit code 0
```

Stonybrook.edu

C. Select and ping 4 servers; each in a different continent.

Server 1: University of Oxford (United Kingdom, Europe) ox.ac.uk

```
Pinging 151.101.66.216 using Python:
0.00600004196167
0.00600004196167
0.00600004196167
0.00399994850159
0.00399994850159
0.00399994850159
0.00600004196167
0.0160000324249
0.00600004196167
0.00699996948242
--- ox.ac.uk ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max: 0.00399994850159/0.00650000572205/0.0160000324249

Process finished with exit code 0
```

Server 2: Pohang University of Science and Technology (South Korea, Asia) postech.ac.kr

```
Pinging 141.223.5.51 using Python:
0.192000150681
0.27999997139
0.190999984741
0.191999912262
0.202000141144
0.191999912262
0.193000078201
0.194000005722
0.190999984741
0.200000047684
--- postech.ac.kr ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max: 0.190999984741/0.202700018883/0.27999997139

Process finished with exit code 0
```

Server 3: University of Melbourne (Australia) unimelb.edu.au

```
Pinging 43.245.43.59 using Python:
0.0780000686646
0.0789999961853
0.0770001411438
0.0780000686646
0.0759999752045
0.0750000476837
0.0769999027252
0.0789999961853
0.0759999752045
0.0770001411438
--- unimelb.edu.au ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max: 0.0750000476837/0.0772000312805/0.0789999961853

Process finished with exit code 0
```

Server 4: Universidade de Sao Paulo (Brazil, South America) usp.br

```
Pinging 200.144.248.41 using Python:
0.143000125885
0.158999919891
0.186000108719
0.144999980927
0.143999814987
0.144999980927
0.149999856949
0.146000146866
0.144999980927
0.146000146866
--- usp.br ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max: 0.143000125885/0.150900006294/0.186000108719

Process finished with exit code 0
```

D. Explain the differences in minimum round trip time to each of these servers in parts A, B, and C.

In part A, I ping my own machine, this is fast and done almost in an instance (0 second) since packets doesn't need to travel outside my machine.

In part B, I ping the Stonybrook University server from my dorm, in which the server lives on the same network. This has low RTT because packets doesn't need to travel through the Internet.

In part C, the distance affect the RTT the farther the packet needs to travel the higher the RTT will be. In Postech & Universidade de Sao Paulo server, both of them doesn't use CDN, so packets travel to their respective continent (Asia and South America). We can confirm this through the traceroute and a sudden spike in one of the server which indicates travel to a different continent.

```
Command Prompt
C:\Users\bleess>tracert usp.br

Tracing route to usp.br [200.144.248.41]
over a maximum of 30 hops:

  1  6 ms    1 ms    2 ms  172.25.232.3
  2  3 ms    3 ms    3 ms  129.49.7.105
  3  4 ms    5 ms    5 ms  nyc-9208-stonybrook.nysernet.net [199.109.5.129]
  4  16 ms   6 ms    7 ms  I2-NEWY-nyc-9208.nysernet.net [199.109.5.2]
  5  *        *        *    Request timed out.
  6  *        35 ms   *    fourhundredge-0-0-0-16.4079.core2.ashb.net.internet2.edu [163.253.1.3]
  7  32 ms   32 ms   33 ms  fourhundredge-0-0-0-1.4079.core2.atla.net.internet2.edu [163.253.1.135]
  8  33 ms   31 ms   30 ms  fourhundredge-0-0-0-23.4079.core1.atla.net.internet2.edu [163.253.1.104]
  9  32 ms   31 ms   32 ms  fourhundredge-0-0-0-1.4079.core1.jack.net.internet2.edu [163.253.2.33]
 10  38 ms   37 ms   38 ms  64.57.28.62
 11  36 ms   38 ms   40 ms  mia2-mia1.bkb.rnp.br [200.143.252.26]
 12  103 ms  101 ms  101 ms  170.79.213.46
 13  101 ms  101 ms  114 ms  170.79.213.64
 14  145 ms  146 ms  149 ms  170.79.213.61
 15  145 ms  144 ms  143 ms  sp-sp2.bkb.rnp.br [200.143.253.37]
 16  154 ms  148 ms  146 ms  sp-usp.bkb.rnp.br [200.143.255.114]
 17  146 ms  149 ms  146 ms  143.107.151.62
 18  *        *        *    Request timed out.
 19  166 ms  161 ms  147 ms  webhost.uspdigital.usp.br [200.144.248.41]
```

```
Command Prompt
C:\Users\bleess>tracert postech.ac.kr

Tracing route to postech.ac.kr [141.223.5.51]
over a maximum of 30 hops:

  1  1 ms    7 ms    1 ms  172.25.232.3
  2  4 ms    3 ms    7 ms  129.49.7.105
  3  8 ms    4 ms    6 ms  nyc-9208-stonybrook.nysernet.net [199.109.5.129]
  4  5 ms    5 ms    7 ms  nyc32-55a1-nyc32-9208.nysernet.net [199.109.7.202]
  5  11 ms   11 ms   9 ms  syr-55a1-nyc32-55a1.nysernet.net [199.109.7.205]
  6  13 ms   15 ms  13 ms  buf-9208-syr-55a1.nysernet.net [199.109.7.214]
  7  20 ms   20 ms  19 ms  I2-CLEV-buf-9208.nysernet.net [199.109.11.34]
  8  *        24 ms  25 ms  fourhundredge-0-0-0-21.4079.core1.clev.net.internet2.edu [163.253.2.142]
  9  *        26 ms   *    fourhundredge-0-0-0-2.4079.core1.eqch.net.internet2.edu [163.253.1.211]
 10  31 ms   27 ms  28 ms  fourhundredge-0-0-0-2.4079.core1.star.net.internet2.edu [163.253.2.73]
 11  25 ms   27 ms  22 ms  kreonet2-abilene.kreonet.net [134.75.108.45]
 12  181 ms  177 ms  176 ms  rtr1chic-rtr1daej.daej.kreonet2.net [134.75.103.33]
 13  177 ms  176 ms  179 ms  rtr2kreonet2-rtr1.daej.kreonet.net [134.75.103.121]
 14  179 ms  181 ms  180 ms  134.75.8.26
 15  193 ms  195 ms  194 ms  134.75.35.2
 16  192 ms  192 ms  191 ms  141.223.180.2
 17  190 ms  194 ms  192 ms  141.223.253.5
 18  *        *        *    Request timed out.
 19  192 ms  195 ms  191 ms  m.postech.ac.kr [141.223.5.51]
```

University of Oxford server is hosted in Fastly CDN which is hosted in Manhattan, NY. So packet hops a few times.

```
Command Prompt
C:\Users\bless>tracert ox.ac.uk

Tracing route to ox.ac.uk [151.101.194.216]
over a maximum of 30 hops:

  1  3 ms    2 ms    2 ms  172.25.232.2
  2  3 ms    2 ms    3 ms  129.49.7.105
  3  4 ms    6 ms    7 ms  nyc-9208-stonybrook-cdn.nysernet.net [199.109.105.129]
  4  4 ms    6 ms    5 ms  nyc32-55a1-nyc32-9208.cdn.nysernet.net [199.109.107.202]
  5  8 ms    5 ms    5 ms  jfk1.decixny.fastly.net [206.82.104.29]
  6  8 ms    4 ms    7 ms  151.101.194.216

Trace complete.
```

Finally, University of Melbourne server is not hosted in CDN, yet utilize Cloudflare along the route so doesn't need to travel through different router. Packet can travel through the Cloudflare router which means fewer and closer hops.

```
Command Prompt
C:\Users\bless>tracert unimelb.edu.au

Tracing route to unimelb.edu.au [43.245.43.59]
over a maximum of 30 hops:

  1  3 ms    1 ms    2 ms  172.25.232.3
  2  4 ms    4 ms    3 ms  129.49.7.105
  3  4 ms    5 ms    5 ms  nyc-9208-stonybrook-cdn.nysernet.net [199.109.105.129]
  4  5 ms    5 ms    4 ms  nyc32-55a1-nyc32-9208.cdn.nysernet.net [199.109.107.202]
  5  6 ms    5 ms    5 ms  de-cix-new-york.as13335.net [206.82.104.31]
  6  6 ms    6 ms    5 ms  172.70.108.2
  7  5 ms    8 ms    5 ms  172.70.109.69
  8  5 ms    5 ms    5 ms  172.70.109.37
  9  5 ms    5 ms    7 ms  172.70.109.52
 10  *        *        *    Request timed out.
 11  *       77 ms   75 ms  uom.squizedge.net [43.245.43.59]

Trace complete.

C:\Users\bless>
```

4. Currently, the program calculates the round-trip time for each packet and prints it out individually. Modify this to correspond to the way the standard ping program works. You will need to report the minimum, maximum, and average RTTs at the end of all pings from the client.

```
Command Prompt

C:\Users\bless>ping stonybrook.edu

Pinging stonybrook.edu [129.49.2.176] with 32 bytes of data:
Reply from 129.49.2.176: bytes=32 time=3ms TTL=61
Reply from 129.49.2.176: bytes=32 time=3ms TTL=61
Reply from 129.49.2.176: bytes=32 time=3ms TTL=61
Reply from 129.49.2.176: bytes=32 time=6ms TTL=61

Ping statistics for 129.49.2.176:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 6ms, Average = 3ms

C:\Users\bless>
```

```
Pinging www.google.com [142.250.65.164] with 16 bytes of data using Python:
Reply from 142.250.65.164: bytes=16 time=6ms TTL=118
Reply from 142.250.65.164: bytes=16 time=5ms TTL=118
Reply from 142.250.65.164: bytes=16 time=15ms TTL=118
Reply from 142.250.65.164: bytes=16 time=19ms TTL=118
Reply from 142.250.65.164: bytes=16 time=11ms TTL=118
Reply from 142.250.65.164: bytes=16 time=11ms TTL=118
Reply from 142.250.65.164: bytes=16 time=6ms TTL=118
Reply from 142.250.65.164: bytes=16 time=6ms TTL=118
Reply from 142.250.65.164: bytes=16 time=14ms TTL=118
Reply from 142.250.65.164: bytes=16 time=11ms TTL=118

Ping statistics for 142.250.65.164
    Packets: Sent = 10, Received = 10, Lost = 0
Approximate round trip times in milli-seconds:
    Minimum = 5ms, Maximum = 19ms, Average = 11ms

Process finished with exit code 0
```

F. In addition, calculate the packet loss rate (in percentage).

Pinging sunykorea.ac.kr [52.79.41.38] with 16 bytes of data using Python:

Request timed out.

Request timed out.

Request timed out.

Request timed out.

Request timed out.

Request timed out.

Request timed out.

Request timed out.

Request timed out.

Request timed out.

Ping statistics for 52.79.41.38|

Packets: Sent = 10, Received = 0, Lost = 10 (100% loss)

Process finished with exit code 0