

CSUF Fall 2021

CPSC 471-02 Project #2

Professor: Linh Trinh

Required Software

You are required to use Python 3.9.7 or newer for this project.

Project Submission

You are responsible for the content of your submitted file. Double-check your submission in Canvas to ensure that the submitted file is not corrupted, and it is readable and it contains all required files and their contents are what you intend to submit. Submitting a wrong file (from another class for example) is not a valid excuse.

Submit **one zip file** using naming convention: **firstnamelastname_p2.zip** in Canvas.

The zip file shall contain the following files:

- (1) PDF report file using naming convention **firstnamelastname_p2.pdf** (must be a pdf document, not Word).
- (2) Python UDP client code file from Part 1 **xx1.py**
- (3) Python UDP server code file from Part 2 **xx2.py**
- (4) Python UDP server code file from Part 3 **xx3.py**
- (5) Python Heartbeat client code file **xx4c.py**
- (6) Python Heartbeat server code file **xx4s.py**

where xx = initials (the two characters representing the first character of your first and last name).

Part 1: UDP Pinger with No Delay and No Loss

In this project, you will learn the basics of socket programming for UDP in Python. You will learn how to send and receive datagram packets using UDP sockets and, how to set a proper socket timeout. Throughout the lab, you will gain familiarity with a Ping application and its usefulness in computing statistics such as packet loss rate.

You will first study a simple Internet ping server written in the Python and implement a corresponding client. The functionality provided by these programs is like the functionality provided by standard ping programs available in modern operating systems. However, these programs use a simpler protocol, UDP, rather than the standard Internet Control Message Protocol (ICMP)

to communicate with each other. The ping protocol allows a client machine to send a packet of data to a remote machine, and have the remote machine return the data back to the client unchanged (an action referred to as echoing). Among other uses, the ping protocol allows hosts to determine round-trip times to other machines.

You are given the complete code for the Ping server in the next subsection. Your task is to write the UDP Ping client.

Server Code

You are provided with the following code which fully implements a ping server. You need to run this code before running your client program. Your client code in this portion of the project will be graded using this exact same server code (do not modify it for this portion of the project).

You should study this code carefully, as it will help you write your ping client code.

```
# udppingserver_no_loss.py
from socket import *
# Create a UDP socket
serverSocket = socket(AF_INET, SOCK_DGRAM)
# Assign IP address and port number to socket
serverSocket.bind('', 12000)
while True:
    # Receive the client packet along with the address it is coming from
    message, address = serverSocket.recvfrom(1024)
    # The server responds
    serverSocket.sendto(message, address)
```

The server sits in an infinite loop listening for incoming UDP packets. When a packet comes in, the server simply sends it back to the client.

Client Code

Your task is to implement the client program as explained below.

The client should **send a specified number of pings to the server**. Because UDP is an unreliable protocol, a packet sent from the client to the server may be lost in the network, or vice versa. For this reason, **the client cannot wait indefinitely for a reply to a ping message**. You should get the client **wait up to one second for a reply**; if no reply is received within one second, your client program should assume that the packet was lost during transmission across the network. You will need to look up the Python documentation to find out how to set the timeout value on a datagram socket.

Client Code Requirements

The client program should:

- send the ping message using UDP
- print the response message from server if any was received

- calculate and print the round-trip time (RTT), in milliseconds, of each packet if the server responses
- otherwise, print “Request timed out”
- provide a summary report at the end (of all pings) which includes:
 - minimum RTT in milliseconds,
 - maximum RTT in milliseconds,
 - average RTT in milliseconds,
 - percentage packet loss rate

You should run the `udppingserver_no_loss.py` on your machine and test your client by sending packets to the localhost.

Ping Message Format

The client ping message is a one line, consisting of ASCII characters in the following format:

`seq sequence_number date_and_time`

where `sequence_number` starts at 1 and progresses to total number of pings for each successive ping message sent by the client, and `time` is the time when the client sends the message.

Refer to the Appendix section on the last page for sample.

What to Hand in

PDF file report

Create a section called **Part 1 – UDP Pinger with No Delay and No Loss**. Include the followings:

- (1) Describe the operation of your UDP Pinger, for example how it works.
- (2) Explain how to specify the timeout value for a datagram socket. Provide an example.
- (3) Explain how to run your code, i.e., command line and any applicable parameter(s)
 - a. Include run-time screen captures for a sequence consists of 10 pings

Refer to the Appendix section on the last page for sample.

- (4) Include your Python code listing:
 - a. Include as text the listing of your Python code.
Please use consolas font size 10 or equivalent monospace font. The use of these monospace font is to clearly show indentations in your code.

Submission zip file

include your Python client code file **xx1.py**

where `xx` = initials (the two characters representing the first character of your first and last name).

Part 2: UDP Pinger with Delays

Delays

Our experiment so far has been on a local host running both server and client programs, and therefore we saw zero delays. In this portion of the project, you are asked to modify the server code to simulate random RTT delays ranging from 10ms to 40ms.

Hint: Create a variable which holds a randomized integer to determine the delay amount.

What to Hand in

PDF file report

Create a section called **Part 2 – UDP Pinger with Delays**. Include the followings:

- (1) Describe the operation of your UDP Ping Server and explain how it simulates 10ms to 40ms RTT delays.
- (2) Explain how to run your code, i.e., command line and any applicable parameter(s)
 - a. Include run-time screen captures for a sequence consists of 10 pingsRefer to the Appendix section on the last page for sample.
- (3) Include your Python code listing of your UDP Ping Server with 10ms to 40ms RTT delays:
 - a. Include as text the listing of your Python code.
Please use consolas font size 10 or equivalent monospace font. The use of these monospace font is to clearly show indentations in your code.

Submission zip file

include your Python UDP Ping Server with Delays code file **xx2.py**

where xx = initials (the two characters representing the first character of your first and last name).

Part 3: UDP Pinger with Delays and Packet Losses

Packet Loss Injection

UDP provides applications with an unreliable transport service. Messages may get lost in the network due to router queue overflows, faulty hardware, or some other reasons. Because packet loss is rare or even non-existent in typical campus or home networks, you are asked to modify the server code in this portion of the project to inject artificial losses to simulate the effects of network packet loss.

Hint: Create a variable which holds a randomized integer to determine whether a particular incoming packet is lost or not.

What to Hand in

PDF file report

Create a section called **Part 3 – UDP Pinger with Delays and Packet Losses**. Include the followings:

- (1) Describe the operation of your UDP Ping Server and explain how it simulates delays between 10ms and 40ms, with 20% packet losses.
- (2) Explain how to run your code, i.e., command line and any applicable parameter(s)
 - a. Include run-time screen captures for a sequence consists of 100 pings

Refer to the Appendix section on the last page for sample.

- (3) Include your Python code listing of your UDP Ping Server with delays between 10ms and 40ms, and 20% packet losses:
 - a. Include as text the listing of your Python code.
Please use consolas font size 10 or equivalent monospace font. The use of these monospace font is to clearly show indentations in your code.

Submission zip file

include your Python UDP Ping Server with Delay and Losses code file **xx3.py**

where xx = initials (the two characters representing the first character of your first and last name).

Part 4: HeartBeat Monitor Using Python

Another similar application to the UDP Ping would be the UDP Heartbeat. The Heartbeat can be used to check if an application is up and running on the client side and to report one-way packet loss. The client continuously sends a message acting as a heartbeat in the UDP packet to the server, which is monitoring the heartbeat (i.e., the UDP packets) of the client. Upon receiving the packets, the server calculates the time difference. If the heartbeat packets are missing for some specified time interval, the server can assume that the client application has stopped working.

Implement the UDP Heartbeat (both client and server). You are asked to create both the server and client programs.

Use the following file naming convention:

- **xx4c.py** for client
- **xx4s.py** for server

where xx = initials (the two characters representing the first character of your first and last name).

The client program sends a ping message to the server using UDP every 5 seconds.

The server program monitors if a ping is received from the client. If the ping from the client was absent for more than 10 seconds, it prints the message “No pulse after 10 seconds. Server quits”.

What to Hand in

PDF file report

Create a section called **Part 4 – UDP Heartbeat Monitor**. This portion of your report includes:

- (1) Instructions on how to run the code, i.e., command line and any applicable parameter(s) for the client and the server programs
- (2) Run-time screen capture showing:
 - a. the client sends heartbeat pings to the server every 5 seconds.
 - b. server prints the received heartbeat pings from the client, and the time interval.
 - c. server detects absence of client heartbeat and quits.

Refer to the Appendix section on the last page for sample.

- (3) Python code listing:
 - a. Include as text, the client program listing.
 - b. Include as text, the server program listing.

Please use consolas font size 10 or equivalent monospace fonts. The use of these monospace font is to clearly show indentations in your code.

Submission zip file

include both your Python client and server code files **xx4c.py** and **xx4s.py**

where xx = initials (the two characters representing the first character of your first and last name).

Appendix

Part 1: Sequence of 10 Pings with No Delay and No Packet Loss

```
Ping 1: host 127.0.0.1 replied: seq 1 Tue Sep 28 23:19:27 2021, RTT = 0.00 ms
Ping 2: host 127.0.0.1 replied: seq 2 Tue Sep 28 23:19:27 2021, RTT = 0.00 ms
Ping 3: host 127.0.0.1 replied: seq 3 Tue Sep 28 23:19:27 2021, RTT = 0.00 ms
Ping 4: host 127.0.0.1 replied: seq 4 Tue Sep 28 23:19:27 2021, RTT = 0.00 ms
Ping 5: host 127.0.0.1 replied: seq 5 Tue Sep 28 23:19:27 2021, RTT = 0.00 ms
Ping 6: host 127.0.0.1 replied: seq 6 Tue Sep 28 23:19:27 2021, RTT = 0.00 ms
Ping 7: host 127.0.0.1 replied: seq 7 Tue Sep 28 23:19:27 2021, RTT = 0.00 ms
Ping 8: host 127.0.0.1 replied: seq 8 Tue Sep 28 23:19:27 2021, RTT = 0.00 ms
Ping 9: host 127.0.0.1 replied: seq 9 Tue Sep 28 23:19:27 2021, RTT = 0.00 ms
Ping 10: host 127.0.0.1 replied: seq 10 Tue Sep 28 23:19:27 2021, RTT = 0.00 ms
Min RTT = 0.00 ms
Max RTT = 0.00 ms
Avg RTT = 0.00 ms
Packet lost = 0.00 %
```

Part 2: Sequence of 10 Pings with 10ms-40ms Delays and No Packet Loss

```
Ping 1: host 127.0.0.1 replied: seq 1 Tue Sep 28 23:23:52 2021, RTT = 15.64 ms
Ping 2: host 127.0.0.1 replied: seq 2 Tue Sep 28 23:23:52 2021, RTT = 53.41 ms
Ping 3: host 127.0.0.1 replied: seq 3 Tue Sep 28 23:23:52 2021, RTT = 31.26 ms
Ping 4: host 127.0.0.1 replied: seq 4 Tue Sep 28 23:23:52 2021, RTT = 31.23 ms
Ping 5: host 127.0.0.1 replied: seq 5 Tue Sep 28 23:23:52 2021, RTT = 37.83 ms
Ping 6: host 127.0.0.1 replied: seq 6 Tue Sep 28 23:23:52 2021, RTT = 15.63 ms
Ping 7: host 127.0.0.1 replied: seq 7 Tue Sep 28 23:23:52 2021, RTT = 31.25 ms
Ping 8: host 127.0.0.1 replied: seq 8 Tue Sep 28 23:23:52 2021, RTT = 31.25 ms
Ping 9: host 127.0.0.1 replied: seq 9 Tue Sep 28 23:23:52 2021, RTT = 37.80 ms
Ping 10: host 127.0.0.1 replied: seq 10 Tue Sep 28 23:23:52 2021, RTT = 15.64 ms
Min RTT = 15.63 ms
Max RTT = 53.41 ms
Avg RTT = 30.09 ms
Packet lost = 0.00 %
```

Part 3: Sequence of 100 Pings with 10ms-40ms Delays and 20% Packet Loss

```
Ping 1: host 127.0.0.1 replied: seq 1 Tue Sep 28 22:18:21 2021, RTT = 31.25 ms
Ping 2: timed out, message was lost
Ping 3: timed out, message was lost
Ping 4: host 127.0.0.1 replied: seq 4 Tue Sep 28 22:18:23 2021, RTT = 37.80 ms
Ping 5: host 127.0.0.1 replied: seq 5 Tue Sep 28 22:18:23 2021, RTT = 15.63 ms
Ping 6: host 127.0.0.1 replied: seq 6 Tue Sep 28 22:18:23 2021, RTT = 31.25 ms
Ping 7: timed out, message was lost
Ping 8: timed out, message was lost
Ping 9: timed out, message was lost
Ping 10: host 127.0.0.1 replied: seq 10 Tue Sep 28 22:18:26 2021, RTT = 31.25 ms
Ping 11: host 127.0.0.1 replied: seq 11 Tue Sep 28 22:18:26 2021, RTT = 53.44 ms
Ping 12: host 127.0.0.1 replied: seq 12 Tue Sep 28 22:18:26 2021, RTT = 31.25 ms
Ping 13: host 127.0.0.1 replied: seq 13 Tue Sep 28 22:18:26 2021, RTT = 37.81 ms
Ping 14: host 127.0.0.1 replied: seq 14 Tue Sep 28 22:18:26 2021, RTT = 15.63 ms
Ping 15: host 127.0.0.1 replied: seq 15 Tue Sep 28 22:18:26 2021, RTT = 31.25 ms
Ping 16: host 127.0.0.1 replied: seq 16 Tue Sep 28 22:18:26 2021, RTT = 53.44 ms
Ping 17: timed out, message was lost
Ping 18: host 127.0.0.1 replied: seq 18 Tue Sep 28 22:18:27 2021, RTT = 31.28 ms
Ping 19: host 127.0.0.1 replied: seq 19 Tue Sep 28 22:18:27 2021, RTT = 46.87 ms
Ping 20: host 127.0.0.1 replied: seq 20 Tue Sep 28 22:18:27 2021, RTT = 22.13 ms
Ping 21: host 127.0.0.1 replied: seq 21 Tue Sep 28 22:18:27 2021, RTT = 15.65 ms
Ping 22: timed out, message was lost
Ping 23: host 127.0.0.1 replied: seq 23 Tue Sep 28 22:18:28 2021, RTT = 31.23 ms
Ping 24: host 127.0.0.1 replied: seq 24 Tue Sep 28 22:18:28 2021, RTT = 31.28 ms
Ping 25: timed out, message was lost
Ping 26: host 127.0.0.1 replied: seq 26 Tue Sep 28 22:18:29 2021, RTT = 22.18 ms
Ping 27: host 127.0.0.1 replied: seq 27 Tue Sep 28 22:18:29 2021, RTT = 31.25 ms
Ping 28: timed out, message was lost
Ping 29: host 127.0.0.1 replied: seq 29 Tue Sep 28 22:18:30 2021, RTT = 46.88 ms
Ping 30: host 127.0.0.1 replied: seq 30 Tue Sep 28 22:18:30 2021, RTT = 22.17 ms
Ping 31: host 127.0.0.1 replied: seq 31 Tue Sep 28 22:18:30 2021, RTT = 31.24 ms
Ping 32: host 127.0.0.1 replied: seq 32 Tue Sep 28 22:18:30 2021, RTT = 31.26 ms
Ping 33: timed out, message was lost
Ping 34: host 127.0.0.1 replied: seq 34 Tue Sep 28 22:18:31 2021, RTT = 37.81 ms
Ping 35: host 127.0.0.1 replied: seq 35 Tue Sep 28 22:18:31 2021, RTT = 31.26 ms
Ping 36: timed out, message was lost
Ping 37: host 127.0.0.1 replied: seq 37 Tue Sep 28 22:18:32 2021, RTT = 31.25 ms
Ping 38: host 127.0.0.1 replied: seq 38 Tue Sep 28 22:18:32 2021, RTT = 15.63 ms
Ping 39: timed out, message was lost
Ping 40: host 127.0.0.1 replied: seq 40 Tue Sep 28 22:18:33 2021, RTT = 53.42 ms
Ping 41: host 127.0.0.1 replied: seq 41 Tue Sep 28 22:18:33 2021, RTT = 15.61 ms
Ping 42: host 127.0.0.1 replied: seq 42 Tue Sep 28 22:18:33 2021, RTT = 53.43 ms
Ping 43: host 127.0.0.1 replied: seq 43 Tue Sep 28 22:18:34 2021, RTT = 31.23 ms
Ping 44: host 127.0.0.1 replied: seq 44 Tue Sep 28 22:18:34 2021, RTT = 31.28 ms
Ping 45: host 127.0.0.1 replied: seq 45 Tue Sep 28 22:18:34 2021, RTT = 37.81 ms
Ping 46: host 127.0.0.1 replied: seq 46 Tue Sep 28 22:18:34 2021, RTT = 31.23 ms
Ping 47: host 127.0.0.1 replied: seq 47 Tue Sep 28 22:18:34 2021, RTT = 31.28 ms
Ping 48: timed out, message was lost
Ping 49: timed out, message was lost
Ping 50: host 127.0.0.1 replied: seq 50 Tue Sep 28 22:18:36 2021, RTT = 15.62 ms
Ping 51: timed out, message was lost
Ping 52: host 127.0.0.1 replied: seq 52 Tue Sep 28 22:18:37 2021, RTT = 37.81 ms
```



```
Ping 51: timed out, message was lost
Ping 52: host 127.0.0.1 replied: seq 52 Tue Sep 28 22:18:37 2021, RTT = 37.81 ms
Ping 53: host 127.0.0.1 replied: seq 53 Tue Sep 28 22:18:37 2021, RTT = 15.63 ms
Ping 54: host 127.0.0.1 replied: seq 54 Tue Sep 28 22:18:37 2021, RTT = 15.63 ms
Ping 55: host 127.0.0.1 replied: seq 55 Tue Sep 28 22:18:37 2021, RTT = 31.25 ms
Ping 56: host 127.0.0.1 replied: seq 56 Tue Sep 28 22:18:37 2021, RTT = 37.78 ms
Ping 57: host 127.0.0.1 replied: seq 57 Tue Sep 28 22:18:37 2021, RTT = 31.26 ms
Ping 58: timed out, message was lost
Ping 59: host 127.0.0.1 replied: seq 59 Tue Sep 28 22:18:38 2021, RTT = 15.61 ms
Ping 60: host 127.0.0.1 replied: seq 60 Tue Sep 28 22:18:38 2021, RTT = 37.78 ms
Ping 61: host 127.0.0.1 replied: seq 61 Tue Sep 28 22:18:38 2021, RTT = 46.90 ms
Ping 62: timed out, message was lost
Ping 63: host 127.0.0.1 replied: seq 63 Tue Sep 28 22:18:39 2021, RTT = 31.28 ms
Ping 64: host 127.0.0.1 replied: seq 64 Tue Sep 28 22:18:39 2021, RTT = 57.12 ms
Ping 65: timed out, message was lost
Ping 66: host 127.0.0.1 replied: seq 66 Tue Sep 28 22:18:40 2021, RTT = 15.60 ms
Ping 67: host 127.0.0.1 replied: seq 67 Tue Sep 28 22:18:40 2021, RTT = 15.65 ms
Ping 68: host 127.0.0.1 replied: seq 68 Tue Sep 28 22:18:40 2021, RTT = 15.63 ms
Ping 69: timed out, message was lost
Ping 70: host 127.0.0.1 replied: seq 70 Tue Sep 28 22:18:41 2021, RTT = 37.78 ms
Ping 71: host 127.0.0.1 replied: seq 71 Tue Sep 28 22:18:41 2021, RTT = 31.25 ms
Ping 72: host 127.0.0.1 replied: seq 72 Tue Sep 28 22:18:41 2021, RTT = 46.88 ms
Ping 73: host 127.0.0.1 replied: seq 73 Tue Sep 28 22:18:41 2021, RTT = 22.17 ms
Ping 74: host 127.0.0.1 replied: seq 74 Tue Sep 28 22:18:41 2021, RTT = 46.89 ms
Ping 75: host 127.0.0.1 replied: seq 75 Tue Sep 28 22:18:41 2021, RTT = 53.42 ms
Ping 76: host 127.0.0.1 replied: seq 76 Tue Sep 28 22:18:41 2021, RTT = 31.25 ms
Ping 77: host 127.0.0.1 replied: seq 77 Tue Sep 28 22:18:41 2021, RTT = 15.63 ms
Ping 78: host 127.0.0.1 replied: seq 78 Tue Sep 28 22:18:41 2021, RTT = 15.61 ms
Ping 79: host 127.0.0.1 replied: seq 79 Tue Sep 28 22:18:41 2021, RTT = 15.61 ms
Ping 80: timed out, message was lost
Ping 81: timed out, message was lost
Ping 82: host 127.0.0.1 replied: seq 82 Tue Sep 28 22:18:44 2021, RTT = 31.26 ms
Ping 83: host 127.0.0.1 replied: seq 83 Tue Sep 28 22:18:44 2021, RTT = 37.81 ms
Ping 84: host 127.0.0.1 replied: seq 84 Tue Sep 28 22:18:44 2021, RTT = 31.25 ms
Ping 85: host 127.0.0.1 replied: seq 85 Tue Sep 28 22:18:44 2021, RTT = 31.25 ms
Ping 86: host 127.0.0.1 replied: seq 86 Tue Sep 28 22:18:44 2021, RTT = 22.33 ms
Ping 87: host 127.0.0.1 replied: seq 87 Tue Sep 28 22:18:44 2021, RTT = 31.25 ms
Ping 88: host 127.0.0.1 replied: seq 88 Tue Sep 28 22:18:44 2021, RTT = 31.24 ms
Ping 89: host 127.0.0.1 replied: seq 89 Tue Sep 28 22:18:44 2021, RTT = 37.80 ms
Ping 90: host 127.0.0.1 replied: seq 90 Tue Sep 28 22:18:44 2021, RTT = 31.25 ms
Ping 91: timed out, message was lost
Ping 92: host 127.0.0.1 replied: seq 92 Tue Sep 28 22:18:45 2021, RTT = 15.65 ms
Ping 93: host 127.0.0.1 replied: seq 93 Tue Sep 28 22:18:45 2021, RTT = 53.42 ms
Ping 94: host 127.0.0.1 replied: seq 94 Tue Sep 28 22:18:45 2021, RTT = 31.24 ms
Ping 95: host 127.0.0.1 replied: seq 95 Tue Sep 28 22:18:45 2021, RTT = 15.65 ms
Ping 96: host 127.0.0.1 replied: seq 96 Tue Sep 28 22:18:45 2021, RTT = 53.42 ms
Ping 97: host 127.0.0.1 replied: seq 97 Tue Sep 28 22:18:45 2021, RTT = 31.23 ms
Ping 98: host 127.0.0.1 replied: seq 98 Tue Sep 28 22:18:45 2021, RTT = 31.28 ms
Ping 99: timed out, message was lost
Ping 100: host 127.0.0.1 replied: seq 100 Tue Sep 28 22:18:46 2021, RTT = 15.60 ms
Min RTT = 15.60 ms
Max RTT = 57.12 ms
Avg RTT = 31.23 ms
Packet lost = 23.00 %
```

Part 4: Heartbeat Monitor Run-Time Samples

Client Side

```
heartbeat pulse 1
heartbeat pulse 2
heartbeat pulse 3
heartbeat pulse 4
heartbeat pulse 5
heartbeat pulse 6
heartbeat pulse 7
heartbeat pulse 8
heartbeat pulse 9
heartbeat pulse 10
heartbeat pulse 11
heartbeat pulse 12
heartbeat pulse 13
heartbeat pulse 14
heartbeat pulse 15
```

Server Side

```
Server received heartbeat pulse 3 Pulse interval was 6 seconds
Server received heartbeat pulse 4 Pulse interval was 5 seconds
Server received heartbeat pulse 5 Pulse interval was 5 seconds
Server received heartbeat pulse 6 Pulse interval was 5 seconds
Server received heartbeat pulse 7 Pulse interval was 5 seconds
Server received heartbeat pulse 8 Pulse interval was 5 seconds
Server received heartbeat pulse 9 Pulse interval was 5 seconds
Server received heartbeat pulse 10 Pulse interval was 5 seconds
Server received heartbeat pulse 11 Pulse interval was 5 seconds
Server received heartbeat pulse 12 Pulse interval was 5 seconds
Server received heartbeat pulse 13 Pulse interval was 5 seconds
Server received heartbeat pulse 14 Pulse interval was 5 seconds
Server received heartbeat pulse 15 Pulse interval was 5 seconds
No pulse after 10 seconds. Server quits
Server stops.
```