THE UNIVERSITY OF SYDNEY

COMP5416 Assignment 1

Due: Week 6 (02/September/2015)

You have to study an Internet ping server written in the Python programming language, and implement a corresponding client. The functionality provided by these programs is similar to 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 will email your submission to liwei@it.usyd.edu.au. Your submission should include the complete client code and screenshot verifying that your ping program works as required by submitting a zipped archive assignment1.zip of a folder named with your unikey and containing simply these four files under name client.py, screenshot-10ping.jpg, screenshot-30ping.jpg and screenshot-50ping.jpg.

Task 1 Server code

You are given the complete code for the Ping server below. Your task is to write the Ping client. The following code fully implements a ping server. You need to run this code before running your client program. You do not need to modify this code. In this server code, 30% of the client's packets are simulated to be lost. You should study this code carefully, as it will help you write your ping client.

```
# UDPPingerServer.py
# We will need the following module to generate randomized lost packets
import random
from socket import *

# Create a UDP socket
# Notice the use of SOCK_DGRAM for UDP packets
serverSocket = socket(AF_INET, SOCK_DGRAM)
# Assign IP address and port number to socket
serverSocket.bind(('', 12000))

while True:
    # Generate random number in the range of 0 to 10
    rand = random.randint(0, 10)
    # Receive the client packet along with the address it is coming from
    message, address = serverSocket.recvfrom(2048)
    # Capitalize the message from the client
    message = message.upper()
    # If rand is less is than 4, we consider the packet lost and do not respond
    if rand < 4:
        continue
```

```
22          # Otherwise, the server responds
23          serverSocket.sendto(message, address)
```

The server sits in an infinite loop listening for incoming UDP packets. When a packet comes in and if a randomized integer is greater than or equal to 4, the server simply capitalizes the encapsulated data and sends it back to the client.

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 networks, the server in this lab injects artificial loss to simulate the effects of network packet loss. The server creates a variable randomized integer which determines whether a particular incoming packet is lost or not.

Task 2 Client code

You need to implement the following client program. The client should send a specific number pings to the server and the number is determined as one of the input parameters to run the client program. 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.

Specifically, your client program should

1. Send the ping message using UDP (Note: Unlike TCP, you do not need to establish a connection first, since UDP is a connectionless protocol.)
2. Print the response message from server, if any
3. Calculate and print the round trip time (RTT), in seconds, of each packet, if server responses
4. Otherwise, print "Request timed out"

During development, you should run the UDPPingerServer.py on your machine, and test your client by sending packets to localhost (or, 127.0.0.1). After you have fully debugged your code, you should see how your application communicates across the network with the ping server and ping client running on different machines.

Tips:

1. Your program needs to use three input parameters for its running, namely, server address, port number and ping times.
2. Use the settimeout function in your client program
3. Use the try: and except clause to catch exception when the sent packet is lost and take an appropriate action in response in your client program

Task 3 Statistics

The program calculates the round-trip time for each packet and prints it out individually. Run your ping program with different ping times and use 10, 30 and 50 as the ping times at different run. You will need to report the minimum, maximum, and average RTTs at the end of all pings from the client. In addition, calculate the packet loss rate (in percentage). Put these numbers into a report with screenshot for each run.

The University of Sydney

School of
Information Technologies

Unit of Study:

COMP5416 Advanced Network Technologies

Assignment 1

Individual Assessment

Assignment name: Assignment 1

Lecturer: Wei Li

Declaration

I declare that I have read and understood the *University of Sydney Student Plagiarism: Coursework Policy and Procedure,* and except where specifically acknowledged, the work contained in this assignment/project is my own work, and has not been copied from other sources or been previously submitted for award or assessment.

I understand that understand that failure to comply with the *Student Plagiarism: Coursework Policy and Procedure* can lead to severe penalties as outlined under Chapter 8 of the *University of Sydney By-Law 1999* (as amended). These penalties may be imposed in cases where any significant portion of my submitted work has been copied without proper acknowledgement from other sources, including published works, the internet, existing programs, the work of other students, or work previously submitted for other awards or assessments.

I realise that I may be asked to identify those portions of the work contributed by me and required to demonstrate my knowledge of the relevant material by answering oral questions or by undertaking supplementary work, either written or in the laboratory, in order to arrive at the final assessment mark.

Student ID: _____

Student Name: _____

Signed: _____

Date: _____