

ASSIGNMENT – 2

SOCKET PROGRAMMING

School of Computing, National University of Singapore

Learning Objectives

This assignment/project provides you a chance to learn about popular network, transport and application layer protocols on the Internet (User data Gram Protocol-UDP, Transmission Control Protocol-TCP, Hyper-Text Transmission Protocol-HTTP, Internet Control Message Protocol-ICMP) and to develop services using socket programming.

A) Network Tools [20 marks]

1. (5 marks) DNS queries
 - i. (3 points) Run “**dig** www.anuflora.com” from any linux based system to find the **IP** address of the server www.anuflora.com. [Give the **screen-shot** showing the output]. State the IP address of the server (2 marks). Is the server hosted inside NUS? (1 mark)
 - ii. (2 points) Use dig to find the **IP** addresses of **mail servers** of the domain comp.nus.edu.sg.
2. (5 marks) Use **telnet** or **openssl** to connect to our SMTP server (smtp.gmail.com or smtp.comp.nus.edu.sg or smtp.nus.edu.sg) other SMTP servers where you have an account. Follow the SMTP commands as given in the Luminus->Multimedia->SMTP-demo to send an email to your yahoo/gmail/hotmail account. Give the **screen-shot** showing the list of commands (including telnet/openssl connection line) you have used to send the mail successfully.

Tip1: NUS SMTP and POP servers require TLS/SSL. Hence, instead of Telnet, you can use "openssl". Alternatively, you can connect to other SMTP and/or POP servers.

```
$openssl s_client -connect smtp.gmail.com:465 -crlf -ign_eof
```

Tip2: Command for base64 encoding:

```
$echo -n "your text to encode" | base64  
or use online tools like: https://www.base64encode.org/
```

3. (5 marks) Use **telnet** or **openssl** to connect to our POP server (pop.nus.edu.sg) or other POP servers and access your mailbox by sending raw POP commands/messages. Retrieve the latest mail from your mailbox [Give **screen shot** showing your communication with POP server]. --- (Refer Appendix A for raw POP commands).

4. (5 marks) Use **telnet ftp.ietf.org 21** (or any other FTP server; you can use **openssl** instead of telnet for secured connection) to connect to our ftp server and retrieve the list of files and folder inside root or any other folder. How many TCP connections are required to retrieve a file (directory listing text file or any other file)? [Give **screen shot** showing your communication with FTP server. Include both control connection and data connection].

B) Socket programming (TCP)– Level 1 [10 marks]

1. (10 marks) Develop a TCP client program using the basic c/c++ socket library to construct a http request and retrieve the file “yourip.php” from the url:
<http://nwtools.atwebpages.com/yourip.php> or https version at
<http://varlabs.comp.nus.edu.sg/tools/yourip.php> [need to handle encryption]

Parse the contents and display your current public IP address from the file returned by the server “varlabs”.

The output should be in the following format:

My public IP address is a.b.c.d

[Note: You should do the assignment by creating the proper HTTP messages as per the RFCs and using basic Socket library.]

C) Socket programming (RAW) – Level 2 [40 marks]

1. (25 marks) Write a c/c++ program using the basic c/c++ socket library to implement traceroute tool using TCP. The program/tool should take destination URL or IP address as input. The tool should send TCP SYN segments to the destination successively by increasing IP TTL by 1 each time. Select a destination port/service that is most possibly not running in the destination host. This will result in “host unreachable” or “port unreachable” ICMP error message. Your program should have a RAW socket to retrieve this ICMP error message. Your program should output the sender’s IP address of the ICMP error message.
 - i. [25 marks] Prints list set of IP addresses from successive ICMP error messages.
 - ii. [5 marks] Terminates after receiving and printing the IP address from ICMP “port unreachable” or “TCP RST” messages. If there is no ICMP or TCP RST, the program should terminate after “Time Out” without printing the destination IP.
 - iii. [10 marks] Checks whether the received ICMP error messages are related to the TCP SYN packets (check only whether the reply is for TCP protocol, no need to check port number).

Note:

- Firewalls at your laptop or router may block ICMP error messages. You may need to turn-off firewall to receive ICMP error messages.
- WINDOWS users can install VM ("VMWARE workstation Player" or Virtualbox) and then install Ubuntu/Linux on top of the VM to get Linux environment for the assignment.
- If you are using a VM to run your code, you have to turn on 'bridged connection' otherwise the ICMP packets won't be passed into the VM.
- To check – You can use tools like Wireshark (www.wireshark.org) to check whether your laptop/VM can receive ICMP error message.
- TCP SYN packets may be dropped by firewall when it is sent to unknown port number. Check RFC 4953 (Defending TCP Against Spoofing Attacks) for other defending mechanisms used by hosts and firewalls.
- Some info on TCP sequence numbers:
TCP segments include a sequence number, which enables out-of-order receiver processing as well as duplicate detection. The sequence number space is also used to manage congestion, and indicates the index of the next byte to be transmitted or received. For RSTs, this is relevant because legitimate RSTs use the next sequence number in the transmitter window, and the receiver checks that incoming RSTs have a sequence number in the expected receive window. Such processing is intended to eliminate duplicate segments (somewhat moot for RSTs, though), and to drop RSTs that were part of previous connections.

D) Smart Mailer Program– Level 3 [30 marks]

The smart mailer program sends mails to a list of emails provided in a *csv* file using any SMTP services **of your choice** (eg. smtp.gmail.com or smtp.comp.nus.edu.sg or smtp.nus.edu.sg). The mailer should introduce some delay between emails to reduce spam mails. You mailer should establish an SMTP connection and exchange SMTP messages to send mail. You can use any programming language to implement the smart mailer (c, c++, java, python and use any high-level API (such as *openssl API*) to establish connection with the SMTP server.

Requirements

- [5 marks] The mailer should accept an input file (maildata.csv) with list of email ids, names, department codes
- [8 marks] The mailer should accept a department code and send mails only to those departments that are in the list. If the department code is "all" then the mails should be send to all email ids in the list.
- [10 marks] The mailer should accept subject and body information from a text file. The body should have placeholders which are marked with special characters (such as, #name#, #department#). The placeholders should be replaced by actual names and department names before sending the mail.
- [5 marks] The program print a report showing number of emails sent grouped by department code.
- UI can be GUI or CUI.
- [2 marks] Your codes should be well written and well commented.

Submission:

- All submissions should be done in respective folders of LumiNUS before due date. Late submission penalty: **10% per day**.

Submission Date 1: (For Section A and B) - **Wednesday 16-Sep-2020**

Submit **one RAR/ZIP file** (change filename to your *Student Number*) containing **one** word-docx (or PDF) for Part-A and **one** (.c or .cpp) file for Part-B to 'LumiNUS->Assignment2->PartAB-Submission'. Add comments in the beginning of the .c or .cpp file with instructions for compiling/running your source code in Part-B.

Submission Date 2: For Section C. - **Monday 28-Sep-2020**

Submit **one RAR/ZIP file** (change filename to your *Student Number*) containing **one** word-docx (or PDF) and **one** (.c or .cpp) file to 'LumiNUS->Assignment2-PartC-Submission'.

Submission Date 3: For Section D. - **Friday 9-Oct-2020**

Submit **one RAR file** (change filename to your *Student Number*) containing **one** (.c or .cpp) file and **one** word-docx (or PDF) with instructions for compiling/running your source code to 'LumiNUS ->Assignment2-PartD-Submission'.

If you have any question/clarification, discuss through **discord.com** under the channel 'Assignment'.

Bhojan Anand /NUS

'Students who approach education from a **deep-learning perspective** make significant improvements, remember what they learn, and feel empowered to make a difference. Those who take a surface approach to learning may get good grades, but they rarely benefit much in the long term'. – *from several research findings*.

Learning in depth is the key focus of CS3103. Try more beyond the questions above.

[We have a Zero Tolerance for Plagiarism Policy. If you are here only for marks/grades, just let me the grade/mark you prefer to have!]

More on deep learning approach ...

Students Who Take a Deep Approach--

- Attempt to understand material for themselves*
- Seek rigorous and critical interaction with knowledge content*
- Relate ideas to previous knowledge and experience*
- Discover and use organizing principles to integrate ideas*
- Relate evidence to conclusions*
- Examine the logic of arguments*