



Department of Computer Science  
Computer Networks  
Due: Sunday 5th September (23.59)

**Your name:**

TA Name:

Time Taken:

Estimated Time: 20 hours

This is an individual assignment and should be submitted as a pdf, with accompanying code, using Canvas.

For those who like to dabble in the dark arts, the latex version is also available. You may submit in any legible form you wish, but please use tar to bundle files.

Practical programming exercises may be done on your local laptop, or using your account on skel.ru.is. Part of this assignment is getting your programming environment setup for the rest of this course. We strongly recommend that you create a suitable environment on your laptop or other machine which you can use to run client and monitoring software such as tcpdump. If you have any issues at all in getting setup, please contact us **immediately**.

Marks are awarded for question difficulty. While there is typically a relationship between difficulty and length of answer, it may not be a strong one. Always justify your answer if necessary, especially with somewhat open ended design questions.

All submissions must be bundled up using the *tar* command. If you submit an assignment using anything else (zip say), and persuade Canvas to accept the upload by renaming the file, you will receive an automatic 0.

Question:	1	2	3	4	Total
Points:	5	18	17	10	50
Score:					

Useful Network Commands (Linux)	
man	Online manual for command
man -k	keyword search on manual
whois	Domain registry information
htop	Enhanced top, show processes
iftop	Show top traffic/network
iptraf-ng	IP traffic monitor
route	show local routing
arp -v	Show address resolution cache(root)
ncat < ip > < port >	Connect to remote system
nmap	Scan remote host
nmap -A -T4 scanme.nmap.org	scan host ports - os, version and traceroute
netstat	Show network statistics
netstat -antup	Process path info
netstat -t	Show only tcp connections
netstat -u	show only udp connections
ss	Show detailed network info
ss -s	Summary of network info
ifconfig	Network Interface
ip	Enhanced ipconfig/network info.
dig	DNS record lookup
dig < domain >	Get full record for < domain >
nslookup	interactively query DNS
set type=A	Specify A records
server=	Specify server to query

## Introduction

The commands above are a summary list of command line tools that can be used for networking purposes. In particular, nmap and ncat (note, another version of ncat called nc exists, but is not always as reliable), will be useful for this assignment. Some of these commands may need to be installed using the package management for your machine, and should be available via the bash console on Windows.

## Network Connectivity

The goal of this exercise is to first introduce you to some useful network command line tools, to help you explore and debug network programs, and then get you to write your first very simple network program.

- 1 ..... 5 points  
(5 points) Download the server.cpp file attached to this assignment. This is a very simple TCP/IP server, which listens for incoming TCP connections on port 5000. Compile and run it using the commands:

```
Linux and Windows/Bash :
    g++ -Wall -std=c++11 server.cpp -o server
OSX
```

```
g++ -std=c++11 server.cpp -o server  
  
./server
```

Note, you may need to install g++ (On OSX use brew).

Now connect to it from the same machine you ran the server on using the standard loopback network interface "127.0.0.1" by performing the following steps.

From a separate terminal, verify that you can connect to the server which is listening on port 5000, using the ncat command:

```
ncat 127.0.0.1 5000
```

The server should tell you a client has connected. Now on a third terminal, run the tcpdump command (which may require sudo privilege) to monitor the traffic between the client and the server, using the following incantation:

```
tcpdump -X -i lo host 127.0.0.1 and port 5000
```

The server supports one command, `SYS < command >` which will run the one word command specified on the server. Enter a command of your choice (eg. who, ls, w, etc), and observe the results on your terminal.

- (a) (5 points) For full marks on this question, repeat the above series of steps, connecting to the server from a **different** machine than the one the server is running from, and submit a screen capture of each terminal, ncat, ./server and tcpdump (three in total).

If for **any** reason you are not able to connect from a different machine, submit the screen captures from the same machine as above, and a convincing explanation, including the name of the TA you spoke to to get help, of the issue you ran into.

2..... *18 points*  
(18 points) For the second part of this assignment, you need to write a client program that connects to the server remotely, in the same way you used ncat to do. The client needs to

- connect to the server,
- send a SYS command to the server
- receive the response from the server and print it out.

For full marks for this question submit:

- (a) (13 points) Client code connecting as described above
- (b) (2 points) Screen shot of server receiving command and executing it
- (c) (3 points) tcpdump of command being sent to server and response sent to client

3..... *17 points*  
(17 points) For the third part of this assignment, you need to modify both the server, and the client you have just written.

The server should be modified to send the client the output from the command executed as a response (instead of the status response) and also to handle parameters on the command being sent from the client. For example, "ls -sal". The client should be modified to read the command it should sent from the command line and be ready to accept and send another command after the response was received. That is, the client operates in a loop, send command, receive results, print, send command, etc., as network clients often do.

- (a) (7 points) Modified server code meeting above specification
- (b) (5 points) Modified client code meeting above specification
- (c) (5 points) tcpdump of at least two commands being sent to server and response being received.

4..... *10 points*  
(10 points) In addition to the marks for the individual questions, the following marks will be awarded:

- (a) (2 points) A Makefile is included which compiles the code submitted
- (b) (2 points) README file is provided explaining how to compile and run submitted code. File should include command line commands used to compile (IDE's will not be accepted), and instructions on how to run the programs.
- (c) (2 points) Code compiles using the Makefile, and runs following instructions in README file
- (d) (4 points) Code is clearly structured (no 200 line functions), and well commented. Variable names are informative, and each function/class has a header describing its purpose.