



**Department of Computer Science**  
**Computer Networks**  
**Due: Sunday 30th August (23.59)**

**Your name: Eyór Óli Borgórsson**  
TA Name: Jón Pétur  
Time Taken: 23-30 august  
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:	10	15	15	6	46
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
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.
nmap -A -T4 scanme.nmap.org		scan host ports - os, version and traceroute	
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 ..... 10 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 127.0.0.10 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) (10 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.

## Solution

Below is a screenshot of:

- The client (ncat skel.ru.is 4098)
- The server running on skel.ru.is and port 4098 (./server 4098)
- tcpdump (sudo tcpdump -X -i en0 host skel.ru.is and port 4098)

```

eythoroli — ncat skel.ru.is 4098 — 86x24
Last login: Sat Aug 29 18:41:33 on tty000
Eythor.borghthorsson@Eythoroli-MacBook-Pro ~ % ncat skel.ru.is 4098
SYS ls
SYS who
SYS hello server
SYS pwd

```

```

eythoroli — ./server 4098 — /server — ssh eythorb19@skel.ru.is — 80x24
steinarb20 pts/25      Aug 29 14:56 (89-160-203-78.du.xdsl.is)
eythorb19 pts/27      Aug 29 18:42 (89-160-193-160.du.xdsl.is)
elinhb15 pts/28      Aug 29 17:41 (mobile-194-144-176-43.3g.internet.is)
arnih14 pts/29      Aug 29 17:47 (nova-157-097-014-121.cpe.nova.is)
benediktt13 pts/31    Aug 26 10:18 (89-160-199-59.du.xdsl.is)
gudbjorn20 pts/34    Aug 29 15:04 (89-160-193-64.du.xdsl.is)
runarfl18 pts/36    Aug 29 13:26 (nova-157-097-014-099.cpe.nova.is)
sandrao16 pts/37    Aug 29 18:15 (mobile-194-144-159-37.3g.internet.is)
tinna17 pts/39      Aug 29 14:41 (89-160-223-214.du.xdsl.is)
gislig18 pts/40    Aug 29 12:52 (85-220-67-243.dsl.dynamic.simnet.is)
alexandra120 pts/41  Aug 29 16:39 (89-160-198-251.du.xdsl.is)
sigurbjorg15 pts/44  Aug 29 15:25 (89-160-175-203.du.xdsl.is)
stefano10 pts/46    Aug 29 17:27 (31-209-205-173.dsl.dynamic.simnet.is)
stefanh18 pts/49    Aug 29 16:17 (nova-078-040-249-241.corp.f
marcin15 pts/50    Aug 29 15:59 (89-160-137-89.du.xdsl.is)
ragnarr18 pts/52    Aug 29 15:47 (10.3.27.141)
ingibjorgg19 pts/55  Aug 29 15:36 (122-146-17-89.fiber.hringdu.is)
SYS hello server

sh: hello: command not found
SYS pwd

/home/hir.is/eythorb19/TSAM/P1-TSAM

```

```

eythoroli — tcpdump -s 171x27
18:45:14.934522 IP 130.208.243.61.drmsfsd > 192.168.1.36.62595: Flags [I], ack 15, win 227, options [nop,nop,TS val 3088972923 ecr 670785853], length 0
 0x0000: 4500 0034 a712 4000 3606 65d7 82d0 f33d E..@.6.e...=
 0x0010: c0a8 0124 1002 f483 9c1c ae9d 1e5f cb59 ...$......Y
 0x0020: 8010 00e3 c734 0000 0101 080a b81d fc7b ...4.....{
 0x0030: 27fb 6130                                     +,a=
18:45:27.809322 IP 192.168.1.36.62595 > 130.208.243.61.drmsfsd: Flags [P.], seq 15:32, ack 1, win 2052, options [nop,nop,TS val 670798706 ecr 3088972923], length 17
 0x0000: 4500 0045 0000 4000 4006 02d9 c0a8 0124 E..E..@.e....$
 0x0010: 82d0 f33d f483 1002 1e5f cb59 9c1c ae9d ...=.....Y....
 0x0020: 8010 0804 4e0b 0000 0101 080a 27fb 9372 ...N.....'..r
 0x0030: b81d fc7b 5359 5320 6065 6c6c 6f20 7365 ...{SYS.hello.se
 0x0040: 7276 6572 0a                                rver:
18:45:27.813112 IP 130.208.243.61.drmsfsd > 192.168.1.36.62595: Flags [I], ack 32, win 227, options [nop,nop,TS val 3088985803 ecr 670798706], length 0
 0x0000: 4500 0034 a713 4000 3606 65d6 82d0 f33d E..@.6.e...=
 0x0010: c0a8 0124 1002 f483 9c1c ae9d 1e5f cb6a ...$......_j
 0x0020: 8010 00e3 629e 0000 0101 080a b81e 2ecb ...b.....
 0x0030: 27fb 9372                                     +,r
18:45:40.138943 IP 192.168.1.36.62595 > 130.208.243.61.drmsfsd: Flags [P.], seq 32:40, ack 1, win 2052, options [nop,nop,TS val 670810998 ecr 3088985803], length 8
 0x0000: 4500 003c 0000 4000 4006 02e2 c0a8 0124 E..<..@.e....$
 0x0010: 82d0 f33d f483 1002 1e5f cb6a 9c1c ae9d ...=....._j....
 0x0020: 8010 0804 b06d 0000 0101 080a 27fb c376 ...m.....'..v
 0x0030: b81e 2ecb 5359 5320 7077 640a                ...SYS.pwd.
18:45:40.142939 IP 130.208.243.61.drmsfsd > 192.168.1.36.62595: Flags [I], ack 40, win 227, options [nop,nop,TS val 3088998133 ecr 670810998], length 0
 0x0000: 4500 0034 a714 4000 3606 65d5 82d0 f33d E..@.6.e...=
 0x0010: c0a8 0124 1002 f483 9c1c ae9d 1e5f cb72 ...$......_f
 0x0020: 8010 00e3 0268 0000 0101 080a b81e 5ef5 ...h.....^..
 0x0030: 27fb c376                                     +,v

```

2 ..... *15 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. All the client needs to do is to connect to the server, and send a SYS command to the server. For full marks for this question submit:

- (a) (10 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



## 3..... 15 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 and also to handle parameters on the command being sent from the client. For example, "ls -sal". The client should be modified to receive the output of the command from the server, and to print it out on the command line, and then be ready to accept and send another command. That is, the client operates in a loop, send command, receive results, print, send command, etc., as network clients often do.

- (a) (5 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.

## Solution

Below is a screenshot of:

- The client, connecting and sending requests to the server, *SYS ls*, *SYS hello*, *SYS echo hello*, *hello*. (`./client_mod 127.0.0.1 5000`)
- The server running on 127.0.0.1 and port 5000 (`./server_mod 5000`)
- tcpdump (`sudo tcpdump -X -i lo0 host 127.0.0.1 and port 5000`)

The screenshot shows a terminal window with three panes. The top-left pane shows the source code for `server_mod.cpp` and `client_mod.cpp`. The top-right pane shows the output of the server program, which is connected to the client and receives requests like `SYS ls`, `SYS who`, `SYS hello`, and `SYS echo hello`. The bottom-left pane shows the output of the client program, which connects to the server and sends requests like `SYS ls`, `SYS who`, `SYS hello`, and `SYS echo hello`. The bottom-right pane shows the output of a `tcpdump` command, which captures network traffic on the `lo0` interface for host `127.0.0.1` and port `5000`.

```

client_mod.cpp — Tölvasamskipti
server_mod.cpp x
P1-TSAM > server_mod.cpp > ...
6 // Author: Jacky Mallett (jacky@ru.is)
7 // Modified by: Eyþór Óli Borgþórsson (eythorbi9@skel.ru.is)
8 //
9 //
10 //
11 #include <stdio.h>
12 #include <errno.h>
13 #include <stdlib.h>
14 #include <unistd.h>
15 #include <sys/types.h>
16 #include <sys/socket.h>
17 #include <netinet/in.h>
18 #include <netinet/tcp.h>
19 #include <netdb.h>
20 #include <arpa/inet.h>
21 #include <arpa/inet.h>

P1-TSAM > client_mod.cpp > ...
5 //
6 // Compile: g++ -Wall -std=c++11 client_mod.cpp -o client_mod
7 //
8 // Command line: ./client_mod <server_ip>
9 //
10 // Author: Eyþór Óli Borgþórsson;
11 //
12 //
13 #include <netdb.h>
14 #include <stdio.h>
15 #include <stdlib.h>
16 #include <string.h>
17 #include <sys/un.h>
18 #include <netinet/in.h>
19 #include <arpa/inet.h>
20 #include <arpa/inet.h>

P1-TSAM — server_mod 5000 — 69x24
Client connected on server
SYS ls
Media
P1-TSAM.pdf
client
client.cpp
client_mod
client_mod.cpp
eythorbi9@skel.ru.is
server
server.cpp
server_mod
server_mod.cpp
SYS who
Eythor.borgthorsson console Aug 4 18:40
Eythor.borgthorsson ttys004 Aug 30 16:27
Eythor.borgthorsson ttys006 Aug 30 16:52
SYS hello
sh: hello: command not found
SYS echo hello
hello
hello
Unknown command from client:hello

P1-TSAM — tcpdump - sudo — 66x27
whatsup
Last login: Sun Aug 30 16:27:46 on ttys004
Eythor.borgthorsson@Eythoroli-MacBook-Pro ~ % sudo tcpdump -X -i lo0 host 127.0.0.1 and port 5000
Password:
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on lo0, link-type NULL (BSD loopback), capture size 2621
44 bytes
16:53:28.689490 IP localhost.65119 > localhost.complex-main: Flag s [P.], seq 3643161852:3643161869, ack 3571672577, win 6379, options [nop,nop,TS val 750132903 ecr 750063113], length 17
0x0000: 4500 0045 0000 4000 4006 0000 7f00 0001 E..E..@.
@.....
0x0010: 7f00 0001 fe5f 1388 d926 3cfc d4e3 6601 .....
.<...f.
0x0020: 8018 18e7 fe39 0000 0101 080a 2cb6 1ea7 .....9..
.....
0x0030: 2cb5 0e09 5359 5320 6563 686f 2057 6861 ...SYS.
echo.Wha
0x0040: 7473 7570 00 .....tsup.
16:53:28.689522 IP localhost.complex-main > localhost.65119: Flag s [P.], ack 17, win 6379, options [nop,nop,TS val 750132903 ecr 750132903], length 0
0x0000: 4500 0034 0000 4000 4006 0000 7f00 0001 E..4..@.
@.....

```



4 ..... *6 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 names are informative, and each function/class has a header describing its purpose.

**Solution**

Makefile and README file is included. Compile by using make all and you will get further instructions.