

# Internetworking exercise 1

Perform the simple tasks given and fill in what you see in the spaces provided. Please feel free to take extra notes or to ask questions of the instructors.

## 1. Review the IP configuration on your workstation

Try the following commands (you do not need to be 'root' here):

```
$ ifconfig          --- shows all interfaces with current settings
                    (some Unices require 'ifconfig -a')
$ cat /etc/rc.conf  --- shows values to be used at startup
                    (this file is specific to the BSD family)
```

In `/etc/rc.conf` you should see settings like

```
ifconfig_fxp0="x.x.x.x"      write down the setting: _____
defaultrouter="y.y.y.y"     write down the setting: _____
```

---

Aside: on some networks, IP addresses are assigned dynamically by a DHCP server (Dynamic Host Configuration Protocol). At startup, the machine broadcasts a request to find a DHCP server, and the server allocates an IP address which it is permitted to use for a period of time.

Normally, servers are given fixed IP addresses, but if your FreeBSD machine were configured to get its IP address via DHCP, `/etc/rc.conf` would contain

```
ifconfig_fxp0="DHCP"
```

---

## 2. Testing at layer 1

Look again at the output from "ifconfig"

Identify your ethernet interface, and write down what you see for:

```
media: _____
status: _____
```

Look at the network card on your PC where the cable enters. Are there any lights? What colour(s) do they show?

```
LED colour(s): _____
```

Now, **unplug the cable** from your ethernet port. Repeat the ifconfig command, and look again at the lights. How is it now?

```
media: _____
status: _____
LED colour(s): _____
```

After doing this, plug the cable back into your PC. Check that the link status is OK again.

Can you use link status lights to identify which hub or switch port in the classroom your PC is plugged into, without following the cable?

### 3. Testing at layer 3

3.1 Find the IP address of one of your neighbour's machines, and type the following command where x.x.x.x is their IP address:

```
# ping x.x.x.x
```

This sends "ECHO REQUEST" packets to their machine, and displays any "ECHO RESPONSE" packets which may come back. After a few seconds, terminate the command by pressing Ctrl-C

Is their machine responding? \_\_\_\_\_

What is the average round-trip time for packets? \_\_\_\_\_ milliseconds

Now, repeat the command with the IP address 147.28.0.62

Is this machine responding? \_\_\_\_\_

What is the average round-trip time for packets? \_\_\_\_\_ milliseconds

Would you say this is fast or slow? What might explain this?

3.2 Have a look at the route out to the Internet for traffic going to that machine:

```
$ traceroute -n 147.28.0.62
```

How many router "hops" does traffic take to get there? \_\_\_\_\_

Look at the round-trip time for each hop (note that by default you get three probes sent to each router). Can you deduce anything from this about the Internet topology between here and the outside world?

### 4. Testing a webserver at layer 7

```
$ telnet ws.edu.isoc.org 80
GET / HTTP/1.0
Host: ws.edu.isoc.org
<blank line>
```

Beware that the line GET / HTTP/1.0 is case-sensitive; "GET" and "HTTP" must be entered in upper-case letters. After the "Host:" line, you hit Enter again to indicate the end of the request.

What's the first line of the response? A lot of information scrolls off the screen, so repeat this exercise using 'script', which starts a subshell and captures the screen output to a file.

```
$ script webtest.txt
$ telnet ws.edu.isoc.org 80
GET / HTTP/1.0
Host: ws.edu.isoc.org
<blank line>
$ exit
$ less webtest.txt
```

Write down the first line of the response, which should start with 'HTTP/'

HTTP/\_\_\_\_\_

Type a lower-case 'q' to get out of 'less'.

Repeat this test, but this time asking for a non-existent webpage:

```
$ telnet ws.edu.isoc.org 80
GET /zzzzz HTTP/1.0
host: ws.edu.isoc.org
<blank line>
```

Write down the first line of the response you get this time:

HTTP/\_\_\_\_\_

You have been sending commands and recording responses of the HyperText Transfer Protocol, which is the layer 7 protocol used by web browsers to communicate with web servers. If you want the full details of this particular protocol (for example to understand the meanings of the result codes), it is defined in document RFC 2616, which you can find at <http://www.rfc-editor.org/>, <ftp://ftp.isi.edu/in-notes/rfc2616.txt> or many mirrored locations.

Note that the port number (80) is the standard port number assigned for HTTP. Some web servers run on different port numbers, but unless you are told otherwise, you can always expect that a webserver will be listening on port 80. If you were testing a different application, it would be listening on a different port, and speaking a different protocol than HTTP.

## 5. Extra exercises (if time available)

5.1 Use 'tcpdump' to look at traffic going in and out of your machine - you may also see traffic to and from other machines. You need to be 'root' to do this.

```
# tcpdump -i fxp0 -n -s1500 -X
```

Terminate with Ctrl-C. Now, try using tcpdump's filtering facilities so that you see only traffic coming in and out of your own machine: replace x.x.x.x with your own IP address.

```
# tcpdump -i fxp0 -n -s1500 -X host x.x.x.x
```

In another window (use Alt-F1, Alt-F2 etc), login again and send some pings. Go back to the tcpdump window and check that you can see the packets going in and out.

Add the flag "-e" if you wish to see the ethernet layer 2 information for each packet.

5.2 Try pinging your neighbour's machine with a maximum-sized ping (you need to be 'root' to do this).

```
# ping -s1472 x.x.x.x
```

How does the round-trip time for this compare with a normal-sized ping?

5.3 Use the "-c" flag to ping to send exactly 100 large-sized pings to a remote site on the Internet of your choice. Use this to estimate the percentage packet loss to that site.