Name 1: ALI BELBARAKA
Name 2:

# COM-407: TCP/IP NETWORKING

# LAB EXERCISES (TP) 0
# BASIC CONFIGURATION, IP SUITE, AND PACKET INSPECTION: PING(6), TRACEROUTE(6), NETSTAT, NSLOOKUP

September 22, 2017

**Abstract**

In this lab you will practice some networking commands that enable you to obtain information about Internet machines and about the connectivity and the paths between them. You will also learn to use a GUI-based packet capture/inspection tool called Wireshark. Optionally, in research exercises, you will use tshark (command-line version of Wireshark) for packet capture/inspection.

## 1 ORGANIZATION OF THE TP AND WIRESHARK TOOL

### 1.1 TP REPORT

Type your answers in this document. We recommend you use Adobe Reader XI to open this PDF, as other readers (such as SumatraPDF, but also older versions of Adobe!) don't support saving HTML forms. That will be your TP report (one per group). When you finish, save the report and upload it on moodle. Don't forget to write your names on the first page of the report. **The deadline for report submission is September 27 (Wednesday) 11.55 PM.**

### 1.2 WIRESHARK

You will be using Wireshark to sniff packets. Since there are a lot of packets generated by the applications running on your machine, you may want to use filters. `http://wiki.wireshark.org/DisplayFilters` Please note that there are two types of filters: *capture* and *display*. Capture filters are used to selectively capture the traffic whereas with display filters, you capture all the traffic but the traffic is displayed as per the filter rules.

## 2 THE IPV4 INTERNET

Connect to the Internet in IPv4 and disable IPv6 connectivity, if needed.

To disable IPv6:

On MacOSX, you can execute the following command from the *Terminal* app

```
# networksetup -setv6off "InterfaceName"
```

If you do not know the InterfaceName, you can use the following command

```
# networksetup -listallnetworkservices
```

On Red Hat-based Linux, following commands can be used from the *Terminal* app

```
# sudo sysctl -w net.ipv6.conf.all.disable_ipv6=1
# sudo sysctl -w net.ipv6.conf.default.disable_ipv6=1
```

On Debian-based Linux, add the following in /etc/sysctl.conf file and reboot the machine.

```
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
```

On Windows, Click Start, and then click Control Panel. Click Network and Sharing Center. In the View your active networks area, click Local Area Connection, and then click Properties. On the Networking tab, clear the Internet Protocol Version 6 (TCP/IPv6) check box, and then click OK.

After disabling the IPv6 connectivity, in order to determine the following information:

- the IP address(es) of your machine `<my_ip>`,
- the netmask `<my_netmask>`, and
- the default gateway of your machine `<my_gateway>`.

In MacOS use following commands in *Terminal* app

```
# ifconfig
# netstat -nr
```

In Linux use following commands in *Terminal* app

```
# ip addr show
# ip route show
```

or in Windows use following commands in *cmd* app

```
> ipconfig /all
```

**Q1/** List your findings here:

[A1.a] <my_ip>= 128.179.162.245
[A1.b] <my_netmask>= 255.255.255.0
[A1.c] <my_gateway>= 128.179.162.1

**Q2/** Is your IP address public or private? What does the netmask in IPv4 mean? Why a default gateway is configured?

[A2] My IP address is public, i can find it in the internet at www.whatismyipaddress.com. The netmask is a 32-bit number that masks an IP address and divides it into network address and host address. A default gateway is configured to allow machines in one network to

Now, download Wireshark and install it on your computer. Start it (as administrator) and use the menu `Capture->Interfaces` to start capturing packets on the interface that you are currently using for the Internet connectivity.

**Q3/** Do you see any packet captured with destination IP address of your default gateway if you navigate to a webpage through your browser? If yes/no, explain the reason behind your observation?

[A3] No, because we don't communicate directly to the default gateway (unless we ping it)

## 2.1 PING

PONG

The ping command uses the ICMP protocol to probe whether a host is up:

```
# ping <hostname>
```

**Q4/** Start a new capture with Wireshark and then ping `www.facebook.com`. Observe the traffic generated by the ping command. Do you see only ICMP packets?. Stop the ping program and start it again after a couple of seconds. Is there a difference from the first captured packets? Explain.

[A4] No, there is also DNS packets, since my machine need to figure out the IP address of facebook.com. In the second capture, we don't observe this DNS packets since the IP address of facebook.com has been cached.

**Q5/** In a browser open `www.swisscom.ch`. Next, try pinging it. Explain.

[A5] We fail pinging it, since they have probably configured their server to block access for ICMP packets.

## 2.2 TRACEROUTE AND NETSTAT

**traceroute** is a tool for displaying the route to a destination.

In MacOS and Linux:

```
# traceroute www.facebook.com
```

In Windows:

```
> tracert www.facebook.com
```

**Q6/** Do you see more than one name/IP address at any of the hops? If so, why? Which OS (Linux, MacOSX, or Windows) are you running?

[A6] Yes for a hop there is three different IP addresses. The reason is that next hop router has multiple paths hence multiple IP addresses with equal "cost"; trying to balance load on these different paths.

I am using MacOSX.

**netstat** is a tool for displaying TCP connections, routing table, interfaces and network statistics. Open a web browser, go to `lca.epfl.ch`, and leave the browser open for the moment.

Look at the active TCP connections.

```
# netstat -t -n
```

The `-n` switch prevents name resolving and makes netstat display results faster (but obviously without the names of the hosts).

**Q7/** Identify the TCP connections opened by visiting the `epfl.ch` webpage. Write them down and describe them here. Is there one, or are there several such connections? Why?

[A7]

| | | | | | |
|---|---|---|---|---|---|
| tcp4 | 0 | 0 | 128.179.162.245.53649 | 128.178.156.24.80 | ESTABLISHED |
| tcp4 | 0 | 0 | 128.179.162.245.53648 | 128.178.156.24.80 | ESTABLISHED |
| tcp4 | 0 | 0 | 128.179.162.245.53647 | 128.178.156.24.80 | ESTABLISHED |
| tcp4 | 0 | 0 | 128.179.162.245.53646 | 128.178.156.24.80 | ESTABLISHED |

There is several of them since to load a web page the browser loads in parallel : HTML page and graphics (images..)

# 3 NAMES IN THE INTERNET

*Juliet*: [...]
What's in a name? That which we call a rose
By any other name would smell as sweet.

W.S.

Replace your DNS servers by an inexisting IP address, say `1.2.3.4`. If you configured statically your DNS servers, don't forget to write them down somewhere before changing them to `1.2.3.4`.

Go to the `Properties` of your Internet connection. Click on Internet Protocol Version 4, `Properties`, choose `Use the following DNS server addresses`, and write `1.2.3.4`

Use the manual configuration in the network settings and set the DNS address to `1.2.3.4`

Switch to root mode using `su` and edit the `/etc/resolv.conf` file. Comment out the lines that begin with `nameserver` (precede them with the `#` character) and add one line `nameserver 1.2.3.4`

**Q8/** Try pinging Facebook and observe the traffic with Wireshark. What happens?

[A8] My machine continually sends DNS request to the DNS servers. It can't obtain the IP address of Facebook from the DNS server we put.

**Q9/** Try pinging the IP address of Facebook that you discovered in Sections 2.1 and 2.2. Does it work?

[A9] Now it works, since we don't need to ask the DNS server for Facebook's IP address.

**nslookup** is a command-line tool for querying Domain Name System (DNS) name servers. Run `nslookup` with the address of the Google public DNS server.

```
# nslookup - 8.8.8.8
```

**Q10/** In the > prompt, type `lca.epfl.ch`. Give the IPv4 and IPv6 addresses of `lca.epfl.ch`. Use `set type=A` for IPv4 or `set type=AAAA` for IPv6

[A10] IPv4 address : 128.178.156.24
IPv6 address : 2001:620:618:19c:1:80b2:9c18:1

**Q11/** Do you recognize the IPv4 address in the IPv6 address, or vice-versa?

[A11]
128 to hex is : 80
178 to hex is : b2
156 to hex is : 9c
24 to hex is : 18,  we recognize the last portion of IPv6 : 80b2:9c18

Restore now your initial DNS configuration.

Start a capture in `wireshark` and do a traceroute in IPv4 to `www.facebook.com`. Focus on the line:

```
swiel2 (192.33.209.33)  1.219 ms  0.968 ms  0.944 ms
```

**Q12/** Look at the capture and identify the packet in which you see the name `swiel2`. How does this differ from the usual DNS response observed in previous questions? Based on the observed difference, comment on how `traceroute` works.

[A12]
Traceroute perform a reverse DNS query. It sets the Time To Live for a packet to 1, sends it to the destination host and listens for reply. When it receives "time exceeded" response it examines the packet and determines where it came from, hence identifying the machine one hop away. Then we set the TTL to 2 and send the packet to destination host and determine machine at 2 hops, and so on

**Q13/** Analyze the capture and comment on how `traceroute` finds successive hops.

[A13]
Traceroute fins successive hops by varying the TTL of packets sent.

# 4 THE IPV6 INTERNET

Now let's examine the situation when only IPv6 connectivity is present.

Find access to an IPv6 network and **disable IPv4 on your machine**.

To disable IPv4:

On MacOSX, you can execute the following command from the *Terminal* app

```
# networksetup -setv4off "InterfaceName"
```

On Red Hat-based Linux, following commands can be used from the *Terminal* app

```
# sudo sysctl -w net.ipv4.conf.all.disable_ipv4=1
# sudo sysctl -w net.ipv4.conf.default.disable_ipv4=1
```

On Debian-based Linux, add the following in /etc/sysctl.conf file and reboot the machine.

```
net.ipv4.conf.all.disable_ipv4 = 1
net.ipv4.conf.default.disable_ipv4 = 1
net.ipv4.conf.lo.disable_ipv4 = 1
```

On Windows, Click Start, and then click Control Panel. Click Network and Sharing Center. In the View your active networks area, click Local Area Connection, and then click Properties. On the Networking tab, clear the Internet Protocol Version 4 (TCP/IPv4) check box, and then click OK.

If IPv6 networking is disabled (which might be the case if you used the same interface as for second section of the TP), enable it before accessing an IPv6 network.

To re-enable IPv6 for a network interface (if not already enabled):

On MacOSX, you can execute the following command from the *Terminal* app

```
# networksetup -setv6automatic "InterfaceName"
```

On Red Hat-based Linux, following commands can be used from the *Terminal* app

```
# sudo sysctl -w net.ipv6.conf.all.disable_ipv6=0
# sudo sysctl -w net.ipv6.conf.default.disable_ipv6=0
```

On Debian-based Linux, remove the lines you added in /etc/sysctl.conf file while disabling IPv6 connectivity and reboot the machine.

On Windows, Click Start, and then click Control Panel. Click Network and Sharing Center. In the View your active networks area, click Local Area Connection, and then click Properties. On the Networking tab, mark the Internet Protocol Version 6 (TCP/IPv4) check box, and then click OK.

IPv6 access is provided in or around INF019 room via a wireless access point (`SSID: lca2-tcpip-labs, password: tcpip1516`).

Use wireshark to observe the traffic. On your computer type

```
# ping6 www.facebook.com
```

**Q14/** Describe some differences in the observed traffic compared to the IPv4 case. Write the average RTT you get and compare it with the IPv4 case. Explain the differences if any.

[A14] We notice that IPv6 take a different path to reach facebook.com. The average RTT for IPv4 is 15.9 ms and average RTT for IPv6 is 16.1ms. There isn't a big difference probably because we didn't experience congestion control.

Repeat the test with the `traceroute` command from Section 2. Use:

In Linux or MacOS:

```
# traceroute6 www.facebook.com
```

In Windows:

```
> tracert -6 www.facebook.com
```

**Q15/** Write the result. Does the path to Facebook in the IPv6 Internet cross the same routers as in IPv4?

[A15]
```
Alibel:~ Belbaraka$ traceroute6 www.facebook.com
traceroute6 to star-mini.c10r.facebook.com (2a03:2880:f106:83:face:b00c::25de) from
2002:80b2:9c51::ffff:4a, 64 hops max, 12 byte packets
 1  2002:80b2:9c51::cccc:1  18.228 ms  1.376 ms  1.512 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  *
    edge-star-mini6-shv-01-amt2.facebook.com  43.667 ms  24.591 ms
```

Now, open the web browser (new window), go to `lca.epfl.ch`.

**Q16/** Do you notice a difference between two versions of `lca.epfl.ch` pages? Can you imagine by which mechanism such a difference may occur ?
*Hint: Which device (default gateway, intermediate routers, the web server, etc) do you think is in charge of displaying the web content for IPv4 if you are connected to an IPV4 network or for IPv6 otherwise?*

[A16] We can observe the IPv6 logo. It's web server who putted that logo, since it managed to identify that our request arrived over IPv6.

Look at the active connections.

```
# netstat -t -n
```

**Q17/** Compare the output that is related to `epfl.ch` with the one that you wrote down for IPv4. Comment about it

[A17] We observe the same TCP connections.

**Q18/** Try pinging `www.swisscom.ch` again. Did it work? Explain.

[A18] This time it works. The server has probably blocked ICMP packets over IPv4 and allowed them over IPv6.

# 5 IPv4 AND IPv6

Let's see what happens when both IPv4 and IPv6 Internet connectivities are present. **Stay connected in IPv6, but enable IPv4.**

From your computer do a traceroute in IPv4 and IPv6 to `www.switch.ch`

**Q19/** Does it work in both cases?. Write down any difference in the traceroutes

[A19] Yes it works in both cases and same routers are traversed.

Now, start a new `Wireshark` capture, open a browser and type `www.switch.ch`.

**Q20/** Check the capture in Wireshark, your connection to the webpage is done in IPv4 or in IPv6?

[A20] The connection is done in IPv6. (on Mac OS X)

**Q21/** Explain how do you think your machine could decide whether it uses IPv4 or IPv6.

[A21] Generally IPv6 takes precedence over IPv4, but it might the case that my machine decides to use IPv4 over IPv6 depending on network latency.

# RESEARCH EXERCISES (OPTIONAL)

## 6 NETWORK PACKET INSPECTION

We need to see or inspect the packets leaving or coming to our computer or other computers for various reasons. These reasons vary depending on the person and his motivations. For example, network administrators need this for troubleshooting network-related problems, software developers for debugging network-related code and newtork protocol implemtations, and security engineers for analyzing the network traffic for security purposes. In general, we all can use these tools to understand how machines actually communicate with each other, i.e., to understand the internals of the network protocols.

There exists many tools for network packet inspection. Under the hood, all these tools use pcap library on linux and winpcap on windows but they differ in the way users can interface with them and the featueres they provide. For example, Wireshark is a powerful sniffer which can decode lot of protocols. Wireshark filter syntax lets you capture only the packets you are interested in. It provides a nice GUI. There also exists a command line version of wireshark, called tshark. Depending on one's needs, abilities, and familiarity, one may sometimes find tshark more handy than wireshark or vice-versa.

tcpdump is another tool, that comes pre-installed with almost all unix distributions, to capture the live traffic. However, tcpdump has limited protocol decoding features, compared to tshark and wireshark. Most of the times, as a network packet sniffer and decoder, the best bet is to use either tshark or wireshark.

In this section, we introduce you with tshark. The capture and display filters, we are going to use in tshark, can also be used in wireshark.

### 6.1 TSHARK

tshark lets you capture packet data from a live network, or read packets from a previously saved capture file. The captured packets are decoded by tshark and then, can either be printed to the standard output or written to a file. TShark's native capture file format is pcap format, which is also the format used by wireshark and tcpdump.

#### 6.1.1 A SHORT TUTORIAL ON TSHARK

To capture all the traffic passing through `eth0` interface and save it in `captured_packets.pcap` file, the following command can be used:

```
# tshark -i eth0 -w captured_packets.pcap
```

where `-i` should be followed by the name of the interface and `-w` with the name of the file for captured data.

Now, using a web browser, visit few web pages like facebook.com or cnn.com. Once you're done, stop the packet capture by pressing Ctrl + C.

To read the packets captured in `captured_packets.pcap` file, use the `-r` option. Following should read all the packets captured in the `captured_packets.pcap` file:

```
# tshark -r captured_packets.pcap
```

If you want only http request packets to be displayed, please do:

```
# tshark -r captured_packets.pcap -Y http.request
```

where `-Y` option lets you specify display filters (using the same syntax as in Wireshark).

Now, let's display the hosts you connected through http. To specify that, you need to use `-T` option to specify that we want to extract fields and `-e` option to specify the field you want to be displayed. Therefore, the whole commands becomes:

```
# tshark -r capture.pcp -Y http.request -T fields -e http.host
```

If you want to check whole list of available options in tshark, you can do:

```
# tshark -help
```

or the help page can be accessed through web with this link

`https://www.wireshark.org/docs/man-pages/tshark.html`

The capture and display filters used in tshark are the same as in Wireshark and can be accessed with below links.

Capture Filters: `https://wiki.wireshark.org/CaptureFilters`

Display Filters: `https://wiki.wireshark.org/DisplayFilters`

### 6.1.2 EXERCISE 1

Every computer on Internet is assigned an address, called ip address. To send a packet to a remote machine, you need its ip address. While you use your computer, you want to know how your computer finds the ip addresses of other machines. For example, when you type `facebook.com` in your browser, before you connect to the facebook server, your computer needs to find the ip address of the facebook and therefore, your machine contacts the local DNS server, whose ip address your machine knows. The DNS server responds to your request and sends the ip address of the requested machine in its response.

In this exercise, your goal is to use tshark and capture only the packets that are related to the interaction of your machine with the DNS server. You should display the DNS server(s) your machine talks to, the request it makes, and finally, the response it receives from the DNS server. Once you come up with the tshark command, run the command in the terminal, and then, visit some web pages you have never visited in the recent past. In the terminal, your tshark command should display the dns server contacted, the server for which the DNS query is made, and the response from the DNS server, i.e., the ip address of the requested server, as you browse the web.

Hint: Caputre only DNS response packets from the DNS server as it will contain both the request query and the response.

**Q22/** Please write below all the tshark commands (and their corresponding results) you tried in the order you typed in, even if you did not succeed. We are interested in your thought process, how did you proceed,

the commands you tried, and how close you finally could come to the solution even if you did not succeed.

[A22]

### 6.1.3 EXERCISE 2

Alice is soon going to have her holidays. She is searching for holiday offers on the web. She finds a very interesting and cheap offer at a website and therefore, she hurries up to book it. She enters all her details in a html form, including her name, date of birth, phone numbers, email addresses, home address, and registers for this offer. After registration, when she wants to pay for this offer, she realizes that her connection to this website (until now) is not encrypted. So she stops the online payment.

The pcap file, named alice.pcap, stores all the above-mentioned activities of Alice, captured by tshark at her network interface. Now, your job is find the packet in the pcap file that contains all her information. You should use tshark command to get hold of all her details she typed in for reserving this trip.

**You are not allowed to use Wireshark.**

Hint: The details are filled by Alice in a html form. Therefore, an http post request body should contain her details.

**Q23/** Please write below all the tshark commands you tried in the order you typed in, even if you did not succeed to get her details. We are interested in your thought process, how did you proceed, the commands you tried, and how close you finally could come to the solution even if you did not succeed.

[A23] 

### 6.1.4 EXERCISE 3

Alice uses telnet to log into her remote machine. As you might probably already know that telnet traffic is not encrypted and therefore, Alice's telnet communication can be read at any intermediate machines. While Alice connects to her remote machine, her internet connection interface packets are captured using tshark in a file called telnet.pcap.

Now you job is to find the password of Alice. Her login to the remote machine is `testuser`. So you need to find the password for the user `testuser`.

Hint: The captured telnet session is in raw (per-character) mode. You will see each character of password in a seperate line.

**Q24/** Please write below all the tshark commands in the order you tried, even if you did not succeed to get her password. We are interested in your thought process, how did you proceed, the commands you tried, and how close you finally could come to the solution even if you did not succeed.

[A24]