Name 1:
Name 2:

COM-407: TCP/IP NETWORKING

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

With Solutions

September 18, 2019
Deadline: September 25, 2019 at 23.55 PM

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 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.

2 THE IPV4 INTERNET AND NETWORK PACKET INSPECTION

2.1 IFCONFIG

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, we now want 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 *powershell* app

```
> ipconfig /all
```

Q1/ List your findings here:

- IP address: 128.178.151.219
- Network Mask: 255.255.255.0
- Default Gateway: 128.178.151.1

Q2/ Is your IP address public or private? What does the netmask in IPv4 mean?

Solution. *In this case, the IP address is public, which can be confirmed by navigating to the link* `http://www.myipaddress.com` *and confirming that the IP address given in the web page is the same as the one given to the Ethernet adapter. OR We have a range of private ip addresses and as we can see that this ip address does not belong to this range, it is public. The netmask or prefix is used to distinguish the "network" and the "host" parts of an IP address.*

2.2 NETWORK PACKET INSPECTION. WIRESHARK

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 network protocol implementations, 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 packet capture libraries such as libpcap, winpcap or npcap but they differ in the way users can interface with them and the features they provide. For example, Wireshark is a powerful sniffer which can decode lot of protocols. It provides a nice GUI for make usage more user friendly.

Since there are a lot of packets generated by the applications running on your machine, you may want to use filters, for more details see
`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.

Now, download Wireshark and install it on your computer. (Note that, if you are using Windows, it will require to install npcap library. While installation process, you do not need to install the npcap loopback adapter). 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/ Write a command that filters only the packets with destination IP address of your default gateway. Do you see any packet captured if you navigate to a webpage through your browser? If yes/no, explain the reason behind your observation?

Solution. *The filter command is:* `ip.dst == your_default_gateway`
No, unless you are pinging your default gateway or communicating directly with it by any mean (DNS, FTP, HTTP, SCP, etc). In IP, communication is done end-to-end thus in general we should not see IP packets with destination IP address of any of the intermediate devices, including the default gateway.

2.3 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`. Which exchanges of messages is happening after first ping command according to the theory? Now find these messages in the Wireshark output. 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?

```
128.178.151.139 128.178.15.227 DNS 74 Standard query 0x59d0  A www.google.com
28.178.15.227 128.178.151.139 DNS 154 Standard query response 0x59d0  A 173.194.35.17
      A 173.194.35.18 A 173.194.35.19 A 173.194.35.20 A 173.194.35.16
128.178.151.139 173.194.35.18 ICMP 74 Echo (ping) request  id=0x0001, seq=3/768, ttl=128
173.194.35.18 128.178.151.139 ICMP 74 Echo (ping) reply    id=0x0001, seq=3/768, ttl=52
```

Solution. *First a DNS query is performed, next a ping request is sent to the IP address of facebook.*

The second time the DNS request is typically not performed. The IP address was cached.

Other valid observations: the ARP request for the gateway is not performed either (ARP cache), the sequence numbers continue from where they left off during the first ping, another IP address is used (due to Facebook's load balancing system), etc.

Q5/ In a browser open `www.swisscom.ch`. Next, try pinging it. Does it work? Explain the result.

Solution. *The server hosting the website is up, yet it is configured not to respond to ping (ICMP is disabled).*

Q6/ Ping `www.canterbury.ac.nz` and `www.newzealand.com`. What are Round-trip times (RTTs) for each ping? Based on your observation, can you identify which server can be located in New Zealand?

Solution. *Ping* `www.canterbury.ac.nz` *is around* 350 *ms, whereas for* `www.newzealand.com` *it is around* 20 *ms. Since the minimum possible RTT for the packet to go around the globe is* 200 *ms, the* `www.newzealand.com` *is for sure not in New Zealand.* `www.canterbury.ac.nz` *may be in New Zealand, however we cannot be sure based only on RTT.*

2.4 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
```

Q7/ Start Wireshark and do `traceroute` to `www.facebook.com`. How did you filter the packets that are coming from your machine during `traceroute`? Which OS (Linux, MacOSX, or Windows) are you running? Does your system uses ICMP, TCP or UDP protocol for `traceroute`? Write down the result of the traceroute.

```
traceroute to star-mini.c10r.facebook.com (31.13.64.35), 64 hops max, 52 byte packets
 1  cv-gigado-v436 (128.179.136.1)  1.815 ms  1.138 ms  2.439 ms
 2  c6-ext-v200 (128.178.200.1)  2.064 ms  1.647 ms  1.163 ms
 3  swiel2 (192.33.209.33)  1.625 ms  1.626 ms  1.573 ms
 4  swice1-100ge-0-1-0-3.switch.ch (130.59.38.193)  2.586 ms  2.600 ms  2.527 ms
 5  80.249.212.175 (80.249.212.175)  18.695 ms  18.558 ms  20.104 ms
 6  po111.asw01.ams3.tfbnw.net (31.13.30.48)  18.386 ms
    po111.asw02.ams3.tfbnw.net (31.13.31.14)  18.275 ms
    po111.asw01.ams3.tfbnw.net (31.13.30.48)  18.787 ms
 7  po226.psw01.ams2.tfbnw.net (129.134.40.133)  17.656 ms
    po236.psw01.ams2.tfbnw.net (129.134.33.201)  17.991 ms
    po241.psw04.ams2.tfbnw.net (157.240.35.185)  18.063 ms
 8  173.252.67.59 (173.252.67.59)  18.011 ms
    173.252.67.175 (173.252.67.175)  18.688 ms
    173.252.67.69 (173.252.67.69)  18.065 ms
 9  edge-star-mini-shv-01-amt2.facebook.com (31.13.64.35)  18.402 ms  17.989 ms  17.993 ms
```

Solution. *We need to filter the packets with destination address of the facebook server:* `ip.dst == facebook_ip_address`*. If you have Mac or Linux, you will see UDP packets.* `tracert` *in Windows uses ICMP, the protocol used to transport test and error messages of the IP layer.*

netstat is a tool for displaying TCP connections, routing table, interfaces and network statistics. On Linux, netstat (part of net-tools) is superseded by **ss** (part of iproute2).

Open a web browser, go to `www.epfl.ch`, and leave the browser open for the moment.

Look at the active TCP connections.

In MacOS and Windows:

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

In Linux:

```
# ss -t -n
```

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

Q8/ Identify the TCP connections where destination IP address is IP address of `www.epfl.ch` webpage. Is there one, or are there several such connections?

Solution. *Several connections are established, on lab machine there are 4 connections.*

2.5 MAC ADDRESSES

A MAC address (media access control address) of a device is a unique identifier assigned to a network interface controller (NIC). MAC addresses are linked to the hardware of network adapters. A MAC address is given to a network adapter when it is manufactured. It is hardwired or hard-coded onto your computer's network interface card (NIC) and is unique to it.

Q9/ What is the MAC address of your wireless interface? How can you find a MAC address of your default gateway?

Solution. *On my machine (MAC OS) the MAC of wireless interface `en0` is `f4:5c:89:99:e8:4b`.*
In order to find the default gateway, run `netstat -nr` (`ipconfig /all` on Windows and `ip route show` on Linux) to see the IP address of your gateway. Then ping that IP address. Finally, run `arp -a` and find the gateway IP address. Next to it you'll see the MAC address.

If you want to do it with Wireshark, ping the gateway and capture the ICMP packets. Look at the ethernet layer for the echo request destination MAC. MAC of my default gateway is `0:8:e3:ff:fc:4`.

Q10/ Are your and your lab partner's machines in the same subnet? Can you find a MAC address of your lab partner's machine from your machine? How?

Solution. *If the machines are in the same subnet then you can ping the machine of your partner and then find his MAC address using `arp -a`. Otherwise, you will not be able to get MAC address.*

Q11/ Ping `www.facebook.com`. What is the MAC address of the packet received from facebook while pinging? Is this the MAC address of the facebook server?

Solution. *The MAC address of received packet is the MAC of your default gateway. It is impossible to know the MAC address of facebook server, since you are not in the same LAN.*

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`

Q12/ Try pinging Facebook and observe the traffic with Wireshark. What happens?
Solution. *A DNS request is sent to the bogus server `1.2.3.4` with no reply back*

Q13/ Try pinging the IP address of Facebook that you discovered in Sections 2.3 and 2.4. Does it work?
Solution. *Since there is no need to resolve a name, the ping to Facebook's IP address works fine.*

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
```

Q14/ 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

```
icsil1noteb147:~ barreto$ nslookup
> set type=A
> lca.epfl.ch
Server: 2001:620:618:10f:1:80b2:f07:1
Address: 2001:620:618:10f:1:80b2:f07:1#53

lca.epfl.ch canonical name = lca1srv2.epfl.ch.
Name: lca1srv2.epfl.ch
Address: 128.178.156.24
> set type=AAAA
> lca.epfl.ch
Server: 2001:620:618:10f:1:80b2:f07:1
Address: 2001:620:618:10f:1:80b2:f07:1#53

lca.epfl.ch canonical name = lca1srv2.epfl.ch.
lca1srv2.epfl.ch has AAAA address 2001:620:618:19c:1:80b2:9c18:1
```

Solution. *IPv4 address: `128.178.156.24`*
IPv6 address: `2001:620:618:19c:1:80b2:9c18:1`

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
```

Q15/ Filter the DNS packets in Wireshark. 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.

Solution. *This is a reverse DNS query as opposed to the previous ones.*

The `traceroute` tool works by sending the udp packet (in case of Linux and MacOSX) and ICMP packet (in case of Windows) with increasing TTL values until it reaches the destination. When TTL expires, the intermediate routers reply and that's how it knows all the intermediate machines. By default the `traceroute` tool makes a reverse DNS query for the IP address of each intermediate router, and then it displays the name in the output of the traceroute command. To disable this reverse query (and thus making the command faster), when typing the traceroute command you can use the "`-n`" argument in Mac and Linux, or the "`-d`" argument in Windows

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"
```

You can also turn an interface off without using *Terminal* app. Go to System Preferences and then click on Network. Next, click on the interface you want to change its configuration. Then, select Advanced button. In the new window, go to tab TCP/IP. Now, in the configuration of IPv4, you can turn off IPv4 or select Using DHCP for automatic IPv4 assignment.

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`, the password is announced on Moodle).

Use wireshark to observe the traffic. On your computer type

```
# ping www.facebook.com
```

Note that if it is not pinging with IPv6 by default, instead of `ping` command, you should use `ping6` on MacOSX and `ping -6` on Windows.

Q16/ 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.

Solution. *IPv6 and IPv4 packets may take different paths to reach the destination host, also at any given moment we could experience congestion in the network, thus RTT may be different. Differences are also in packet length, protocol used, etc.*

Repeat the test with the `traceroute` command from Section 2.4. Use:

In Linux or MacOS:

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

Note that if the `traceroute` command is not done by default with IPv6, you should use `traceroute6` command.

In Windows:

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

Note that if the `tracert` command is not done by default with IPv6, you should use `tracert -6` command.

Q17/ Write the result. Does the path to Facebook in the IPv6 Internet cross the same routers as in IPv4?

```
icsil1noteb157:~ mohiuddi$ traceroute6 www.facebook.com
traceroute6 to star-mini.c10r.facebook.com (2a03:2880:f11c:8083:face:b00c::25de) from
 2001:620:618:197:1:80b2:97d7:1, 64 hops max, 12 byte packets
 1  cv-ic-dit-v151-ro  0.477 ms  0.281 ms  0.433 ms
 2  cv-gigado-v100  0.388 ms  0.466 ms  0.378 ms
 3  c6-ext-v200  0.511 ms  0.469 ms  0.441 ms
 4  swiel1-10ge-0-0-0-2.switch.ch  1.168 ms  1.156 ms  1.055 ms
 5  swiel2-10ge-5-3.switch.ch  0.898 ms  1.000 ms  0.755 ms
 6  swice2-10ge-4-1.switch.ch  1.671 ms  1.651 ms  1.552 ms
 7  swice3-p23.switch.ch  1.647 ms  1.724 ms  1.712 ms
 8  2001:7f8:1::a500:6762:1  17.286 ms  17.302 ms  17.253 ms
 9  lo0.franco32.fra.seabone.net  30.441 ms  35.908 ms  30.567 ms
10  2001:41a8:600:2::162  17.802 ms  23.524 ms
    2001:41a8:600:2::15e  23.361 ms
```

```
11  po111.asw04.fra2.tfbnw.net  18.316 ms
    po114.asw01.fra2.tfbnw.net  19.627 ms
    po121.asw01.fra2.tfbnw.net  19.966 ms
12  po203.psw01c.frt3.tfbnw.net  18.722 ms
    po204.psw01c.frt3.tfbnw.net  23.821 ms
    po201.psw01b.frt3.tfbnw.net  19.562 ms
13  po3.msw1ac.01.frt3.tfbnw.net  19.592 ms
    po2.msw1ai.01.frt3.tfbnw.net  25.719 ms
    po3.msw1ad.01.frt3.tfbnw.net  28.020 ms
14  edge-star-mini6-shv-01-frt3.facebook.com  24.834 ms  18.878 ms  18.851 ms

tsf-484-wpa-4-040:~ mohiuddi$ traceroute www.facebook.com
traceroute to star-mini.c10r.facebook.com (31.13.64.35), 64 hops max, 52 byte packets
 1  cv-gigado-v484 (128.179.184.1)  72.081 ms  2.642 ms  1.149 ms
 2  c6-ext-v200 (128.178.200.1)  1.126 ms  3.746 ms  1.599 ms
 3  swiel2 (192.33.209.33)  2.093 ms  1.914 ms  1.825 ms
 4  swiel2-10ge-5-3.switch.ch (130.59.36.78)  3.270 ms  2.346 ms  2.210 ms
 5  swice2-10ge-4-1.switch.ch (130.59.37.65)  2.832 ms  3.245 ms  3.616 ms
 6  swice3-p23.switch.ch (130.59.36.210)  3.708 ms  3.118 ms  4.326 ms
 7  br02.ams1.tfbnw.net (80.249.209.164)  19.682 ms  18.969 ms  18.255 ms
 8  be2.bb01.ams3.tfbnw.net (204.15.20.10)  26.151 ms  27.995 ms  26.756 ms
 9  ae21.bb02.ams2.tfbnw.net (31.13.27.66)  25.323 ms  31.258 ms  26.080 ms
10  ae1.pr02.ams2.tfbnw.net (74.119.79.195)  18.956 ms  18.960 ms  18.236 ms
11  po102.psw01c.amt2.tfbnw.net (157.240.32.17)  18.176 ms  18.973 ms  18.399 ms
12  msw1am.01.amt2.tfbnw.net (173.252.66.217)  18.807 ms
    msw1ac.01.amt2.tfbnw.net (173.252.65.1)  22.155 ms
    msw1al.01.amt2.tfbnw.net (173.252.66.219)  19.376 ms
13  edge-star-mini-shv-01-amt2.facebook.com (31.13.64.35)  20.706 ms  20.008 ms  21.127 ms
```

Solution. *There are some routers with the same name in the two cases. It is not impossible that they are dual-stack routers. The path is however not identical!*

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

Q18/ 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?

Solution. *There is an IPv6 logo at the bottom of the page.*

Who put this logo on the page we received ? The web server did it. In this case the same web server is reached over IPv4 and IPv6 (in other settings they might be different) but the web server itself, when it is contacted by a client, knows on which network (IPv4 or IPv6) the HTTP request arrives (based on sockets, as we will see later in the course). The web server then runs a script that puts the IPv6 logo in the page when the request arrived over IPv6. Intermediate systems are of course not involved in this.

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.

Solution. *Traceroute works in both cases, and they traverse different routers.*

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

Q20/ Check the capture in Wireshark, is your connection to the webpage done with IPv4 or in IPv6?
Solution. *It depends on your operating system. For example, Mac prefers IPv6 if available.*

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

Solution. *It depends on your machine but in general IPv6 is preferred over IPv4 and decision is based on the DNS query. If the target host has an IPv6 address, your machine tries an IPv6 connection; if not it goes for IPv4. However, some vendors have decision-making algorithms that tracks the latency on the IPv4 or IPv6 network and based on that decide which network they will use.*

RESEARCH EXERCISES (OPTIONAL)

6 WIRESHARK VS TSHARK

You already have experience of Wireshark usage (2.2). 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. In the research exercise you will compare tshark and wireshark and see in which cases one tool is better than the other.

In the next section, we introduce you with tshark.

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 a certain interface and save it in `captured_packets.pcap` file, the following command can be used.
On Windows:
1. In powershell navigate to the folder where Wireshark installed using `cd` command
2. Run the following command

```
#./tshark.exe -i interface_name -w captured_packets.pcap
```

On Linux/Mac

```
# tshark -i interface_name -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. In order to get the names of interfaces you can use the `-D` option:

```
# tshark -D
```

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

Alice is soon going to have her holidays. She is searching for holiday offers on the web. She finds a very interesting and inexpensive 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.

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

Q22/ Please write below all the 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. ***Solution.*** *This tshark command prints Alice's details.*

```
tshark -r alice.pcap -Y http.request.method==POST -T fields -e
http.file_data | grep Alice
```

where -Y is for display filter options, -T is for specifying how you want to see the headers, and -e is for what field you want to get displayed. As the http response body should contain the name 'Alice', among all http post bodies, we are only interested in the body that contains the word 'Alice' and therefore, the grep command in the end.

Another possible command:

```
tshark -r alice.pcap -Y 'http.request.method == POST' -T fields -e text
```

6.1.3 EXERCISE 2

In this exercise you will compare the usage of Wireshark and tshark.

Q23/ How can you identify the TCP connections opened by visiting the `www.epfl.ch` webpage **using Wireshark**? Which filter you will use? Describe the steps that you have done and write the filter command that you used. Also write down the connections that you have found. Is there one, or are there several such connections? ***Solution.*** *1. Start Wireshark capture with filter: `tcp.flags.syn == 1 && tcp.flags.ack == 0`, the following filter will show only the first packet of the handshake, the one that is actually requesting that a connection be established*
2. In a browser open `www.epfl.ch`
3. Stop Wireshark
On the lab machine we observe 4 connections. By inspecting the destination IP addresses of these connections, we observe that one of them belongs to facebook and one to instagram. It means that by opening `www.epfl.ch`, the browser also download information from these services.

Now assume that you do not have access to GUI and only command line is available. It means that both browser and tshark should be opened from the command line.

Q24/ How can you identify the TCP connections opened by visiting the `www.epfl.ch` webpage **using tshark**? Describe the steps that you have done and write the command that you use. ***Hint:*** *In this case you will need to know how to run tshark and browser with predefined url simultaneously. Also, how run tshark capture for a certain amout of time.* ***Solution.*** *In order to solve the task, we suggest the following procedure:*
1. Run tshark capture for 20 sec (using duration option) in detach mode
2. In a browser open `www.epfl.ch` (using `Start-Process` on Windows, `open` on Mac)
3. Wait for 30 sec using `sleep` command in order to let tshark finish the capture
4. Parse the resulting pcap file using tshark with display filters
The capture time and sleeping time is taken arbitrary, making sure that the browser has enough time to open web page and tshark has enough time to finish the capture of all information.
The command that performs the steps above is:
On Mac: `tshark -i en0 -w captured_packets.pcap -a duration:20 & open -a "Google Chrome" https://www.epfl.ch; sleep 30; tshark -r captured_packets.pcap -Y "tcp.flags.syn == 1 && tcp.flags.ack == 0" -T fields -e ip.dst`

On Windows: `Start-Process .\tshark.exe -ArgumentList '-w C:\Users\lca2\Desktop\captured_packets.pcap' -i 4 -a duration:20'; start chrome www.epfl.ch; sleep 30; .\tshark.exe -r C:\Users \lca2\Desktop\captured_packets.pcap -Y "tcp.flags.syn == 1 && tcp.flags.ack == 0" -T fields -e ip.dst`