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, 2020
Deadline: September 30th, 2020 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. 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 VIRTUALBOX: INSTALLATION GUIDE

Most of the labs of this course will run in a virtualized environment that can be installed directly on your own computer. With the virtualized environment, you will be able to operate a network with several hosts, routers, and other communication equipment, all in your own machine. This considerably simplifies the operation of the labs and may prove useful outside this course whenever you have to test a communication system.

The virtualized environment is an emulated[1] environment, i.e. the virtual hosts and routers run the same code as real physical hosts and routers; only their hardware is replaced by the virtualized environment.

The virtualized environment is mainly composed of the following items:

- A virtualization software, like VirtualBox, provides hardware emulation to support virtual routers and standard PCs. Version `6.1` was tested for compatibility with the lab, but versions `6.0.x` should all work similarly.
- Next, we need to install the virtual machine in the virtual box. The virtual disk for the virtual machine is available on moodle.
- The virtual machine already has Mininet installed on it, Mininet provides emulation of networks and cables; it will be used to create network topologies from next lab onwards.

Download VirtualBox[2] and install it on your computer.

Download the virtual HD image `MininetVM-disk001.vmdk` from the link provided on Moodle.

2.1 YOUR FIRST VIRTUAL PC

We will now create a virtual machine.

1. Launch the VirtualBox software.
2. Create a new virtual machine, set the name to `MininetVM`, the type to Linux, choose version Other 64bit (and press next).
3. Assign 1GB of RAM (press next).
4. Choose "use an existing virtual hard disk file" and pick the uncompressed file `MininetVM-disk001.vmdk`.

Now that the machine is created, we need to add two features to it. First, we create a shared folder between the guest machine and the host machine (i.e. your computer). With this folder you can easily backup and transfer configuration scripts, snapshots and any file you might find useful during your labs. Second, we will enable the "copy to clipboard" functionality between your host machine and the guest machine, and between guest machines as well. Note that you must do these changes when the VM is not launched.

1. **For the the shared folder:** Right-click on MininetVM and click on "settings", in the "Shared Folders" tab, click on the blue folder with a green cross to add a new folder. Point the "Folder Path" to a folder of your choice on your host machine, and in the "Name" field write `shared`. Check the box `Auto-mount`. Point the "Mount point" to `/media/lca2/shared`.
2. **For the clipboard copy**: Again from the same machine settings window, click on the "General" tab, then on the "Advanced" subtab tab and set the shared clipboard to "Bidirectional". Click on "OK" at the bottom right of the settings window.

[1] In contrast, a *simulated* environment such as ns3 replaces hosts and routers with simplified code.

[2] You can also use a different virtualization software but we will provide support and instructions only for VirtualBox.

Lastly, we will attach a NAT to the VM for it to be able to access the internet via the host connection. This should be done by default, but in case it is not, follow the steps below.

1. Go back to the machine settings by right-clicking on MininetVM and selecting ”Settings”, go to the ”Network” and then select the ”Adapter 1” subtab.
2. Enable the adapter by checking the box “Enable Network Adapter”.
3. Next, under the field “Attached to”, choose “NAT”. Click on ”OK” at the bottom right of the settings window.

Congratulations, you have just created a virtual machine!

2.2 RUNNING THE VIRTUAL MACHINE

In VirtualBox select `MininetVM` and run it (Double-click on MininetVM).

Login using the username `lca2` and password `lca2`. (Password for azerty keyboards on the login page: `lcqè`).

At the first launch of the VM, there could be uptdates to be done, accept them. This step may take some time depending on the amount of updates and on your internet connection.

The virtual machine should already be connected to the Internet via the host’s connection. Open the Firefox web browser and go to a webpage to check that you are indeed connected.

In case you have issues launching your virtual machine, you may need to activate hardware virtualization. To do this you need to access the `BIOS` menu of your computer. You might need to power off your computer after enabling this option (not just a simple reboot!).

Check that the shared folder works: outside of virtual box on your machine, copy files to the shared folder. Now, on your virtual machine, find the shared folder and check that the files are there.

If you have any issues at this point (no Internet connection, cannot launch VM, cannot copy paste, cannot access shared folder) that you cannot resolve yourself, contact a TA for help.

2.3 LINUX CRASH COURSE (OPTIONAL)

This section is meant to provide a brief introduction to Linux commands and best practices. Feel free to skip this section and go directly to Section 3, if you are familiar with Linux. If you decide to skip this section, we encourage you to do the non-mandatory research exercise at the end of the lab. Note however, that these are the basic commands that you will need to be familiar with for all the remaining labs.

2.3.1 KEYBOARD LAYOUT

The keyboard layout on your virtual machine is `us` by default. To change it to a swiss one, open a terminal (LXTerminal, available on the desktop) and type the command:

```
$ sudo setxkbmap ch
```

However, you will have to type this command every time you reboot your system. If you want to make this command executed automatically upon booting the system, you need to write a configuration script (for example `keyboard_conf.sh`) and place it in the folder `/etc/profile.d`. To do so, go to the folder by typing the command

```
$ cd /etc/profile.d
```

then create the configuration file and open it with a text editor

```
$ sudo leafpad keyboard_conf.sh
```

and write the following code in it:

```
#!/bin/sh
setxkbmap ch
```

Then save the file and check that your newly created file exists: `ls` (this command shows you all documents in the folder that you are in). You can check its content in the terminal using the command `cat keyboard_conf.sh`. You have to set your file to executable with the following command

```
$ sudo chmod +x keyboard_conf.sh
```

Now if you check it again with `ls`, the name of the file should be in a different colour than before, this shows that it has been transformed into an executable.

You can test that it works by restarting your VM and making sure that after the reboot, the keyboard layout is in the desired language.

2.3.2 LINUX COMMANDS

The Linux distribution of the virtual machine comes with a friendly graphical interface. For configurating network interfaces however, we will use the terminal. Here are a few things you need to know:

Linux is a multi-user system that uses the Extended file system (e.g., ext2, ext3, ext4) to store files. In extfs each file has a unique owner (a user), belongs to a group of users, and has a set of permissions which define access rights to the file (read, write, and/or execute) for the owner, the group, and everyone else.

Each normal user has a home directory, that is referred to by the symbol ∼ (tilde). To change directories in the terminal use the command `cd` followed by the name of the directory. This can be a relative name, such as `Documents`, or an absolute name, such as `/etc/init.d` (i.e., beginning with the `/`, which is the root of the filesystem). It can also be the home directory (i.e., `cd ~`). To move up in the tree, i.e., out of a directory, use `cd ..` (two dots). The current directory is always represented by a single dot, so `cd .` does nothing. To display the current directory use `pwd` (print working directory).

Try it yourself: open a terminal [3]. Use `pwd` to show the current directory. Then `cd` to `~/Tutorial/`. Use the Tab key for auto-complete. If you cannot find the ∼ tab on your keyboard, try using the F6 key.

In the terminal, you can list the files in a directory by using the command `ls`. You can add switches to a command. For example, if you want to see detailed attributes of all the files in a directory (including the permissions), you can use the `-l` switch:

[3] Tip: The shortcut to open a terminal is `Ctrl+Alt+T`.

```
$ ls -l
```

You should see something like this:

```
lca2@lca2:~/Tutorial$ ls -l
total 4
-rw-r-r-- 1 lca2 lca2   0 Aug 18 12:14 emptyFile.txt
-rw-r-r-- 1 lca2 lca2 12 Aug 18 12:16 helloWorld.txt
```

You can output the contents of a file to the terminal by using commands such as `cat` or `less`:

```
$ cat helloWorld.txt
```

You can see above that the permissions of the `helloWorld.txt` file are `-rw-r--r--`, that it belongs to the user `lca2` and the group `lca2`, and that it is `12` bytes long. The permissions string is 10 characters long. The first character is either a dash `-` for regular files, or other letters for special files (a `d` for directories, etc.). The next three characters give the permissions for the owner of the file, in this case `rw-`. This means that the owner has the right to read the file (`r`), to write/modify the file (`w`), but not to execute the file. If the file was executable, in the third position there would be an `x`. The next three characters describe the permissions of the group, and the last three characters the permissions of all the other users in the system. In this case, the file is read-only for the group and for everyone else.

A file that the user `lca2` does not want anyone to see but herself would have permissions `-rw-------`, whereas a file with full rights for everybody would have permissions `-rwxrwxrwx`.

A file's permissions can be changed by using the command `chmod`. You need to specify whose access rights to the file you want to alter: of the user who owns it (`u`), of the group (`g`), or of others (`o`), whether you want to add (`+`), or remove (`-`) a right, and which right you mean (`r`, `w`, or `x`).

For example,

```
$ chmod o-r,g+w emptyFile.txt
```

removes the reading right for other users than the owner or the group and adds writing for the group. To change the ownership of the file, use `chown`.

When you issue a command in the terminal, you are in fact running a certain executable file. The command interpreter (or the shell) looks for these executable files in one of the several directories specified in the `PATH` environment variable. To list the contents of this variable, run

```
$ echo $PATH
```

The character `$` before `PATH` indicates that we want to display the contents of the variable `PATH`; without it, the command would simply display the string `PATH`. The directories are separated by semicolons, and they

are searched in order. To see which executable you are running, use the command `which` followed by the name of the executable. For example, `which ls` displays `/bin/ls`, the location of the `ls` executable.

Note that the current directory (`.`) is not in the `PATH` for security reasons (a miscreant user might create an executable called `ls` in some directory, which in fact erases the given directory instead of listing it). Therefore, if you really want to execute a file in the current directory, you need to specify the path (the current directory), i.e., to type `./some_script` instead of simply typing `some_script` (the latter results in a "file not found" error).

Normal users cannot alter system configurations files (they do not have permission). For this reason it is safer to use a Linux machine as a normal user, and not as an administrator. This way, you cannot do too much harm.

There is a super-user (administrator) called `root` that has absolute rights (i.e., can do **anything**). In the terminal, the command prompt for a normal user ends with a dollar sign `$`, whereas for the `root` the prompt ends with a hash `#`.

IMPORTANT In these labs, whenever you see the hash `#` sign in front of a command that you are supposed to type, it means that you need `root` access.

There are users called "sudoers" that are allowed to run a single command as `root` (the user `lca2` in our virtual machine is such a user). This is achieved by typing `sudo` followed by the desired command. You will then be prompted for the password of the user.

If you want to run a terminal in `root` mode, type the command `sudo su`. You will then be prompted for the `root` password and you will switch to `root` mode. the password for root is `lca2`.

OTHER USEFUL COMMANDS: if you launch an application using the terminal ex: `$ leafpad`, then the application will open but the terminal you used will be dedicated to the application and you won't be able to type other commands in it, to prevent the use of too many terminal windows simultaneously, you can detach a command using `&` at the end of your command: `leafpad &`, this will launch the application and allow you to type other commands in the same terminal afterwards. Note that if `sudo` is needed to launch the application, the use of `&` may not work because the detachment of the command does not allow you to type the password required by sudo. Another useful command allows you to search for files on the entire machine or in specific branches of the arborescence:

```
sudo find <branch> -iname <file>
```

you can replace `<branch>` by `/` if you want to search everywhere or with the path to where you want to search e.g. `/etc/`, you can replace `<file>` with the name of the file you are searching for `keayboard_conf.sh` or if you don't remember the exact name you can write elements of it and use `*` to signify anything: `*board*`. To sum up, if you want to find your keyboard configuration file but only remember it is somewhere below `/etc/` and that the word "board" is in it you can type

```
sudo /etc/ -iname *board*
```

This will output the path to your file and potentially other files that also satisfy this description.

From the command line, you can write several commands at a time using `|`. For example if you want to run command2 on the output of command1 you can type

```
command1 | command2
```

For example, if you place yourself in the folder `/etc/init.d` and run

```
ls | grep key
```

then the first part `ls` outputs all documents and on this output, the second part only outputs the filenames that include the string `key`, thus it should output your `keyboard_conf.sh` file and no filenames that do not contain the string `key`. Notice the use of `grep`, this command enables you to search for specific strings of characters in an output. Finally, you can write the output of a command to a file with >:

```
command > file.txt
```

or you can append the output of your command to a file which already contains other information using >>:

```
command >> file.txt
```

2.3.3 BEST PRACTICES

In these labs you will often type configuration commands in the terminal, usually one by one, to observe and understand their effects. However, after a reboot, the effects of these commands are usually lost, and you need to type them again, which is cumbersome.

We recommend the following practice:

Keep a text editor open in the virtual machine (for example "Leafpad", located in Accessories, or `nano` in another terminal). Whenever you type a configuration command in the terminal, paste it in the editor. In Linux it suffices to select a text to copy it in the clipboard. For pasting use the middle mouse button. Otherwise use the standard "right-click" + Copy (but be warned that this might not work in all terminals). The shortcuts for copy and paste on the terminal are `Ctrl+Shift+C` and `Ctrl+Shift+V`, respectively.

Save the resulting file in your home directory (for example as `conf.sh`). When you reboot, you can run all the commands in the file as root by

```
# sh conf.sh
```

or as a regular user via `sudo` by

```
$ sudo sh conf.sh
```

2.3.4 ADDITIONAL INFO

There are two main software packages that provide tools for configuring the network: the older, standard `net-tools` (provides `ifconfig`, `route`, `netstat`), and the newer and more powerful `iproute2` (provides `ip`, `ss`). Both are installed on the virtual machine, but we will focus primarily on the second set of tools (here is an angrily argumented viewpoint `http://inai.de/2008/02/19`).

3 THE IPV4 INTERNET AND NETWORK PACKET INSPECTION

Launch MininetVM and do the lab in there.

3.1 FINDING THE INFORMATION ON THE CONNECTION

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

To disable IPv6, use the following commands from the *Terminal* app

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

You can check that your command worked by checking the value of the variable that you just tried to set to 1:

```
sudo sysctl -a | grep disable_ipv6
```

After disabling the IPv6 connectivity, we now want to determine the following information:

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

Use the following commands in the *Terminal* app

```
$ ip addr show
$ ip route show
```

Q1/ List your findings here:

- IP address: 10.0.2.15
- Network Mask: 255.255.255.0
- Default Gateway: 10.0.2.2

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

Solution. *In this case, the IP address is private, 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 different from the one given to the Ethernet adapter. OR We have a range of private ip addresses and we can see that this ip address belongs to this range, it is private. The netmask or prefix is used to distinguish the "network" and the "host" parts of an IP address.*

3.2 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 to 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
`https://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.

Wireshark is already installed on your virtual machine. Start it (as administrator) by typing `sudo wireshark` on the terminal. On the main page, under "Capture" you have a list of interfaces that you can select to observe its passing traffic. Next to the interface name, the amount of traffic on it is illustrated by a signal. Select the interface that is currently used for internet connectivity and capture it by clicking on the blue shark fin.

Q3/ Write a filter command that displays 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.

3.3 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.epfl.ch`. Which exchanges of messages is happening after the 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?

Solution. *First a DNS query is performed, next a ping request is sent to the IP address of epfl. This is the theory and we observe such packets on wireshark.*

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 EPFL's load balancing system), etc.

Q5/ In a browser open `www.netflix.com`. 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.*

3.4 TRACEROUTE AND NETSTAT

traceroute is a tool for displaying the route to a destination:

```
# traceroute www.grimper.ch
```

Q7/ Start Wireshark and do `traceroute` to `www.grimper.ch`. What is the IP address of facebook? Is the system of the virtual machine using ICMP, TCP or UDP for `traceroute`? Write down the result of the traceroute.

```
traceroute to www.grimper.ch (104.26.11.137), 30 hops max, 60 byte packets
 1  cv-svc-ico1-2.epfl.ch (10.0.2.2)  0.303 ms  0.253 ms  0.230 ms
 2  cv-svc-v492-ro.epfl.ch (128.179.192.1)  1.605 ms  1.911 ms  2.014 ms
 3  cv-backbone-cv-svc-40.epfl.ch (10.0.2.40)  8.846 ms  9.483 ms  9.752 ms
 4  c6-ext-cv-backbone-97.epfl.ch (10.0.2.97)  3.061 ms  3.881 ms  4.054 ms
 5  swiel2.epfl.ch (192.33.209.33)  4.143 ms  4.323 ms  4.457 ms
 6  swige3-100ge-0-0-1-1.switch.ch (130.59.36.82)  4.761 ms  4.504 ms  4.735 ms
 7  swice1-100ge-0-1-0-6.switch.ch (130.59.38.193)  4.072 ms  3.038 ms  3.079 ms
 8  ams-ix.as13335.net (80.249.211.140)  19.598 ms  21.292 ms  21.540 ms
 9  104.26.11.137 (104.26.11.137)  18.372 ms  18.790 ms  19.187 ms
```

Solution. *We need to filter the packets with the destination address of the grimper server: `ip.dst == grimper_ip_address`. The IP address of grimper that we saw is 104.26.11.137 but they may have several. You should see UDP packets as this is what Linux uses.*

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:

```
# 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 the destination IP address is the IP address of the `www.epfl.ch` webpage. Is there one, or are there several such connections?

Solution. *Several connections are established, on the lab machine there are 2 connections.*

3.5 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 yourethernet interface? How can you find a MAC address of your default gateway?

Solution. *On my machine the MAC of the wireless interface `enp0s3` is `08:00:27:a2:c8:64`.*
In order to find the default MAC address of the default gateway, run `ip route show` to see the IP address of your gateway. Then ping that IP address. Finally, run `arp -a` and find the gateway IP address, you'll see the MAC address on the same line.

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 `52:54:00:12:35:02`.

Q10/ If you and your friend were on the same subnet, could 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 his MAC address.*

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

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

4 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`.

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 `www.grimper.ch` 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 grimper that you discovered in Sections 3.3 and 3.4. Does it work?
Solution. *Since there is no need to resolve a name, the ping to ikea'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 `set type=A` for IPv4 or `set type=AAAA` for IPv6. Then type `epfl.ch`. Give its IPv4 and IPv6 addresses.

```
  lca2@lca2-vm:~$ nslookup - 8.8.8.8
> set type=A
> epfl.ch
Server: 8.8.8.8
Address: 8.8.8.8#53

Non-authoritative answer:
Name: epfl.ch
Address: 128.178.222.108
> set type=AAAA
> epfl.ch
Server: 8.8.8.8
Address: 8.8.8.8#53

Non-authoritative answer:
Name: epfl.ch
Address: 2001:620:618:1de:1:80b2:de6c:1
```

Solution. *IPv4 address: `128.178.222.108`*
IPv6 address: `2001:620:618:1de:1:80b2:de6c:1`

Restore now your initial DNS configuration.

Start a capture in `wireshark` and do a traceroute in IPv4 to `www.grimper.ch`. Focus on one line of the form:

```
name (IPv4address)  time ms time ms ... time ms
```

Q15/ Filter the DNS packets in Wireshark. Look at the capture and identify the packet in which you see the same `name`. 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*

5 IPv4 AND IPv6

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

To restart the IPv6 connectivity:

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

and reboot the machine. As previously, you can check that your command worked using

```
sudo sysctl -a | grep disable_ipv6
```

Use wireshark to observe the traffic. On your computer type

```
$ ping6 www.grimper.ch
```

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 `traceroute6` command.

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

```
traceroute to www.grimper.ch (2606:4700:20::ac43:49f1) from 2001:0:53aa:64c:1022:302e:7f4c
 1  6to4.fra1.he.net (2001:470:0:150::2)  143.336 ms  29.087 ms  28.652 ms
 2  10gigabitethernet6.switch2.fra1.he.net (2001:470:0:150::1)  110.652 ms *  49.122 ms
 3  * * *
 4  * 2400:cb00:71:1024::a29e:5f9c (2400:cb00:71:1024::a29e:5f9c)  84.822 ms  50.652 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!*

Q18/ If you access a webpage via an IPv4 connection or an IPv6 connection, do you think it has to be the exact same page? Can you imagine by which mechanism 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. *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 scripts that can have different instructions depending on whether the request arrived over IPv4 or IPv6. Intermediate systems are of course not involved in this.*

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.*

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 the 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 (3.2). 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, you first need to create the file and then use the following command:

```
# 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

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. Use the shared folder to place the provided `alice.pcap` file on the Desktop of your virtual machine. In the terminal, place yourself on the Desktop. You should use tshark commands 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
```

Q23/ What is Alice's birth date? ***Solution.*** *Alice's birth date is the 26 of August 1992*

.