TÉCNICO LISBOA

Computer Networks — 2023/24, Q2		**Assignment:**	Lab 2
Network Tools & Configuration		**Version:**	1.0
Issued:	2023-11-13	**Submission Due:**	2023-11-19

Submission note. You will need to submit all the answers to this lab's questions in Moodle, under "Lab 2: Network Tools & Configuration".

1 Networking tools

The goal of this lab is to have hands-on experience using standard networking tools like `ping`, `traceroute`, `ip`, `wireshark`, and `iperf`, to discover network topologies and correctly identify network problems.

Using the provided virtual machine:

1. Execute the command `git pull origin master` in `/home/rc/lab-files`.
2. Run the Core Emulator
3. Open `/home/rc/lab-files/lab-net-tools/net-tools-1.imn`.

1.1 `ip`

`ip` is a networking tool that allows you to control interfaces, routes, devices, and tunnels. Use `man ip` to check its manual page or `ip -h` to print the help menu. You can use it to show all the network interfaces with `ip address show`.

Q1 Use the `ip` tool to catalog every host with their respective interfaces and IPs (with corresponding subnet masks), *except* for Carol.

Q2 Notice that Carol has no IP. Use the `ip` command to assign the IP `10.0.2.20/24` to the interface `eth0` of Carol's machine. Transcribe the complete `ip` command to Moodle lab quiz. After assigning the IP, add a default gateway to the machine so it can find the other networks: `ip route add default via 10.0.2.1`. Now, run the command `ping 10.0.2.20` in alice to check the connectivity with carol. Copy+paste from the terminal a single line that demonstrates that there is connectivity between alice and carol.

1.2 `traceroute` and `wireshark`

`wireshark` is a packet sniffer. It allows users to observe every packet that is either received or sent by any interfaces in a given host. To open wireshark in a given host's interface, simply right-click the host on `Core` and select `Wireshark`, followed by the interface you want to sniff. To filter the packets that are shown by the interface, you can input filter expressions in the display filter (the horizontal input bar at the top of `wireshark`).

`traceroute` allows you to discover the paths taken by packets as they traverse the network from the sender host to the receiver. Again, you can access its man page using `man traceroute`, or print its help menu with `traceroute --help`. If, for example, one

wanted to discover the paths taken by packets going from server A to server B, one would run `traceroute B` inside server A.

To understand how `traceroute` works, it's important to understand 3 things. First, every IP packet includes a Time-To-Live (TTL) field that is *decremented* at each hop/router towards the destination. When a packet arrives at a router, is decremented, and the TTL become zero, the packet is dropped. The goal of the TTL field is therefore to prevent packets from entering a (potentially infinite) network loop, when things go wrong.

`traceroute` uses the TTL cleverly to discover the paths traversed by packets on their way to the destination. For this purpose, it uses the TTL to "incrementally" get closer to the desired destination. First, it sends UDP/IP packets with TTL=1. When this packet arrives at the first hop/router, it is decremented, the TTL is zero, and the packet is dropped. When this occurs, an ICMP `Time to live exceeded in transit` packet is sent back to the source. Then, the source records the IP of the host that sent back the ICMP packet, and increases the TTL of the subsequent packet to 2, to discover the second hop. This process continues (TTL=3, TTL=4, etc.) until the source no longer receives ICMP error packets. At that point, `traceroute` outputs the IPs it collected for the entire route from source to destination.

Q3 Open `wireshark` on interface `eth0` of `router1`. Write `udp` in the display filter, and press enter to show only UDP packets. Now run `traceroute` from alice to bob. Use the output shown in `wireshark` to explain how `traceroute` works, by describing what happens for the different TTLs in the quiz. *Note: Each `traceroute` packet associated with a given TTL is sent 3 times.*

Q4 Catalog the path taken by the packets going from the following pairs of sources and destinations: alice->bob, alice->carol, alice->dan, dan->bob, dan->alice. Annotate the host's name of each hop instead of its interface's IP. *Note: your answer should be presented as a table with 2 columns: "Direction" (with, for example, "alice->bob") and "Route" (with, for example, "alice->router1->router2->bob").*

Q5 Notice the strange paths taken by packets between alice and dan. What is going on out of the ordinary? Explain this strange behaviour.

Q6a Based on **Q5**, when you run `traceroute` on alice, does it find the alice->dan path, or the dan->alice path?

Q6b How could you discover the opposite path?

1.3 `ping`

`ping` allows you to send ICMP echo requests to another machine (a.k.a to *ping* the machine). This is a great tool to check connectivity between hosts. To ping machine B from A, just execute `ping B` on A.

Note. You can filter ICMP packets in `wireshark` by using the `icmp` filter.

Q7 Send ICMP requests from alice to dan, and watch the interface `eth1` from `router1` using `wireshark`. Do you see ICMP requests and replies, just requests, or just replies?

Q8 Send ICMP requests from alice to dan, and watch the interface `eth2` from `router1` using `wireshark`. Do you see ICMP requests and replies, just requests, or just replies?

Q9 Explain what you saw in **Q7** and **Q8** with the information retrieved having analyzed the network with `traceroute`.

1.4 `iperf`

Using `iperf`, we can quickly run performance tests in our network. For this purpose, we first need to have a iperf server instance running, to which the client instance will connect later on. Then, the client generates traffic and sends it to the server. The server then computes the maximum throughput achieved in the connection.

The `iperf` manual page can be accessed by `man iperf`, and its help menu can be seen with `iperf -h`.

To run the server instance, you can run `iperf -s` in a given machine. Then, connect to the server with the `iperf` client using `iperf -c <SERVER_IP> -t 5` (`-t 5` just tells the client to generate traffic for 5 seconds).

Q10a Measure the throughput achieved between alice and bob. What was the result reported by the server?

Q10b What is the throughput achieved between alice and carol?

2 Fixing a networking issue

Open `/home/rc/lab-files/net-tools/net-tools-2.imn`. This network topology is identical to `net-tools-1.imn`, except that now there is a malfunctioning link *somewhere*.

Q11 Discover the malfunctioning link in the network. Which one was it? *Hint: remember ping?*

You can fix the link by stopping the session, right-clicking on top of the link and selecting `Delete`. Next, select the `link tool` option on the left vertical bar and connect the hosts missing the link, thus “replacing the faulty cable”. Check if the anomaly was fixed.