Lab 2 - Networks and protocols

TI602

In this lab we will configure a simple network using two virtual machines and analyze TCP and UDP traffic in order to understand the different functionalities.

The work must be organized in teams of two students (binomial), and you must produce a report containing for each question: description of the experiment, command line + screen output and analysis. The report must be uploaded on Moodle at the latest 4 days after the Lab. Each member of the team must do all the installation, configuration steps and experiments separately on his machine, the teamwork is reserved for the analysis part where you must exchange, discuss, and produce a common answer.

We need for this lab, two Linux virtual machines. if you haven't installed the machines yet, follow the steps bellow :

- Install Virtualbox on your physical machine: https://www.virtualbox.org/
- Download Linux Ubuntu 20.04 as ISO image file: https://ubuntu.com/download/desktop
- Create a new virtual machine on Virtual box : "Machine → new" with at least 2GB of RAM and 25GB of storage space on a VDI type hard disc
- You can always change settings after the creation in the Setting tab
- Install Ubuntu on the created virtual machine: "Setting → storage → add optical drive" then add the downloaded image, start the machine and follow the installation instructions
    - Be careful to select "install" and not "try"
    - Do not forget to set a root user password

Part 1 – Network Configuration

Before starting the **two** virtual machines, create a new network interface: "Setting → Network → adapter 2 → activate interface" then, configure the network access to *internal* **only for the adapter 2**: "Setting → Network → Network access → Internal". This configuration allows the two machines to be connected, all the configurations during the lab must be on this interface. The first Interface that should stay on NAT is used only to access internet, do not change its configuration during the lab.

1. Start the **two** machines and configure the network interfaces. The manipulation of the network interfaces is done using the ifconfig command. To obtain information on all the interfaces available on the system:

    ```
    > ifconfig -a
    ```

    Without the -a option, the command will only display interfaces that are currently active.
    You can turn on / off an interface using the commands:

    ```
    > ifup [interface]
    > ifconfig [interface] up
    > ifdown [interface]
    > ifconfig [interface] down
    ```

Here is a typical command to configure an interface:

```
> ifconfig "interface_name" 192.168.10.5 netmask 255.255.255.0
```

Note:
If the mask is not specified, it is calculated according to the class of the address.
An entry corresponding to the network associated with the interface is automatically added to the routing table.

2. Test the connectivity between the two machines using the command *ping* on both directions ():

```
> ping "destination_address"
```

Part 2 – TCP and UDP traffic analysis

Utilities:

- **tsock** : TCP or UDP traffic generator : **available on Moodle in the lab 2 section**
- **tcpdump** : tool for capturing traffic circulating on the network

Notices and manual available:

- tsock user guide
- Instructions for use of tcpdump
- man command

Using the program tsock

1. Perform a data exchange between a source tsock (on a first Linux machine) and a sink tsock (on the other Linux machine), via UDP then TCP, using a port number of your choice between 5000 and 9000.

   In both cases, start the tsock sink program before the source tsock program.

   a. Do you see disequencing using UDP, using TCP.
   b. Do you see losses using UDP, using TCP.
   c. Propose and realize an experiment which highlights the unreliability of the service offered by UDP (you can use the options -l ##, -n ## and if necessary -w on the Source).

2. Repeat question 1 by running this time the source tsock **before** the tsock well.

   a. What do you observe using UDP, using TCP.
   b. For UDP, check whether the transmitted data is received by the tsock sink program. Do you think that the behavior of UDP is acceptable with regard to the service it offers?

Using the program tcpdump

3. Repeat the experiment of question 1 and capture (via tcpdump) the corresponding traffic by filtering the port number used by the sink:

   Start tcpdump **in another terminal** on the Source or the Sink machine as follows:

   ```
   > sudo tcpdump port "port" -i "interface_name"
   ```
   where “port” designates the port number used by the Sink.

a. Highlight that UDP is a connectionless protocol.
b. Highlight that TCP is a connection oriented protocol. Find the connection establishment, transfer and connection closing phases.
c. Analyze the data transfer phase of TCP and highlight (through an experiment) the causes TCP to not preserve the same messages boundary as they were sent.

4. Repeat the experiment of question 1 (in UDP or TCP) and capture the traffic corresponding (filter on the port number used by the Sink), this time visualizing the content of captured **Ethernet** frames using the option -e.
   To do this, you need to use the command:

   ```
   > sudo tcpdump port "port" -I "interface_name" -e
   ```
   where "port" designates the port number used by the Sink.

   By relying on the format of the Ethernet, IP, UDP and TCP headers presented in the course, Determine:
   a. the Ethernet and IP addresses of the source and destination machines;
   b. the information indicating that the frame carries UDP or TCP;
   c. the port number used by the sink;
   d. the port number used by the source program, where does it come from?

5. So far, you've only done transfers from one machine to one other (unicast transfer).
   a. Remind what is called a transfer in "broadcast" mode.
   b. Using the command `> tcpdump -xx broadcast`, determine experimentally the broadcast address of the network.