



Middle East Technical University



Department of Computer Engineering

CENG 435

Data Communications and Networking

Fall 2021–2022

THE - 3

Due date: 12.01.2022 - 23:59

1 Introduction

This assignment will cover creating a network topology using [mininet](#) and explore ICMP messages and their format with the created topology using wireshark. By using **mininet**, we can simulate network topologies with multiple switches, hosts, routers etc. in a single machine.

This homework will have two sections. In the first section, you will work with the **default minimal topology** of mininet by sending ICMP messages from one host to another and capture the packets using wireshark. In the second section, you will create a more complex network topology using Python API of mininet and answer the questions about the topology that you have created.

In order to begin your implementation, you first have to install **mininet**. You have several options;

- Install it to your local machine following [this guide](#) (if you have other choices, not recommended)
- Use the [virtualbox image\(ubuntu server 18.04\)](#) we provide where mininet and all its dependencies are installed. This image does not have gui. You have work with it using ssh. If your computer has low resources, you can consider using this image. Username: **mininet**, Password: **mininet**
- If you can spare additional resources, you can download [virtualbox image\(ubuntu desktop 18.04\)](#). Mininet, wireshark and all of its dependencies has been installed in this image. You don't need to install or configure anything, just import the image and start the questions. Username: **mininet**, Password: **mininet**

Details about how to install mininet is found in its [github page](#) as well. It is highly recommended to check out [mininet walkthrough](#) as well.

Important Note: If you want to get through this homework as fast as possible, you should just use [ubuntu server 18.04](#) or [ubuntu desktop 18.04](#) images we gave to you.

After you finish the homework, you have to submit a .pdf report and a .py file for the topology that you have created in the second part. For more detailed explanation go to section 6.

In the next section usage of the **virtualbox image** will be explained step by step. If you want to directly install mininet to your computer (which we do not recommend), you can skip the next section.

2 How To Use Virtualbox Image (Ubuntu Server or Desktop 18.04)

If you do not want to do this homework with one of the images we have provided, you can skip this step. Otherwise, make sure that you have installed virtualbox and its extension pack from [this link](#). You also need to enable **virtualization** from your bios settings, if it is disabled, you can enable it like in [here](#)

If everything is installed successfully, you can now download [ubuntu server 18.04](#) or [ubuntu desktop 18.04](#) that we have created and import it into virtualbox. You can import it by double clicking the **.ova** file while virtualbox is open. You should see the page below.

Note: If you want to use **ubuntu desktop image**, just import it and start the first part of the homework. You are good to go in that case.

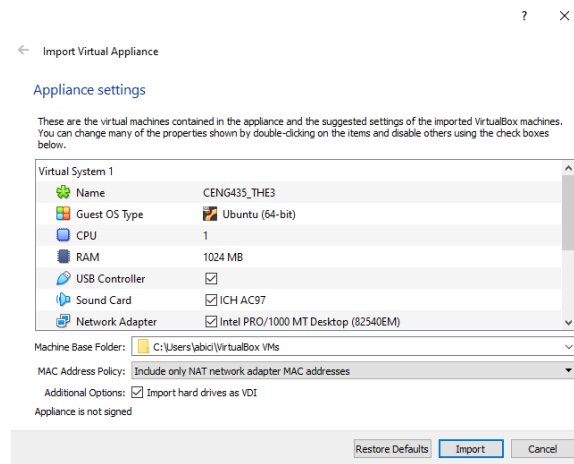


Figure 1: How to import the image

Check **mac address policy** and **additional options** looks same in the picture and click import. After importing, if you try to start the imported virtual machine, you may get this error;

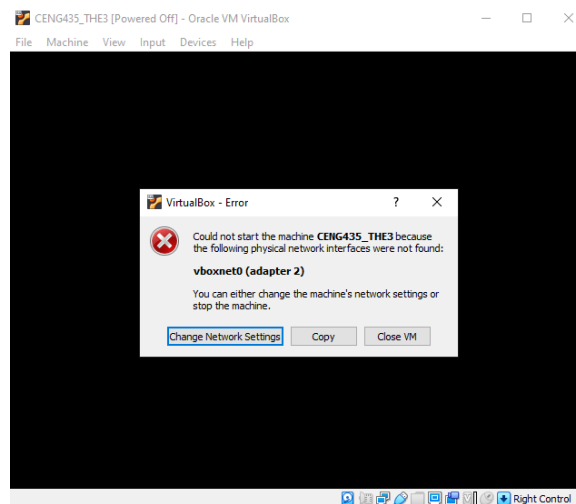


Figure 2: vboxnet0 error

This is because we did not configure host network manager and the virtual machine we have imported. We have to configure a host-only ethernet adapter because we want to connect to our virtual machine with `ssh`(this is very important to run wireshark from inside the virtual machine). In order to solve this, first close the error message and vm and go to **Host Network Manager**;

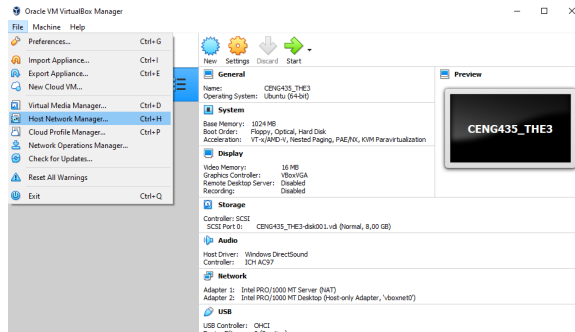


Figure 3: host network manager - 1

After that, click on **create**. If you get errors in this step, try running virtualbox as administrator(sudo). If this does not work, there might be a problem in installation so reinstallation can work. After a New Host-Only Ethernet Adapter is created, click on its properties and you will see two tabs; **Adapter** and **DHCP Server**. Make sure these tabs looks like these pictures;

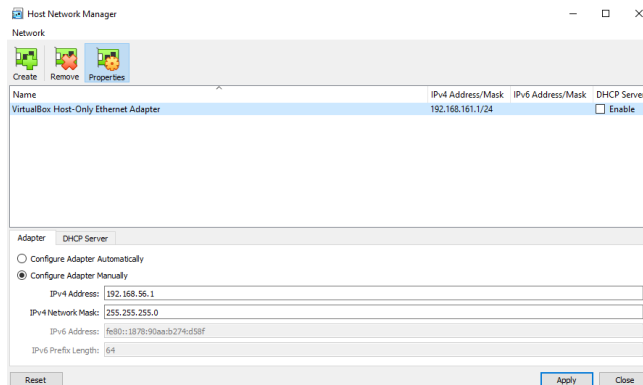


Figure 4: host network manager - 2

and

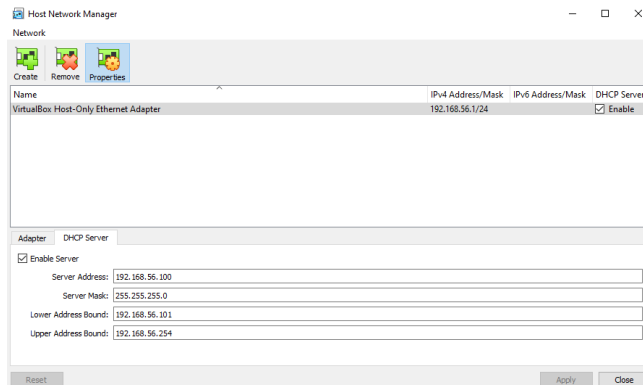


Figure 5: host network manager - 3

After that, we only need to change the settings of the imported virtual machine. By selecting **CENG435_THE3** machine, click on settings;

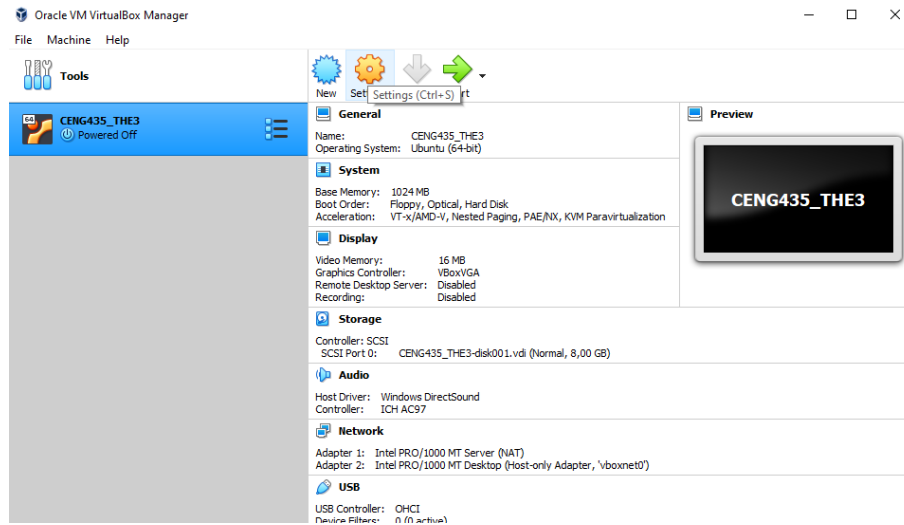


Figure 6: host network manager - 4

and go to network settings and make sure that adapter2 is the new host-only ethernet adapter that we have just created;

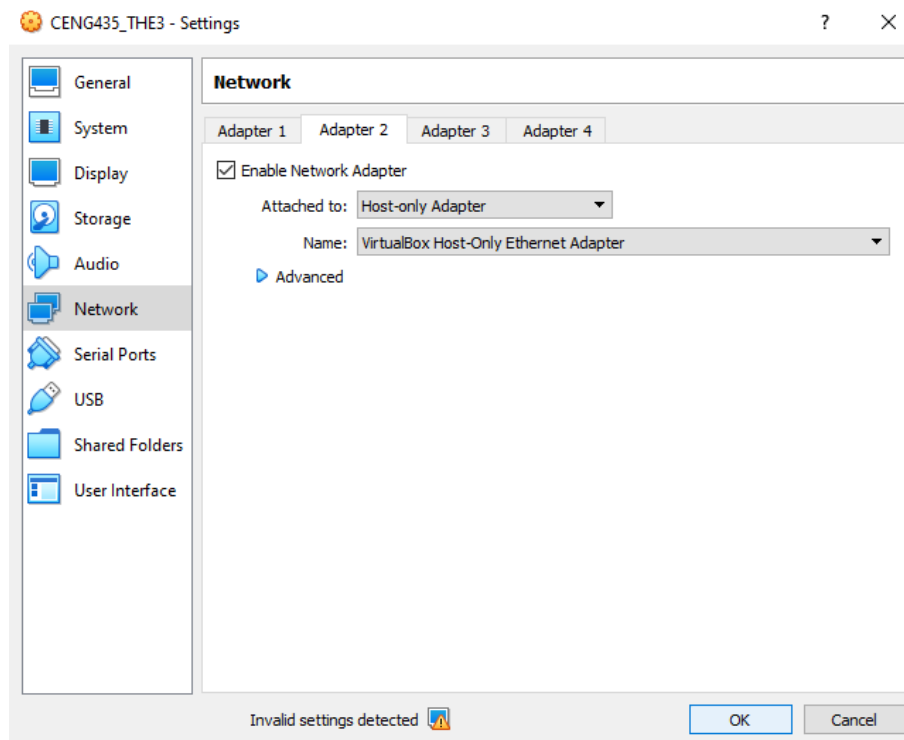


Figure 7: host network manager - 5

In windows hosts, you have to restart your computer after adding the host-only ethernet adapter to get this working. Now you can run the imported virtual machine. Its username and password is **mininet**. Keep in mind that the keyboard settings of the machine is english. After you log in to the vm, you should see this screen;

```
Ubuntu 18.04.6 LTS mininet-vm tty1
mininet-vm login: mininet
Password:
Last login: Sun Dec 26 06:35:26 PST 2021 on tty1
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-163-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection
or proxy settings

mininet@mininet-vm:~$ _
```

Figure 8: inside virtual machine

Now type **ifconfig** and see the ip address given to you in order to connect to this virtual machine from your host machine.

```
mininet@mininet-vm:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.101 netmask 255.255.255.0 broadcast 192.168.56.255
    ether 08:00:27:f6:12:b7 txqueuelen 1000 (Ethernet)
    RX packets 12 bytes 1780 (1.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2 bytes 680 (680.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 120 bytes 9192 (9.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 120 bytes 9192 (9.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

mininet@mininet-vm:~$ _
```

Figure 9: checking ip adress of the virtual machine

We have to look at **eth0**'s ip and see that it is **192.168.56.101** for this machine. It might be different in your case. We will use this ip address for connecting with ssh.

Now open your terminal in the host machine and type the password afterwards(mininet);

ssh -Y mininet@ip_address_we_found (1)

If everything is done correctly, you should be able to make an ssh connection with the virtual machine. Now we have to start wireshark from the virtual machine. In order to do that, you have to write this to terminal;

sudo wireshark &>/dev/null & (2)

Now you should be able to run wireshark as well. If everything works as expected, you are now safe to finish the homework without major issues.

3 PART 1 - ICMP With Minimal Topology

If you have successfully ran the virtual machine, connected it with ssh from your host machine and ran wireshark successfully, or installed mininet to your local machine (not recommended) by looking at the install directives in [github page](#) you can now proceed into the first part of homework.

In this part, first open wireshark. If you are using mininet inside a virtual machine, you have to write the above code inside the ssh session. Calling this code directly from the VM does not work. Initially wireshark looks like this(In VM);

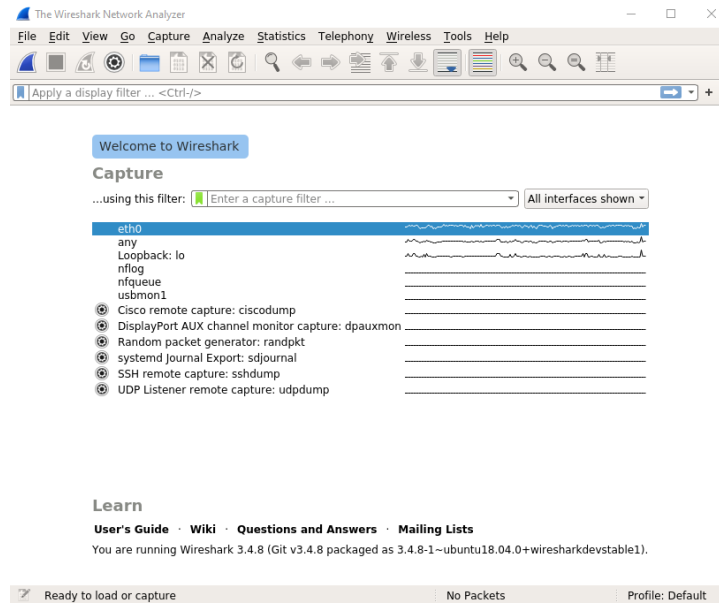


Figure 10: ws1

Now, in the terminal(ssh connection to VM if you installed the image) type;

```
mininet@mininet-vm:~$ sudo mn --topo minimal
```

Figure 11: tm

This should start the minimal topology where there are two hosts(h1 and h2), one switch and a controller. And you see that you are now on the **mininet command line interface**. We will write some commands here. Let it open for now.

```
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> █
```

Figure 12: tm2

If you look at wireshark just after this step, you see that there are newly added interfaces(s-eth1 and s-eth2). In order to proceed further, just listen to **s1-eth1** and apply filter **icmp**

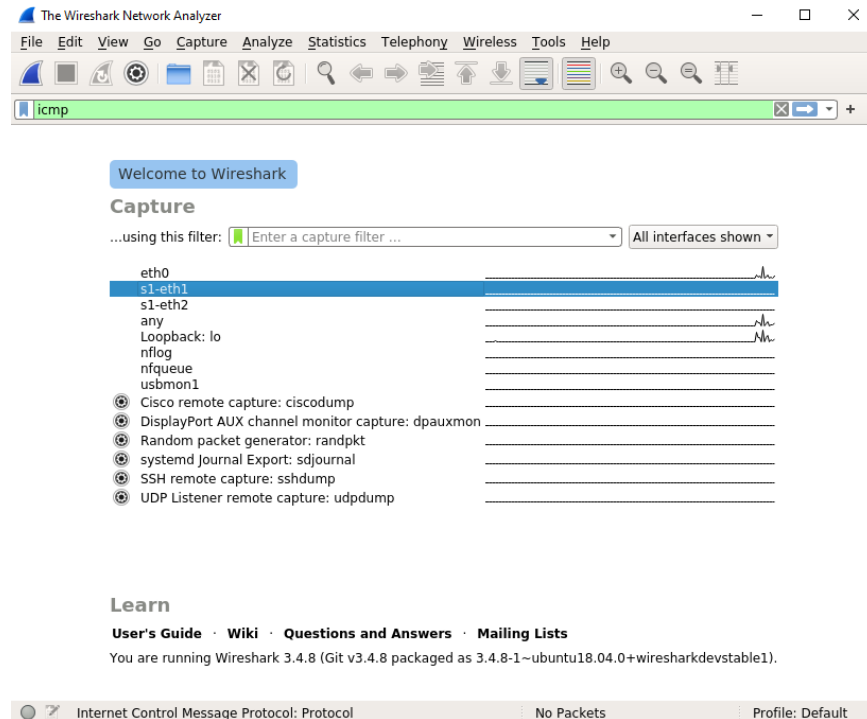


Figure 13: ws2

While listening in wireshark return to the **mininet command line interface** and write the command;

```
mininet> h1 ping -c 3 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.103 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.063 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.049 ms

--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2032ms
rtt min/avg/max/mdev = 0.049/0.071/0.103/0.024 ms
mininet> █
```

Figure 14: csp1

This will make **h1** ping **h2** 3 times. You can now examine **ICMP** packages being sent in wireshark and answer the questions for **PART 1**

3.1 Questions

1. What are the IP addresses of Source and Destination Hosts?
2. Why ICMP packets does not have source and destination port numbers?
3. Examine one of the ping request packets sent by your host. What are the ICMP type and code numbers? What other fields does this ICMP packet have? How many bytes are the checksum, sequence number and identifier fields?
4. Examine the corresponding ping reply packet. What are the ICMP type and code numbers? What other fields does this ICMP packet have? How many bytes are the checksum, sequence number and identifier fields?

IMPORTANT NOTE: While writing your answers in your report, also include screenshots of the packet fields you see in wireshark for questions **3** and **4**.

4 PART2 - Creating a Topology With Mininet

In the first part of the homework, we have worked on the minimal topology already created by **mininet** where there are two hosts, one switch and a controller. Now, for this part, you will create a new network topology by writing a python script **Mininet Python API**. You can find examples about how it is done in [this link](#). In the virtual machine, if you go to `~/mininet/examples`, you will see the example python scripts. You can look at them and see how it is done. You can also look at this [stackoverflow answer](#). Read this answer carefully, it will be a lot more easier if you understand the code written there. You can use that code in stackoverflow and enhance it for the topology we want. You have to run `sudo python3 topologycode.py` to run the topology.

Before starting this part, **traceroute** command is not found, install it with apt-get. We are going to need this command in this part;

`sudo apt-get install inetutils-traceroute` (3)

In this part, we want from you to create this topology;

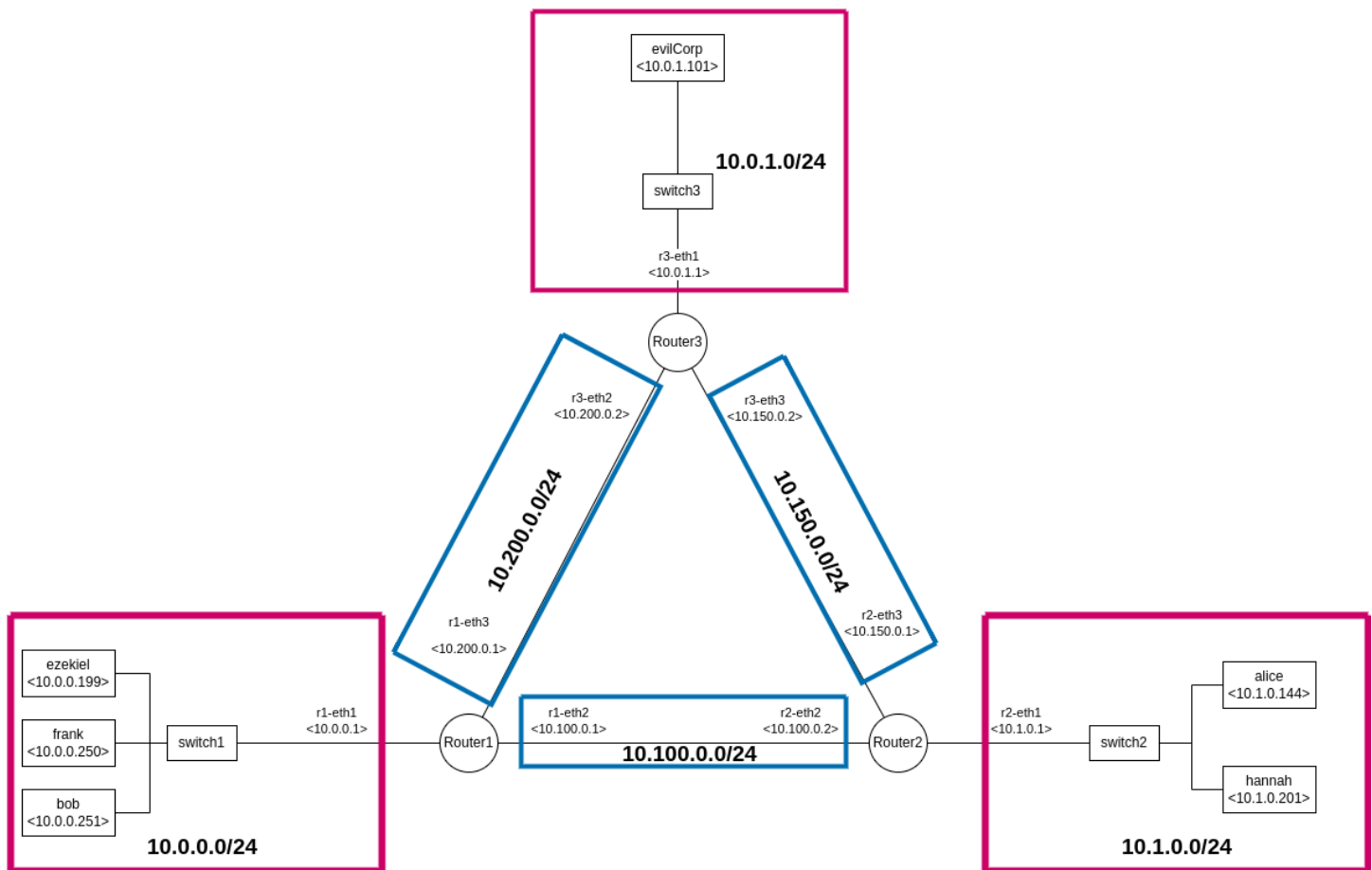


Figure 15: topology

In this topology, there are three switches, three routers and six hosts. You have to give the exact same ip addresses in the subnets. You should give exact same names to the hosts as you see in the diagram above. The filename of your script should be `e<your_student_id(7 digits)>.py`. About the routings of the routers;

- If hosts in subnet 10.0.0.0/24 are trying to reach 10.0.1.0/24, route should be source-Router1-Router3-dest
- If hosts in subnet 10.0.1.0/24 are trying to reach 10.0.0.0/24, route should be source-Router3-Router1-dest
- If hosts in subnet 10.0.0.0/24 are trying to reach 10.1.0.0/24, route should be source-Router1-Router2-dest
- If hosts in subnet 10.1.0.0/24 are trying to reach 10.0.0.0/24, route should be source-Router2-Router1-dest
- If hosts in subnet 10.0.1.0/24 are trying to reach 10.1.0.0/24, route should be source-Router3-Router2-dest
- If hosts in subnet 10.1.0.0/24 are trying to reach 10.0.1.0/24, route should be source-Router2-Router3-dest

sudo python3 topology.py (4)

The script should create the topology and open **mininet command line interface**, just like we did in the first part with minimal topology. Your script has to print informations like this (Python API of mininet does this. Check out the examples);

```
*** Creating network
*** Adding controller
*** Adding hosts:
alice bob evilCorp ezeziel frank hannah r1 r2 r3
*** Adding switches:
s1 s2 s3
*** Adding links:
(alice, s2) (bob, s1) (evilCorp, s3) (ezeziel, s1) (frank, s1) (hannah, s2) (r1,
r2) (r1, r3) (r2, r3) (s1, r1) (s2, r2) (s3, r3)
*** Configuring hosts
alice bob evilCorp ezeziel frank hannah r1 r2 r3
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...
*** Starting CLI:
mininet> |
```

Figure 16: topology

After the **mininet command line interface** is opened, you should answer these questions by writing commands. These commands will test whether routes are correct and all hosts are reachable.

4.1 Questions

After you have created the topology and run the script, answer these questions;

1. Write the command **pingall** to check whether all hosts are reachable from one to another and share the screenshot of terminal.
2. Write the command **ezeziel traceroute hannah** and share the screenshot of terminal.
3. Write the command **alice traceroute bob** and share the screenshot of terminal.
4. Write the command **frank traceroute evilCorp** and share the screenshot of terminal.
5. Write the command **evilCorp traceroute frank** and share the screenshot of terminal.
6. Write the command **evilCorp traceroute alice** and share the screenshot of terminal.
7. Write the command **hannah traceroute evilCorp** and share the screenshot of terminal.

5 Other Specifications

- Feel free to ask questions through ODTUClass discussions or send me a mail on erend@metu.edu.tr.
- See the course syllabus for the late submission policy.
- This is an individual assignment. Using any piece of code, discussion, explanation etc. that is not your own is strictly forbidden and constitutes as cheating. This includes friends, previous homeworks, or the Internet. The violators will be punished according to the department regulations.

6 Submission

- Upload your assignment report in pdf format to the ODTUClass *THE3 Report Submission*. Your report should contain two headings for each part and answers to questions in both parts. Name of your pdf report should be `e<your_student_id(7 digits)>.pdf`.
- Submit your `.py` file through *Topology Code Submission* on ODTUClass. The name of the `.py` file should be `e<your_student_id(7 digits)>.py`

6.1 Grading

- Please ensure that the screenshots you have included in your report are legible.
- Answers without explanations or screenshots to support them (e.g. answering just “5” to a “How many...” question) will get no grade. If only screenshots are wanted however, answer with only a screenshot.