**IEO REPORT**

Week 7 – Week 10



Student Name : **Shanessa Kostaman**

Student Number : **4082419**

Class : **P-CB06**

**Fontys University of Applied Sciences**

**Eindhoven – Netherland**

**2020-2021**

Contents of The Report

[Revision Table 3](#_Toc55073764)

[Introduction 4](#_Toc55073765)

[WEEK 7 - NetworkBasics 5](#_Toc55073766)

[WEEK 8 - IP 10](#_Toc55073767)

# Revision Table

|  |  |  |  |
| --- | --- | --- | --- |
| **WEEK** | **Name of The File** | **Date** | **Time** |
| Week 7 | Report Week 7 – Week 10 | Saturday,24/10/2020 | 15 : 20 |
| Week 8 | Report Week 7 – Week 10 | Monday, 01/11/2020 | 16 : 37 |
| Week 9 |  |  |  |
| Week 10 |  |  |  |

# Introduction

My name Shanessa Kostaman and my student number is 4082419. I am the first-year student of Fontys University of Applied Sciences and I am taking ICT for my major. I have plan to taking Software & Engineering as my profile, but I have not decided yet.

I am choosing ICT as my major because I saw a great opportunity to find jobs. I also did a small research about ICT and they said ICT has the most job vacancies nowadays. So it’s make me easier to find jobs on the future.

I do not have any knowledge about ICT and I never learn ICT on My Junior or Senior Highschool back then. But I am so sure with my capability. If I am willing to learn I am sure I can do it. The only thing that I have only passion. ICT is not a simple thing that we can learn for a day, we need to practice everyday to learn ICT, that’s why we need passion.

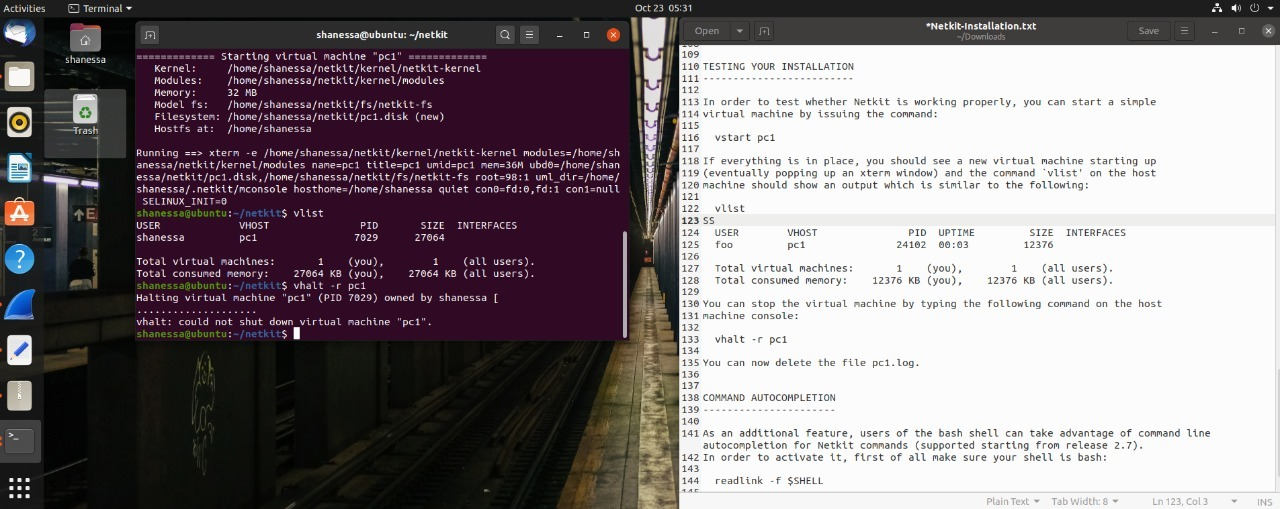
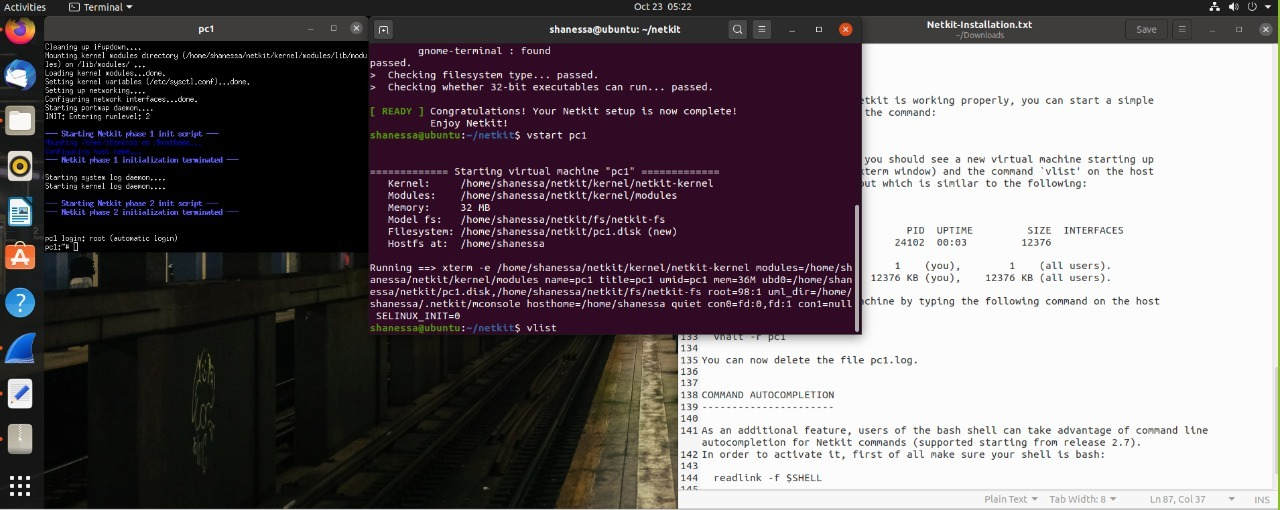
That’s my Introduction, Thank You.

# WEEK 7 - NetworkBasics

Task 1: install and Test Netkit Tool

Consult this week’s theory presentation and use the Netkit commands to start and halt a network node as described in the presentation. Netkity and Wireshark are already installed in the preconfigured Linux. If you installed the Linux yourself, then you need to install these tools yourself. (there is a guideline in the Canvas)

Describe the steps you took and provide screenshot of the started node.



Task 2: TCP/IP Layers in Wireshark

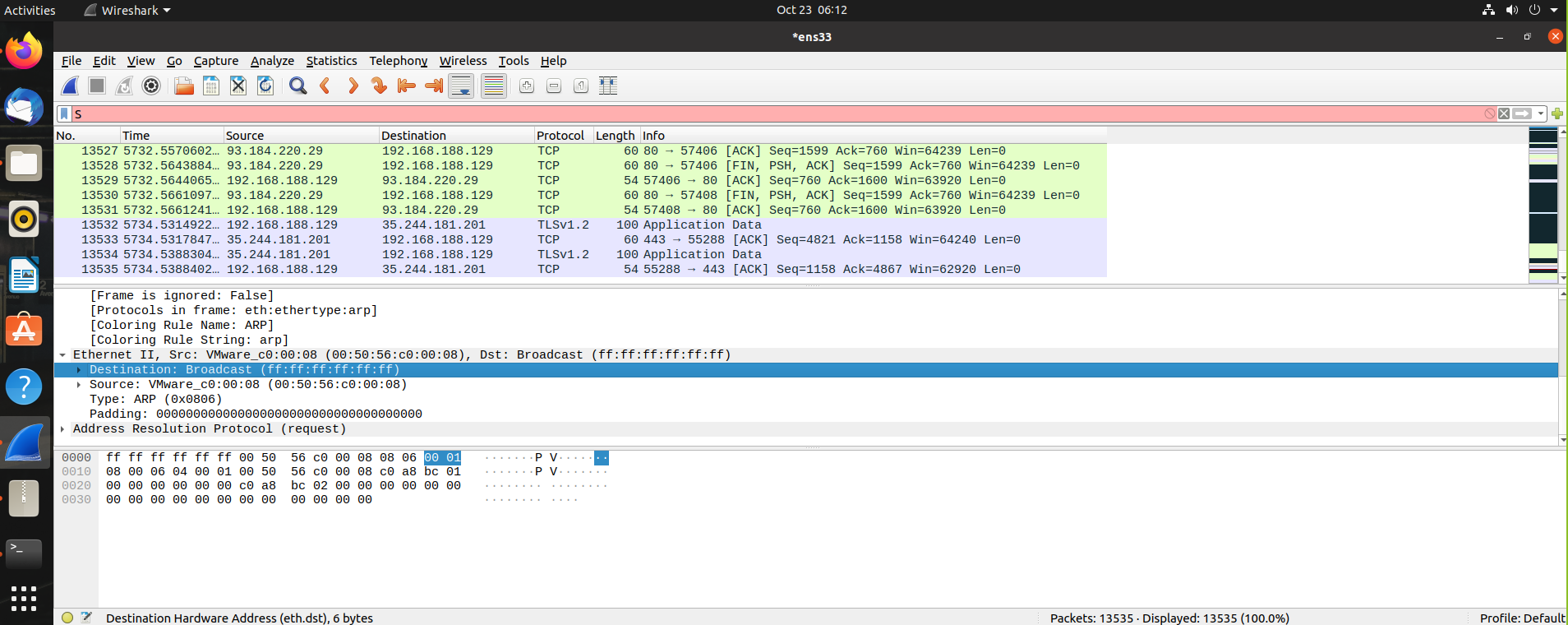
Find a Wireshark Tutorial on the web. Run Wireshark.

Start capturing the network traffic. To generate HTTP traffic, go to <http://courses.codemax.net/w2.html> web browser. Don’t forget to stop capturing as you can get a lot of traffic in your capture. Look at your captured packets and find an HTTP GET packet and Answer the following questions and provide the screenshots:

* What is the source and destination MAC address of this HTTP packet?

Source MAC address: **00:50:56:c0:00:08**Destination MAC address: **ff:ff:ff:ff:ff:ff**

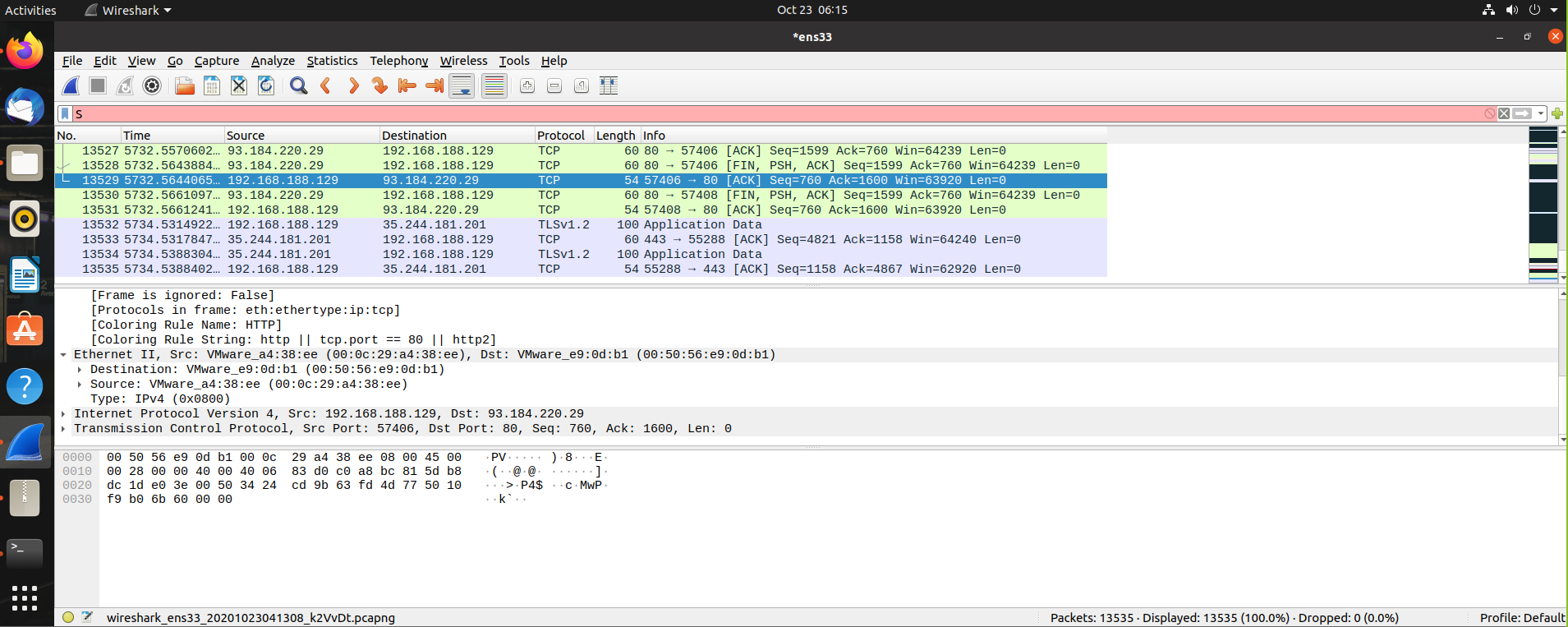
Provide a screenshot below with the Wireshark snapshot and highlight these addresses:



* What is the source and destination IP address of this HTTP packet?

Source IP address : **192.168.188.129**  
Destination IP address : **93.184.220.29**

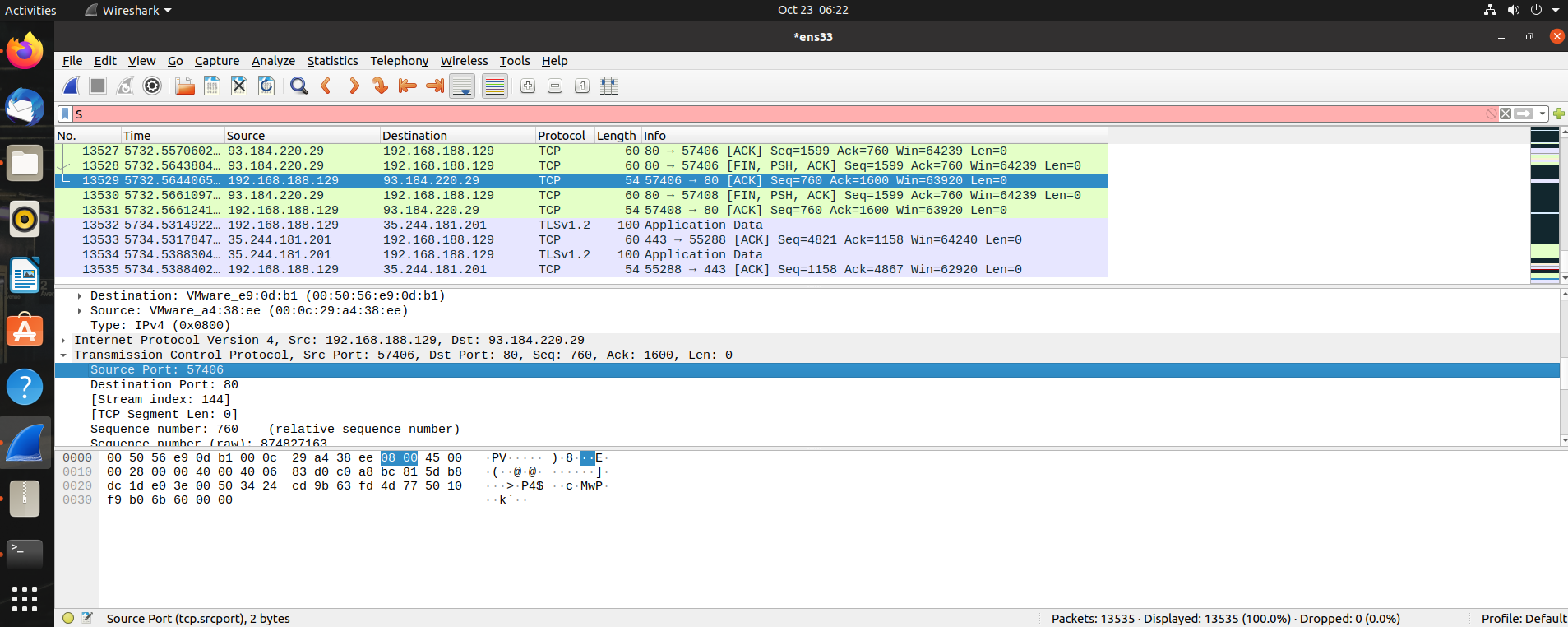
Provide a screenshot below with the Wireshark snapshot and highlight these addresses:



* What is the source and destination port of this HTTP packet? Provide a screenshot to prove it

Source port : **57406**  
Destination port: **80**

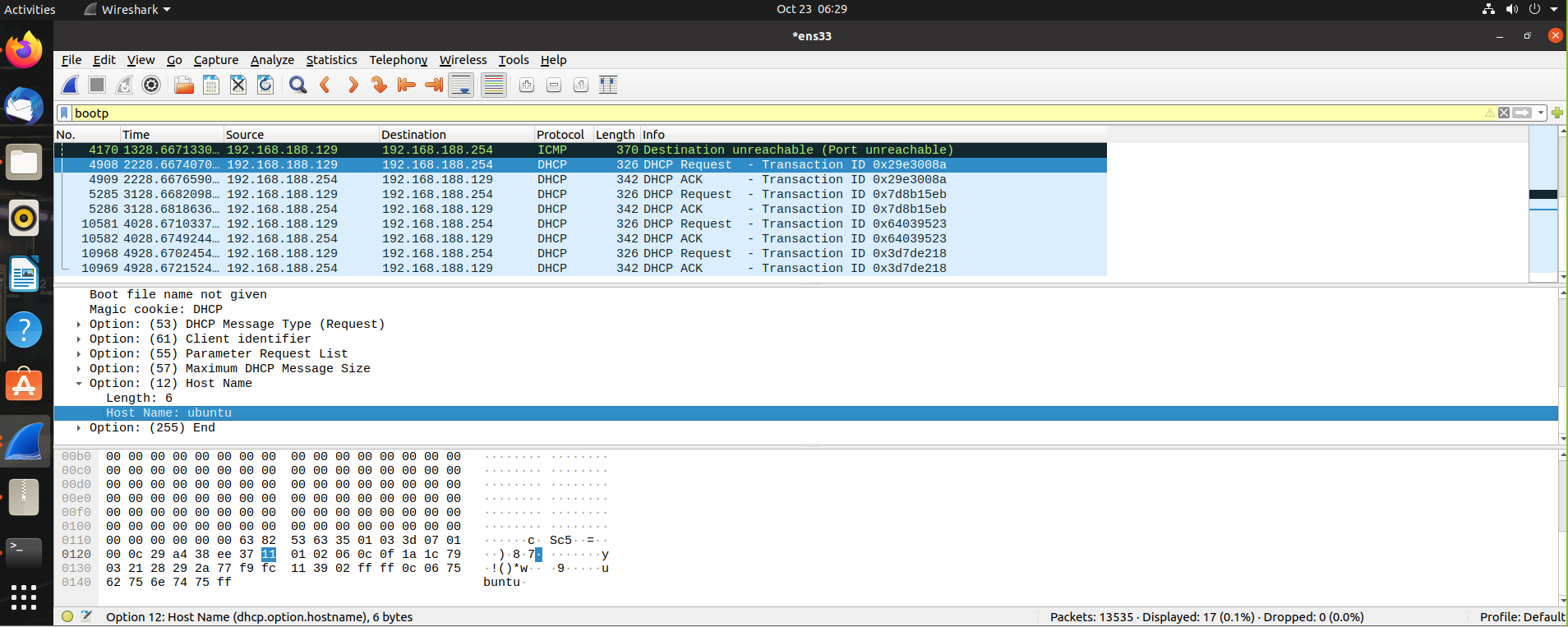
Provide a screenshot below with the Wireshark snapshot and highlight these addresses:



* What is the host name of this HTTP Get packet?

Host name: **ubuntu**

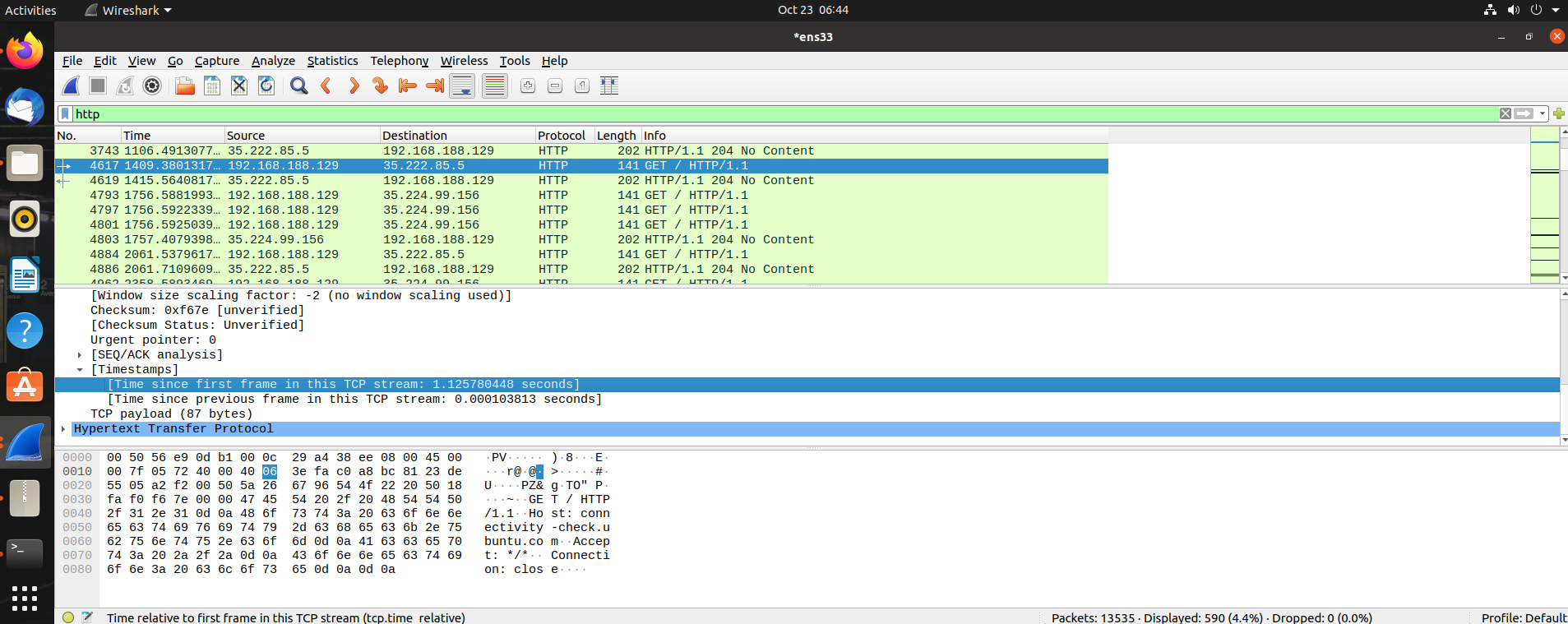
Provide a screenshot below with the Wireshark snapshot and highlight the host name:



* Find the HTTP Response belonging to the HTTP Get packet. How much time elapsed between the HTTP Get and HTTP response?

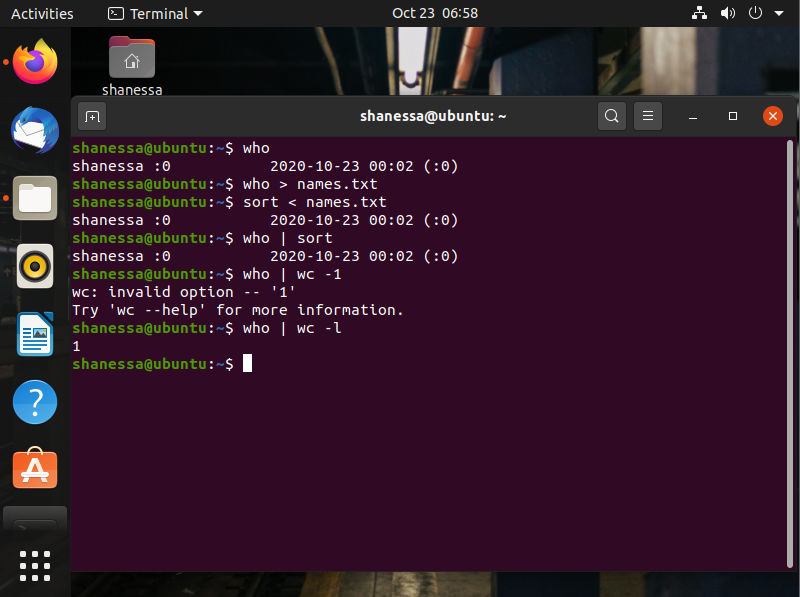
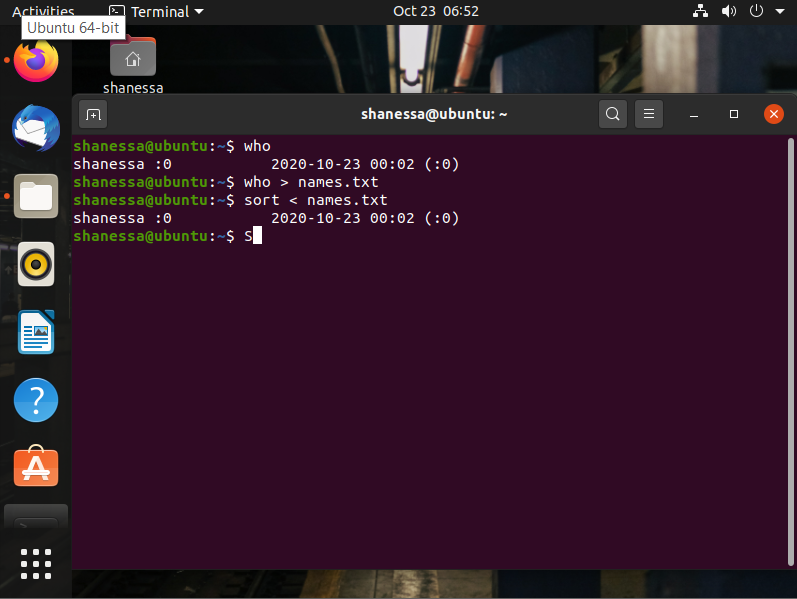
Time elapsed: **1.125780448 - 0.000103813 =** **1.125676635**

Provide a screenshot below with the Wireshark snapshot and highlight the elapsed time:



Task 3: Do Linux Tutorial

Go to <http://www.ee.surrey.ac.uk/Teaching/Unix/index.html> and do the tutorial three.

Provide screenshots of all exercises in section 3.4**

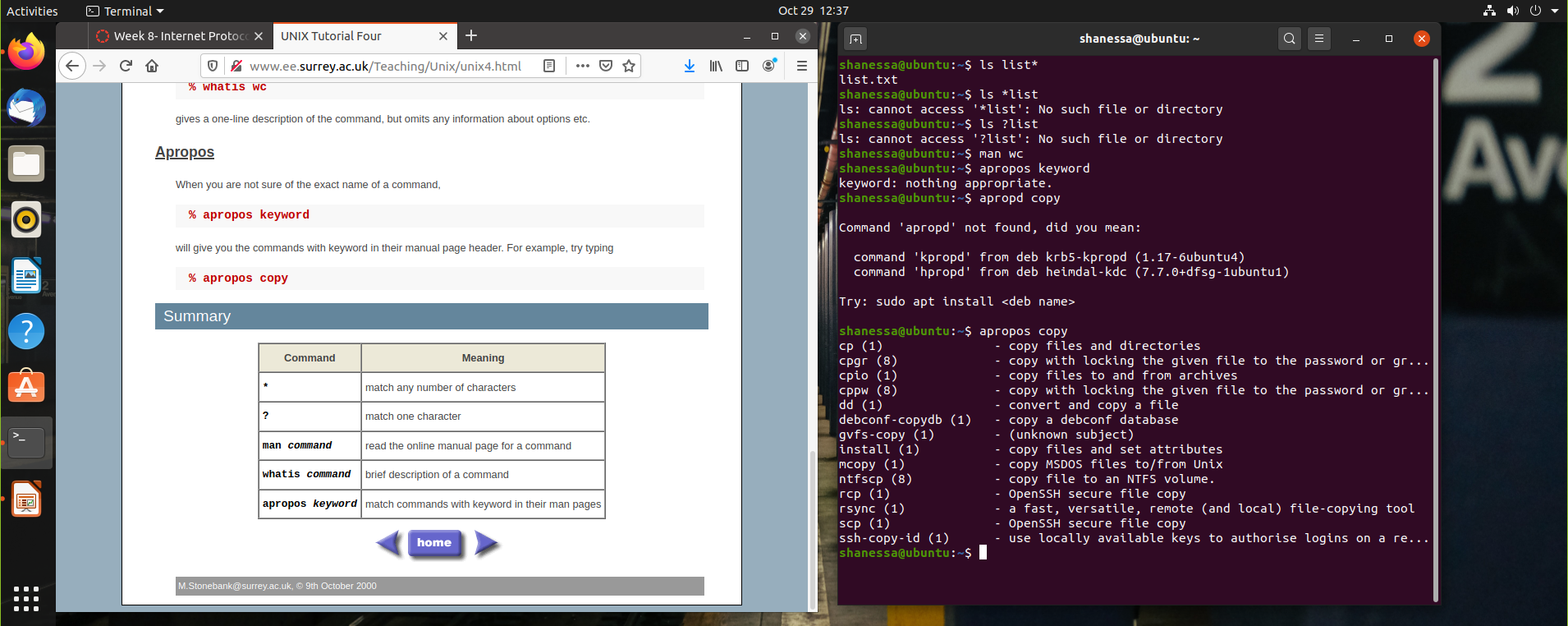
# WEEK 8 - IP

Linux, Static IP address/subnets configuration

**Task 1a**: Do Linux Tutorial

Go to [http://www.ee.surrey.ac.uk/Tea ching/Unix/unix2.html](http://www.ee.surrey.ac.uk/Tea%20ching/Unix/unix2.html) and do the 2nd basic Unix tutorial.

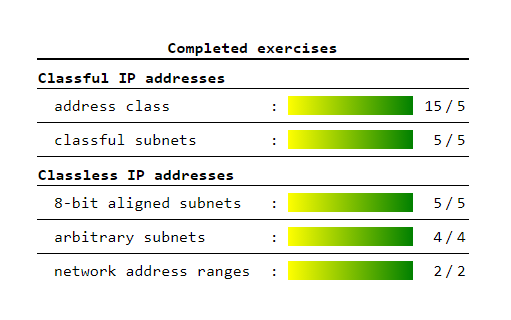
Provide screenshots of all exercises 2a and 2b. Do all subsections of this tutorial – all of them are really useful! This task should be done individually, so each member of the team should provide his/her evidence(screenshots).



**Task 1b**: Networking exercise

Do the netwoerking online exercises via this link <https://courses.codemax.net/w8.html>.

Provide screenshots of all exercises.



**Task 2:** Build A Simple Netkit Network

Read the explanation of the basic Netkit commands and use them to build a simple network of two nodes connected to a LAN interface.

Try the following configurations:

A) Configure the IP addresses of the 2 nodes by using the “ifconfig” command explained in the theory lesson.

1. Node1 has an IP address 102.10.2.1/24
2. Node2 has an IP address 102.20.2.1/24

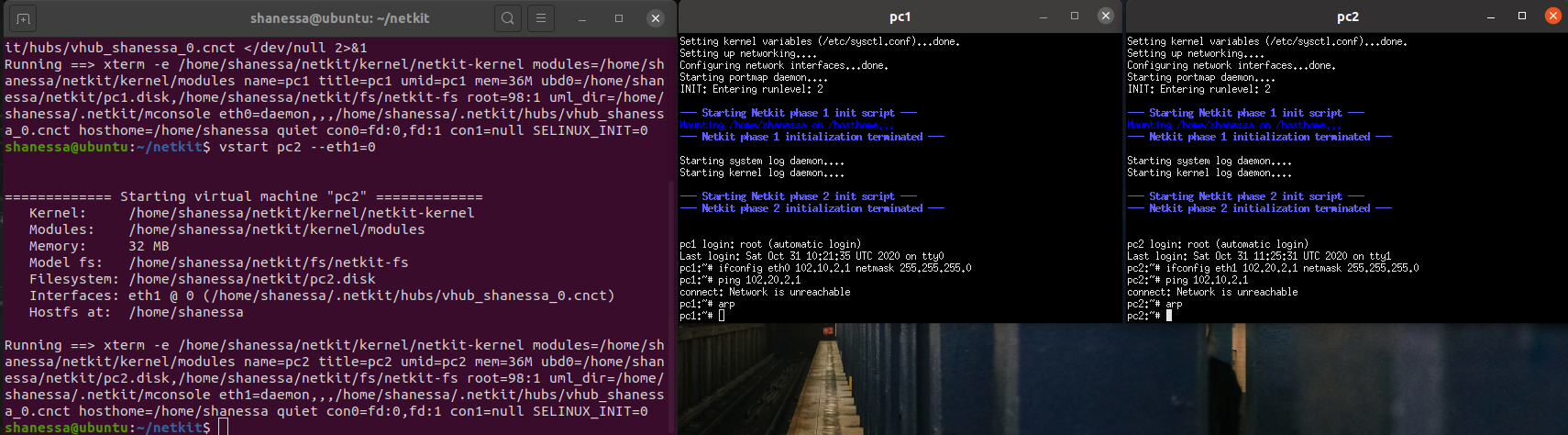
Check whether your configuration was successful by using ping command between these two nodes.

1. What is the result of the ping? Can you explain it? Provide a screenshot.

* The result of the ping is “Network is unreachable” because the subnet / the network is different, so the PC can not connect to each other.

1. Look at the ARP entries of your Node1 and Node2. Which command do you use? Which ARP entries are there?

* The command that I used is ARP
* The list on the ARP is empty because the PC are not connected to each other



B) Configure the IP addresses of the 2 nodes by using the “ip” command explained in the theory lesson.

1. Node1 has an IP address 102.10.2.1/10
2. Node2 has an IP address 102.20.2.1/10

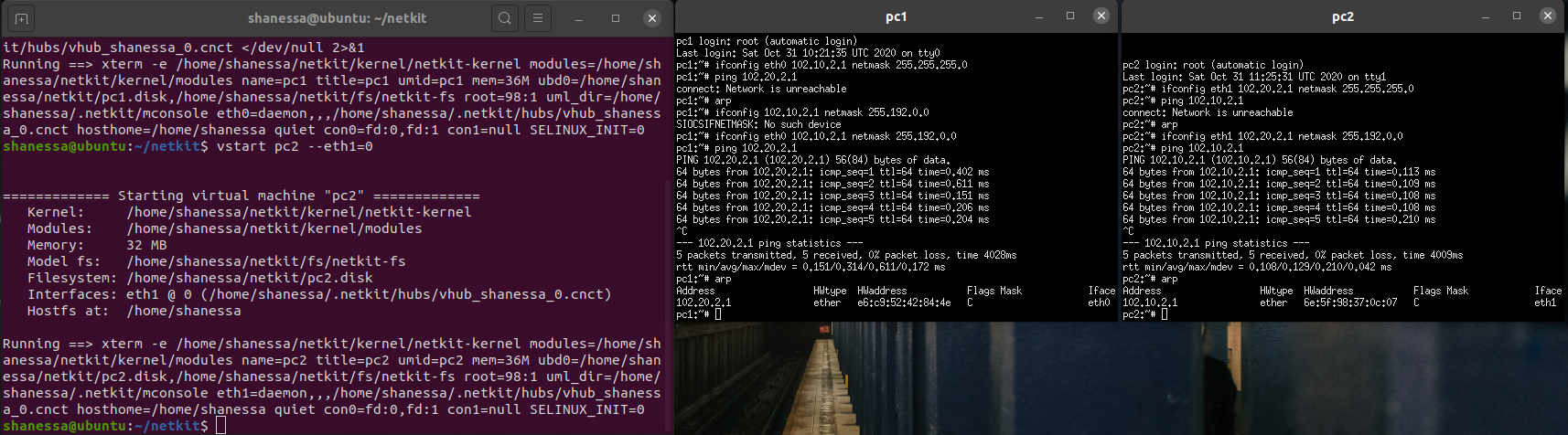
Check whether your configuration was successful by using ping command between these two nodes.

1. What is the result of the ping? Can you explain it? Provide a screenshot of your configured interfaces.

* The result of the ping is succeeded, it’s working. The PC are connected to each other.

1. Look at the ARP entries of your Node1 and Node2. Which ARP entries are there?

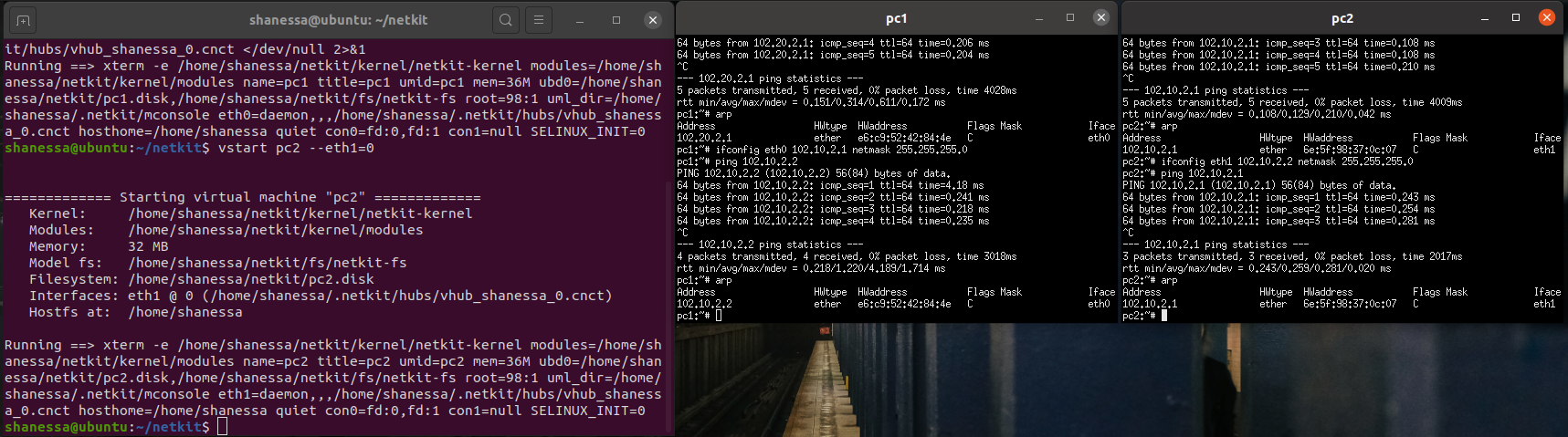
* The ARP of PC 2 already on the PC 1 and vice versa.



C) Configure both nodes to have a subnet mask 255.255.255.0, and change the IP address of Node2 in such a way that the ping between them is successful.

1. Provide a screenshot of your configuration and successful ping.
2. After successful ping ARP entries of both nodes should be changed. Provide a screenshot of the new ARP situation and explain it. What is the command to clear the ARP cache again

* I changed the IP address of PC 2 and the ARP list showed us the new IP address of PC 2 on the PC 1. And when we want to clear the ARP cache we can use command “arp -d and put the IP address”, e.g. arp -d 102.10.2.2



**Task 3**: Configuring Network

For this assignment you can use a preconfigured netkit lab provided in net\_routing.zip file. To do this you need to copy the provided zip file somewhere in your Linux environment, e.g. in ~/netkit\_labs. Unzip the file. You have now a preconfigured lab Deliver the lab network of this task in your *git* project*. Thus, when you are done write below the URL of your git project (I should be able to access your results using “git clone” and the provided git URL).*

Each simulated node has its own directory. Also, each simulated node has a <node>.startup file where any commands can be added that should be executed before startup of the node.

To start the lab issue the following command in the root directory of your lab:

lstart

Note: When you issue this command, you’ll be prompted for a password which in your case is **student**.  
  
Netkit uses the file “labs.conf” in order to initialize the Ethernet devices and their respective collision domains for each node. For example inside the labs.conf there is a line “RouterAC[0]=LANA” and a line “RouterAC[1]=LANC”.   
These two lines have same effect when the node “RouterAC” is initialized, as if we would run the command:  
“*vstart RouterAC* --*eth0=LANA --eth1=LANC*”.

Now all the nodes should be started. However, the nodes are not configured yet. You need to configure them as follows:  
Configure the Ethernet devices connected via the collision domain LANA using the IP range 10.X.0.0/16, where X is the number of your pair/group.

Configure the Ethernet devices connected via the collision domain LANB using the IP range 172.16.X.0/24, where X is the number of your pair/group.

Configure the Ethernet devices connected via the collision domain LANC using the IP range 192.168.X.0/24, where X is the number of your pair/group.

For example if your group number is 230 you should use IP address from the range 10.230.0.0/16 for LANA, 172.16.230.0/24 for LANB and 192.168.230.0/24. (see also table 1).

There are 2 ways to configure your interfaces*. We recommend you all use the first option and for your own experiment you can use the second option but make sure all your submissions follow the first option:*

1. Use either ifconfig or ip commands. Once you know how the commands should look like, it is highly recommended to put them in <node>.startup files, so next time you want to restart and present your lab, you don’t have to reconfigure it by hand again.

Note : Please don’t remove the commands which are already present in the <node>.startup files. They are necessary for starting up Linux networking service.

1. Use <node>/etc/network/interfaces file of the node you want to configure.

In the netkit lab environment you can put any files the contents of which you want to see in the simulated node in the <node> directory. In this way, you can also put there <node>/etc/network/interfaces file. This file is used by Linux system to configure the network interfaces. An example of such a file is provided in the lab for PC1A node.

The network of the lab is as follows:

1. PC1A, PC2A and RouterAC are connected to LANA
2. PC1B, PC2B and RouterBC are connected to LANB
3. PC1C, PC2C, RouterBC, RouterAC and Gateway are connected to LANC
4. Gateway is connected to LANC through fixed eth0 interface with IP address 192.168.1.x/24 and to TAP\_LAN through eth1 interface with IP address 192.168.200.1. The TAP\_LAN is a Netkit-specific interface used for the connection to your guest Linux system. The Gateway node will be used for the optional part of the Assignment 3.
5. Your guest Linux system is connected to your simulated Netkit node Gateway through Netkit specific tap interface nk\_tap\_student 192.168.200.254, see the detail of the connection between the Netkit simulated environment and your Guest machine in the picture below.

**

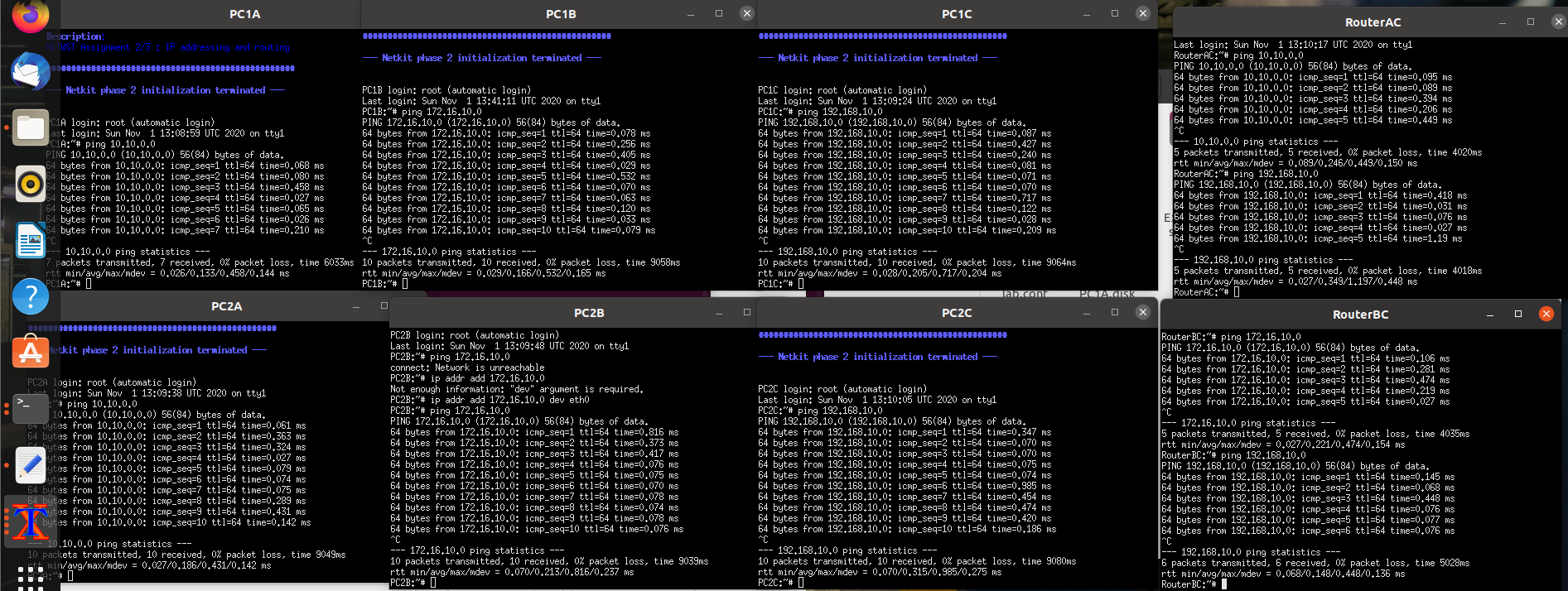
***Provide the network drawing*** *of your lab network you can use* <https://app.diagrams.net/> *and screenshots of the pings which are possible WITHIN LANA, LANB and LANC (PC1A to PC2A, PC1B to PC2B and so on). When creating* the *network drawing, don’t forget to mention* the *IP addresses/subnet masks for all nodes of your network. It is also useful to include the names of the network interfaces (eth0, eth1, …).*

You don’t need to be able to route between all nodes of this network; that is the second part of the assignment, which will be done next week.

Note 1: In the provided netkit lab there are files HOWTO, interfaces.example and Example.startup which can give you more info on how to use and configure the lab.

Table 1 : IPv4 address ranges per student group

|  |  |  |  |
| --- | --- | --- | --- |
| Group | LANA | LANB | LANC |
| 1 | 10.1.0.0/16 | 172.16.1.0/24 | 192.168.1.0/24 |
| 2 | 10.2.0.0/16 | 172.16.2.0/24 | 192.168.2.0/24 |
| … | | | |
| n | 10.n.0.0/16 | 172.16.n.0/24 | 192.168.n.0/24 |



Diagram

Description automatically generated

**Task 4**: CIDR IP Addressing Exercises

1. Suppose we have IP address 122.33.196.145/24

Fill in the following items for this address:

1. Network Address : **122.33.196.0**
2. Broadcast Address : **122.33.196.255**
3. Subnet Mask : **255.255.255.0**

2. Suppose we have IP address 163.249.223.229/25

Fill in the following items for this address:

1. Network Address : **163.249.223.128**
2. First Host : **163.249.233.129**
3. Last Host : **163.249.223.254**
4. Broadcast Address : **163.249.233.255**