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

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	Report Week 7 – Week 10	Sunday, 08/11/2020	22 : 16
Week 10	Report Week 7 – Week 10	Sunday, 15/11/20	21 : 41
Week 11	Final Report Week 7 – Week 10	Sunday, 22/11/2020	10 : 50

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.

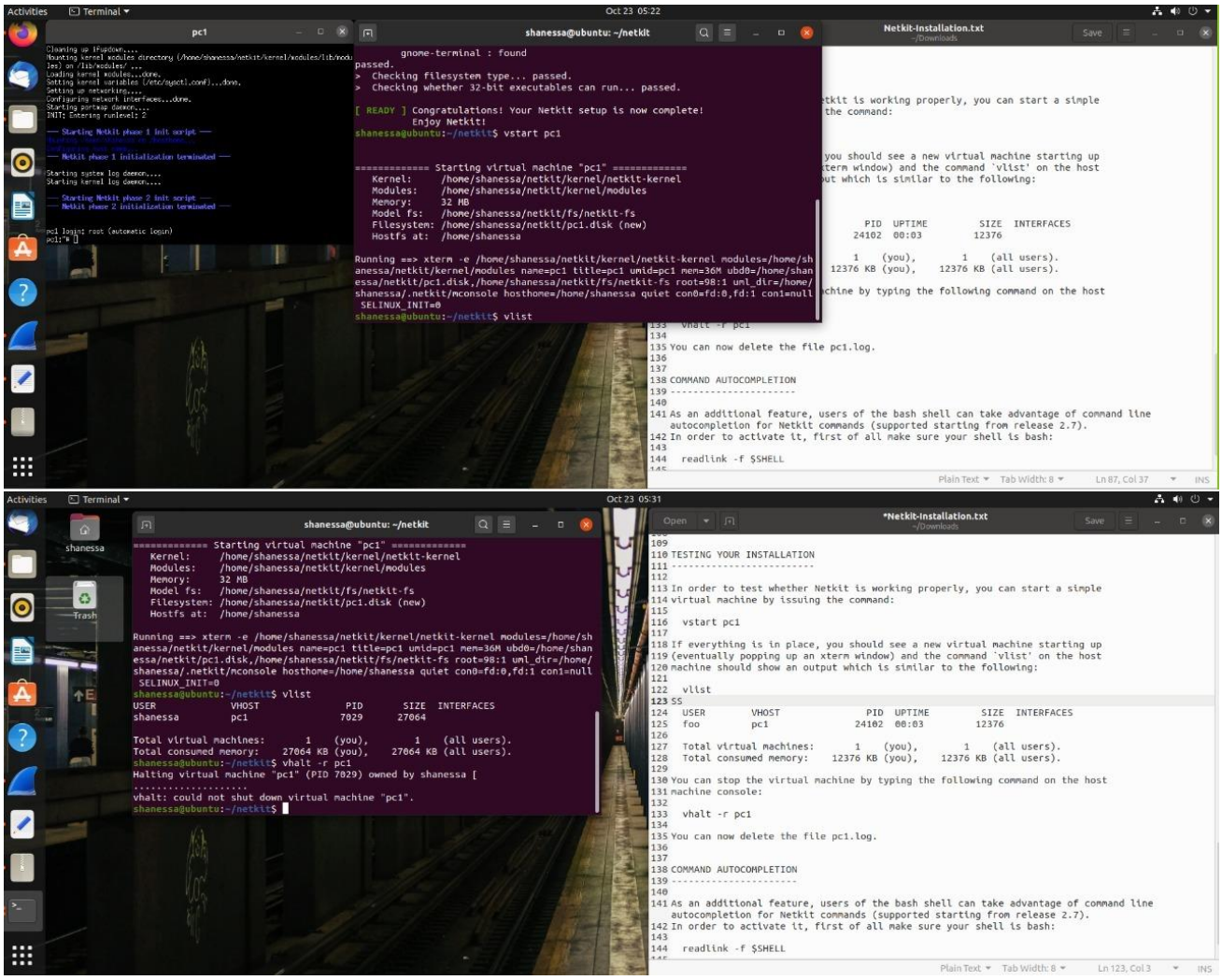
That's my Introduction, Thank You.

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.

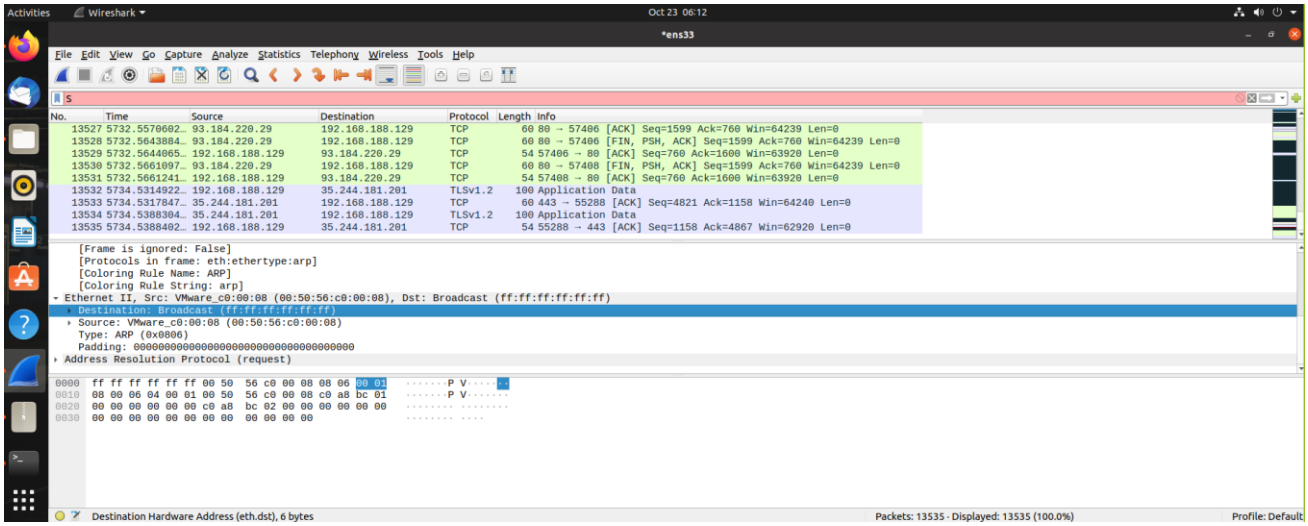


Task 2: TCP/IP Layers in Wireshark

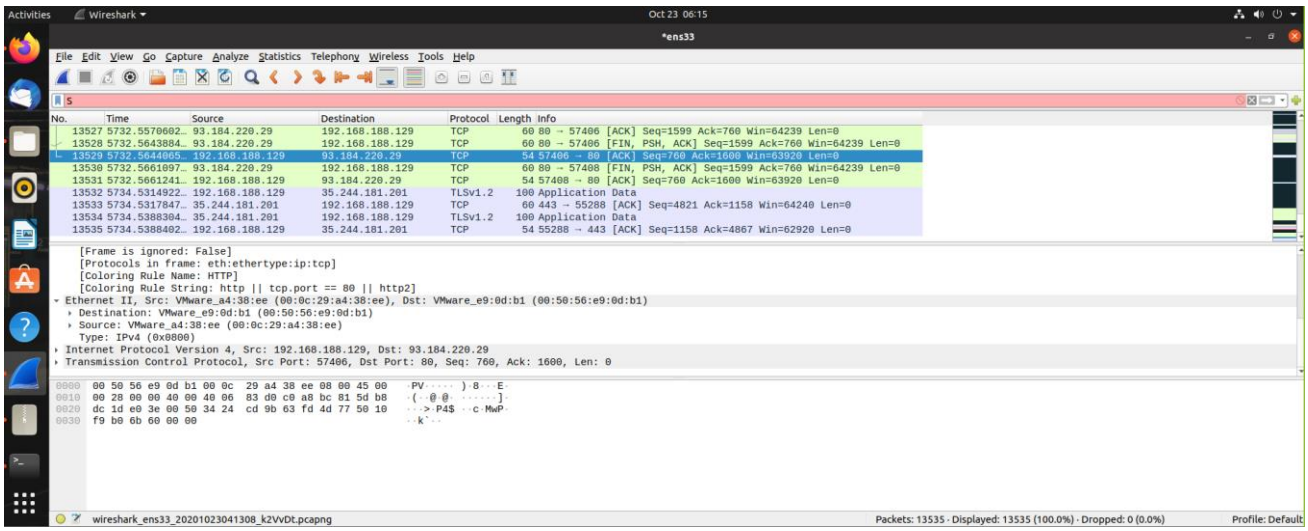
Find a Wireshark Tutorial on the web. Run Wireshark.

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:

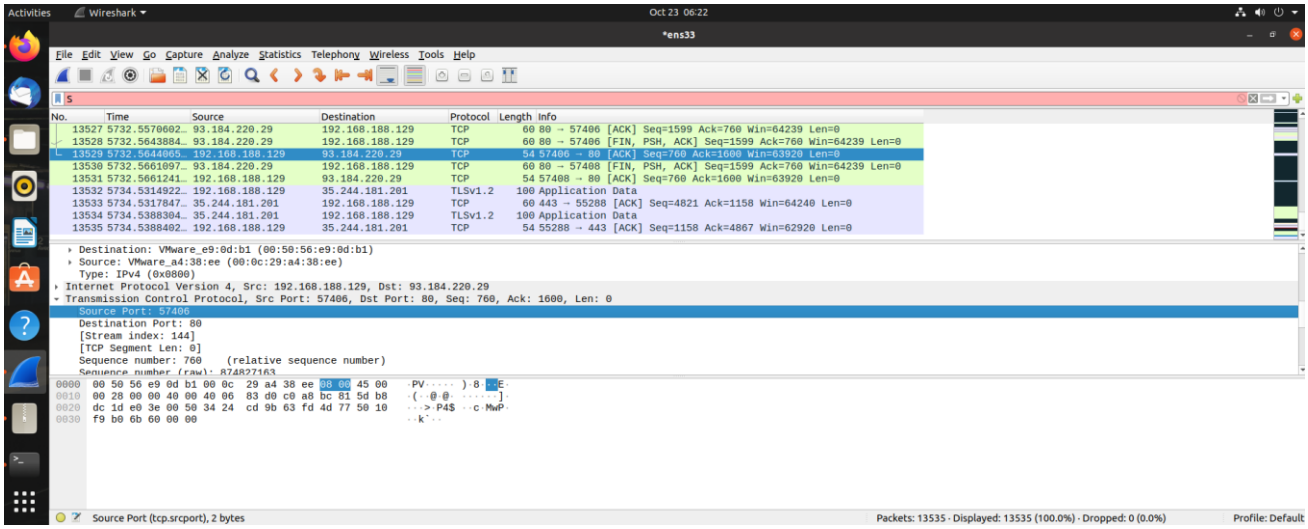
- *What is the source and destination MAC address of this HTTP packet?*
  Source MAC address: ***00:50:56:c0:00:08***
  Destination MAC address: ***ff:ff:ff:ff:ff:ff***
  Provide a screenshot below with the Wireshark snapshot and highlight these addresses:



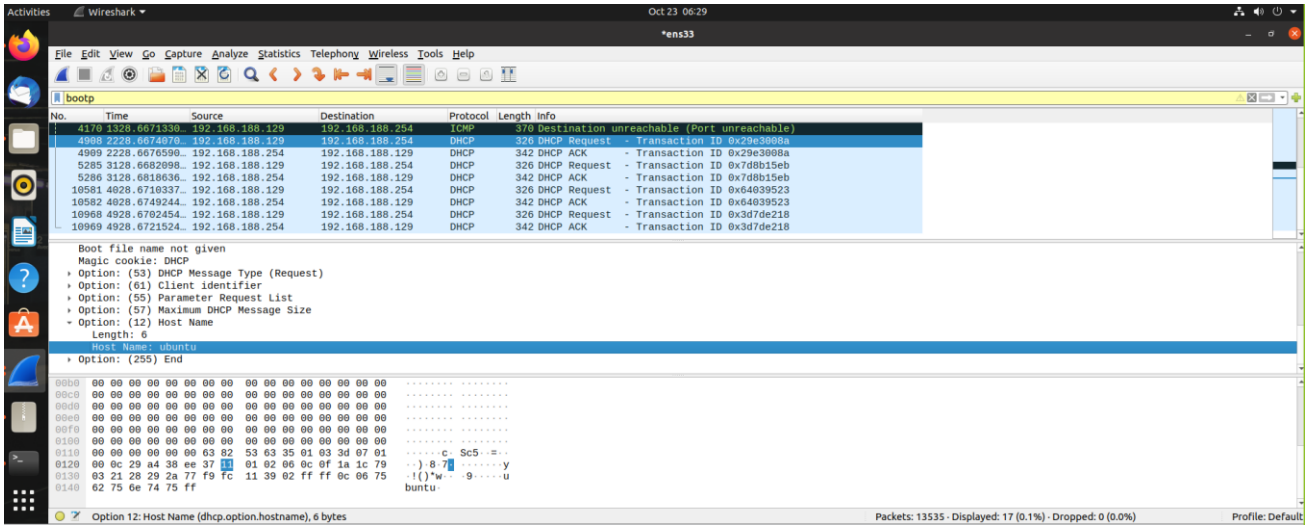
- *What is the source and destination IP address of this HTTP packet?*
  Source IP address : ***192.168.188.129***
  Destination IP address : ***93.184.220.29***
  Provide a screenshot below with the Wireshark snapshot and highlight these addresses:



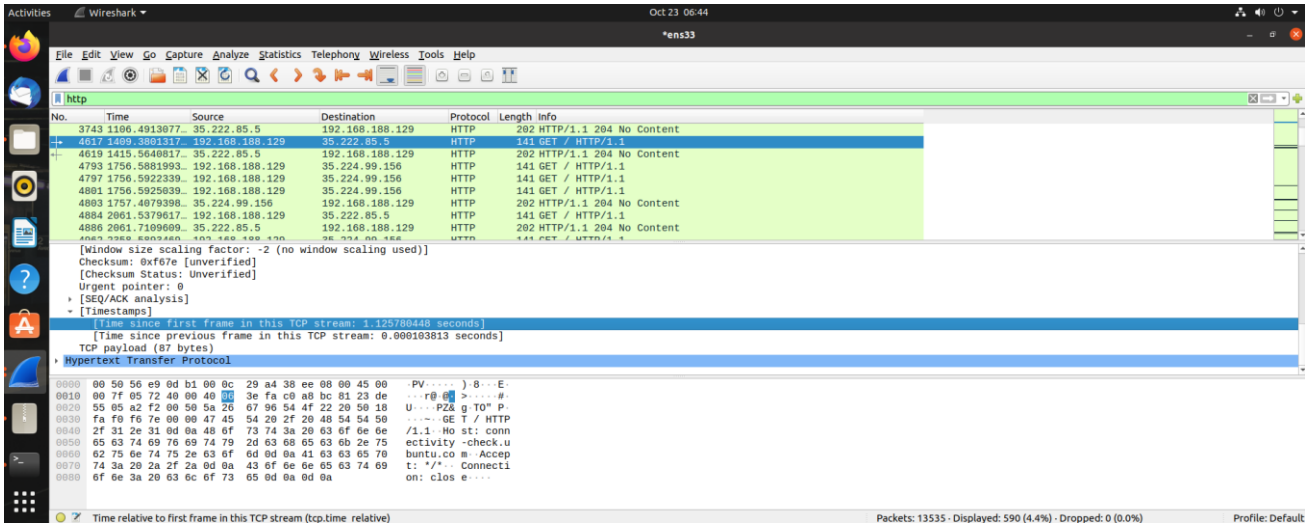
- *What is the source and destination port of this HTTP packet? Provide a screenshot to prove it*
  *Source port : **57406***
  *Destination port: **80***
  Provide a screenshot below with the Wireshark snapshot and highlight these addresses:



- *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:



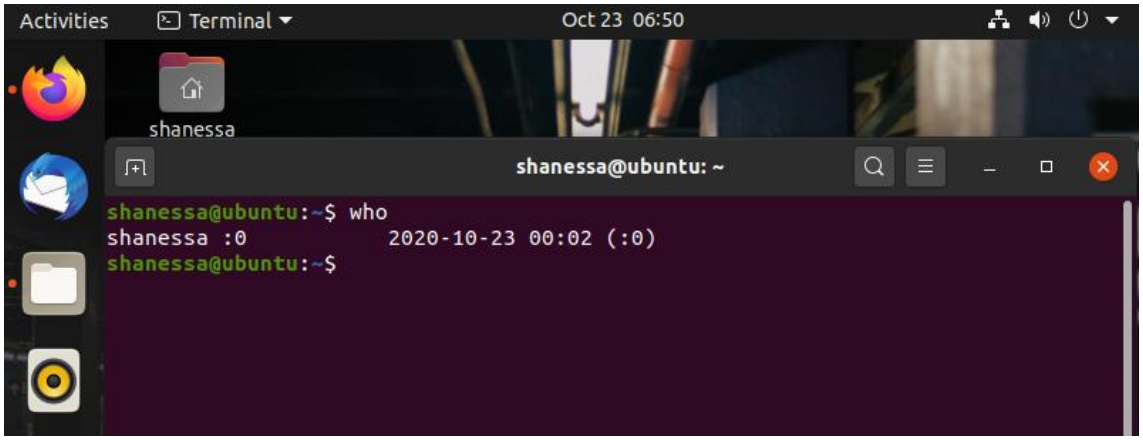
- *Find the HTTP Response belonging to the HTTP Get packet. How much time elapsed between the HTTP Get and HTTP response?*
  Time elapsed: ***1.125780448 - 0.000103813 = 1.125676635***
  Provide a screenshot below with the Wireshark snapshot and highlight the elapsed time:

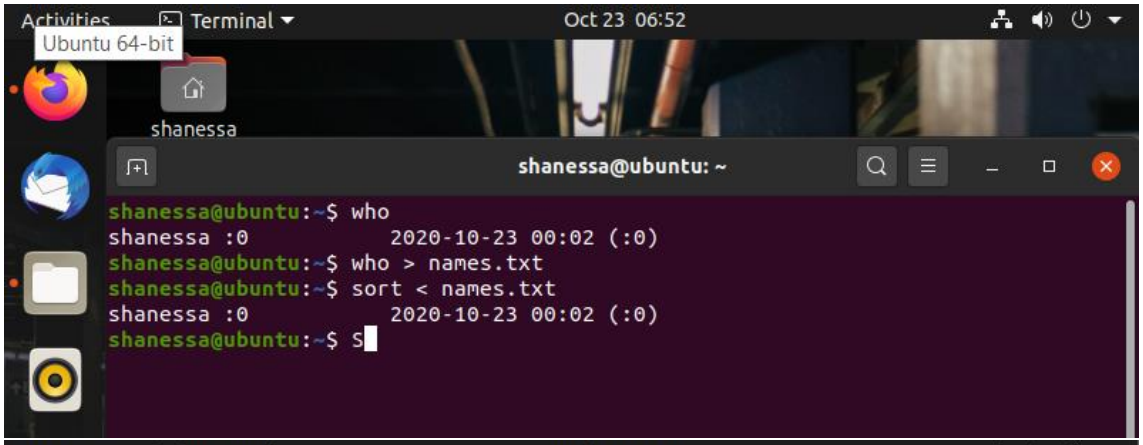


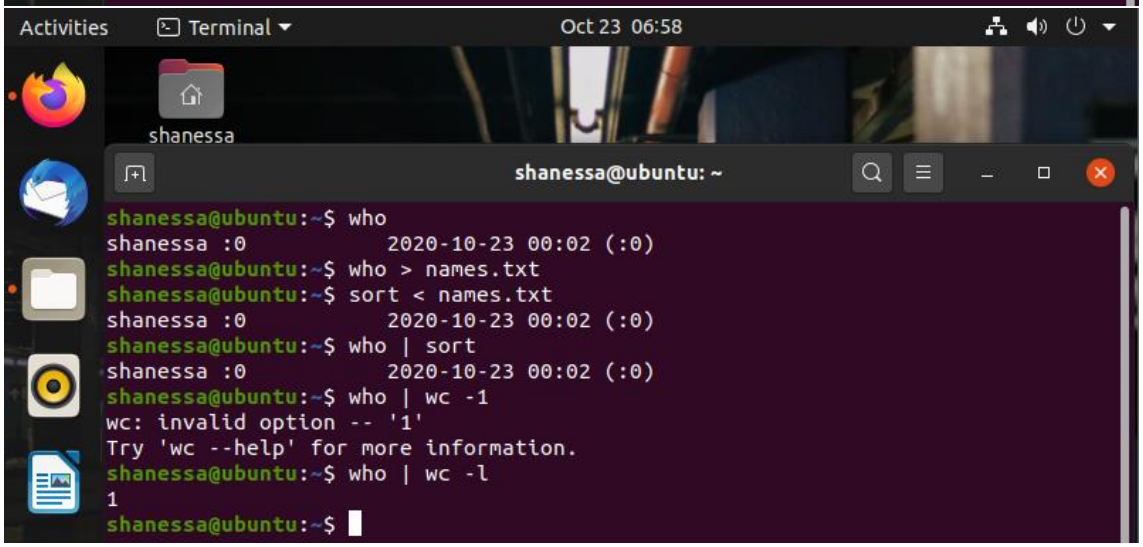
Task 3: Do Linux Tutorial

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







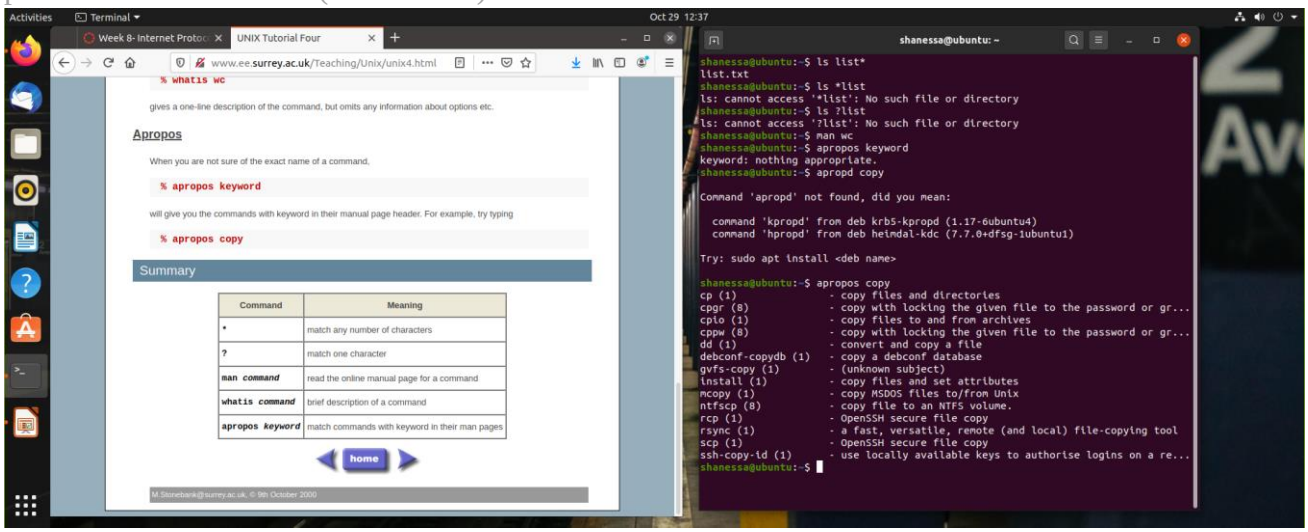
WEEK 8 - IP

LINUX, STATIC IP ADDRESS/SUBNETS CONFIGURATION

***Task 1a**: Do Linux Tutorial*

Go to http://www.ee.surrey.ac.uk/Tea ching/Unix/unix2.html and do the 2nd basic Unix tutorial.

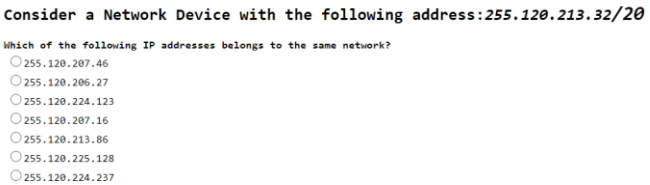
Provide screenshots of all exercises 2a and 2b. Do all subsections of this tutorial – all of them are really useful! This task should be done individually, so each member of the team should provide his/her evidence(screenshots).

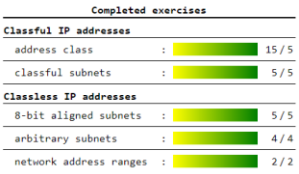


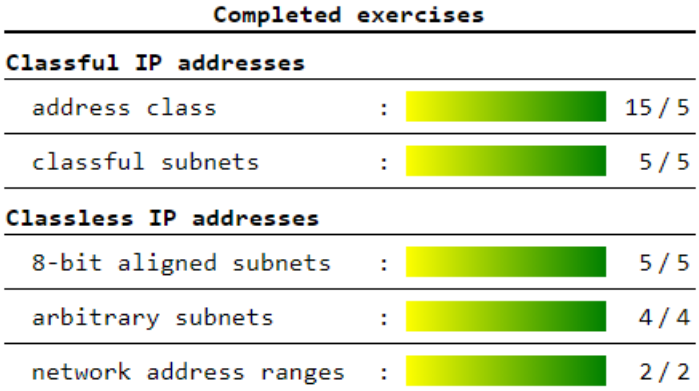
***Task 1b**: Networking exercise*

Do the netwoerking online exercises via this link [https://courses.codemax.net/w8.html.](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*

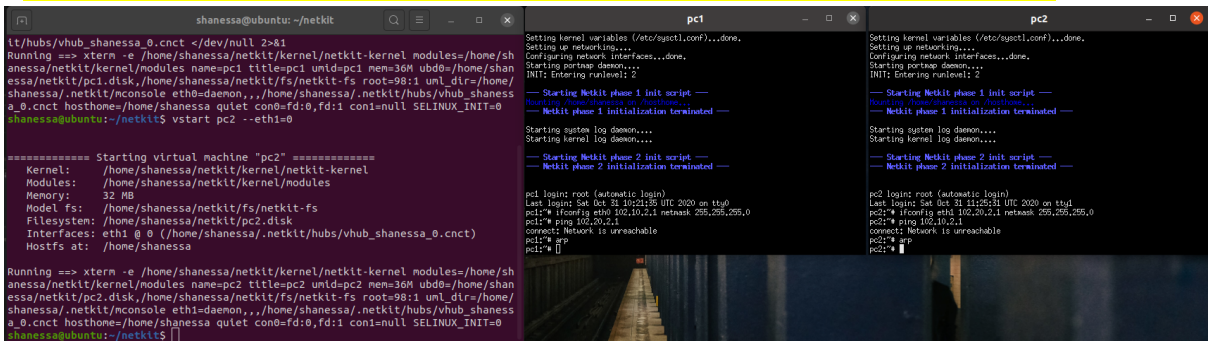
Check whether your configuration was successful by using ping command between these two nodes.

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.*

2. *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*

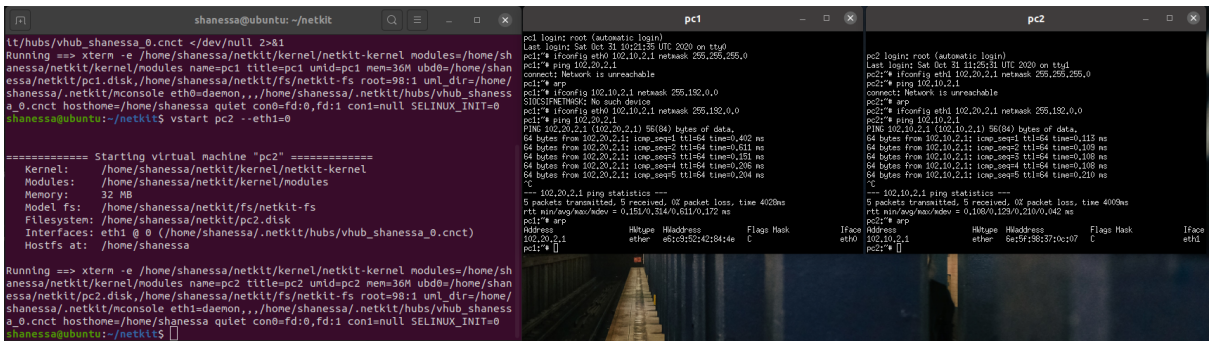
Check whether your configuration was successful by using ping command between these two nodes.

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.*

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

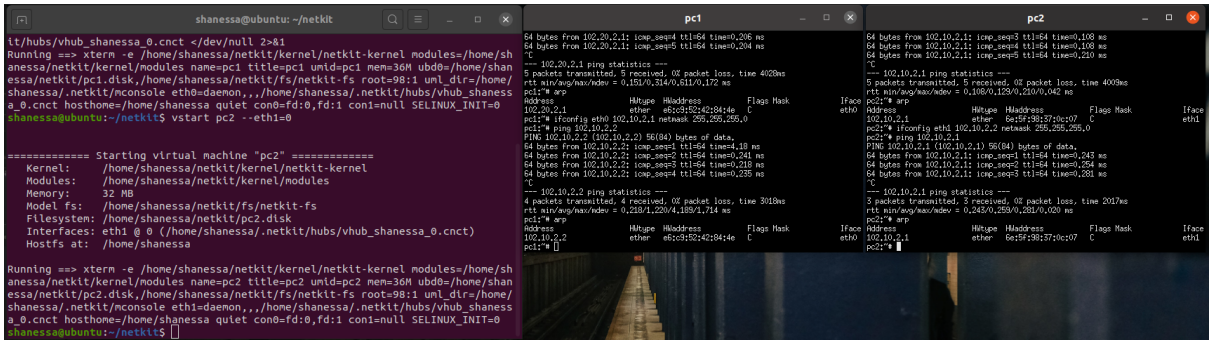
- *The ARP of PC 2 already on the PC 1 and vice versa.*



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:*

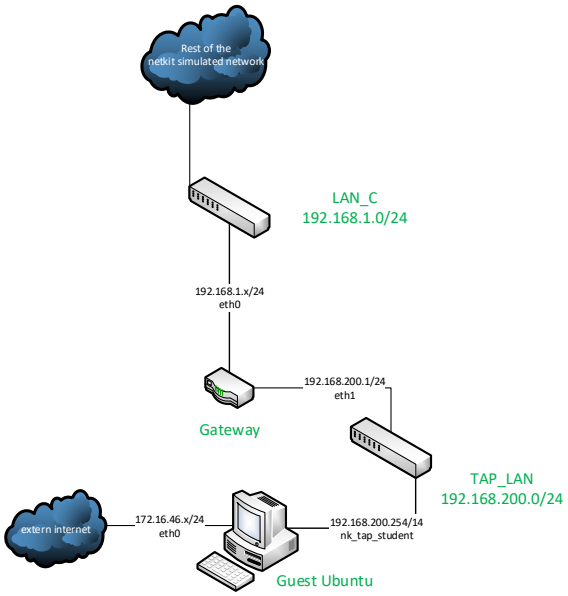
1. Use either ifconfig or ip commands. Once you know how the commands should look like, it is highly recommended to put them in <node>.startup files, so next time you want to restart and present your lab, you don't have to reconfigure it by hand again. Note : Please don't remove the commands which are already present in the <node>.startup files. They are necessary for starting up Linux networking service.

2. 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.



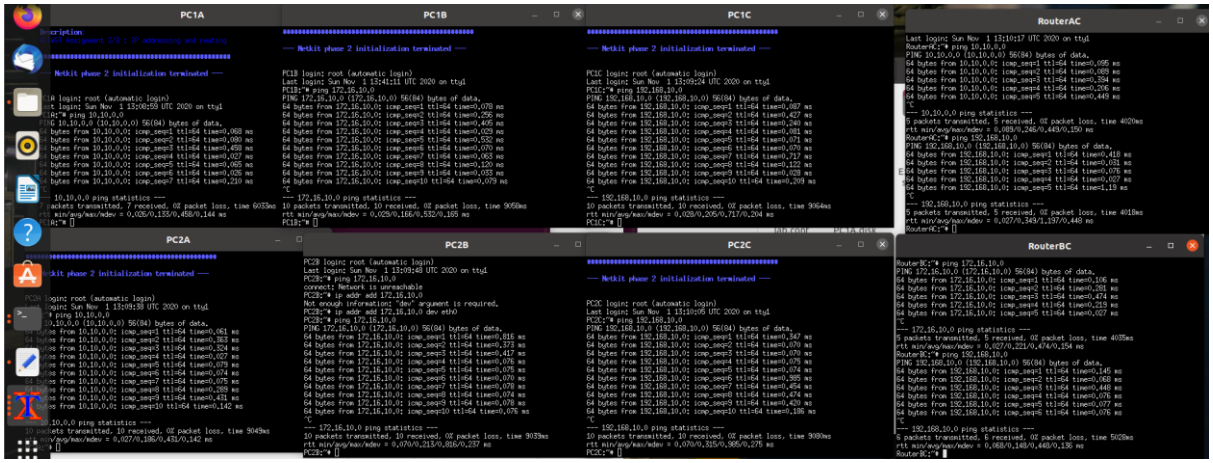
Provide the network drawing *of your lab network you can use* https://app.diagrams.net/ *and screenshots of the pings which are possible WITHIN LANA, LANB and LANC (PC1A to PC2A, PC1B to PC2B and so on). When creating* the *network drawing, don't forget to mention* the *IP addresses/subnet masks for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

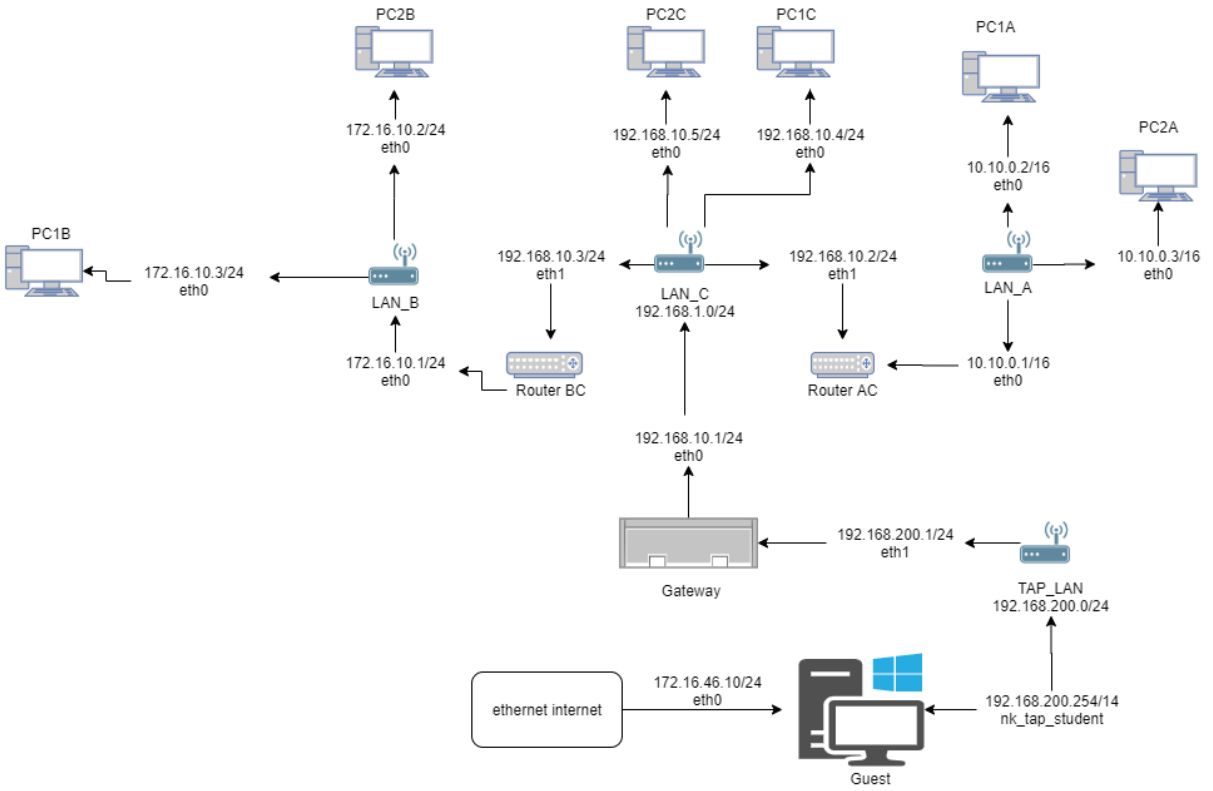
You don't need to be able to route between all nodes of this network; that is the second part of the assignment, which will be done next week.

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





***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**

WEEK 9 - IP Routing

Task 1a: Online exercises

*Complete **all** the online exercises in the following URL and provide a screenshot as evidence*

https://courses.codemax.net/w9.html

Networking exercises (week9)

direct/indirect routes | route configuration | routing tables

Completed exercises

Network routing		
direct/indirect routes	:	25 / 20
route configuration	:	20 / 20
routing tables	:	15 / 15

Task 1b: A bit more complex network: Part 2

Last week you did the configuration of your IP network for the preconfigured lab.

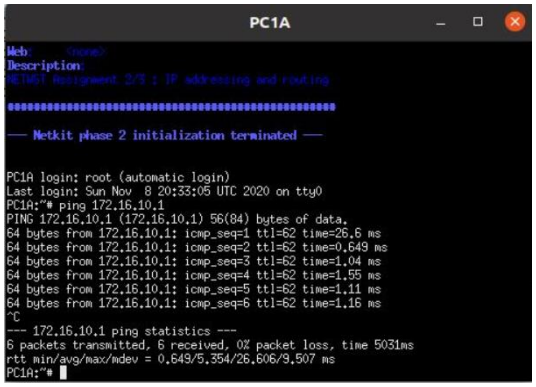
If you have done well and used either scripts or network/interfaces files, you should be able to restart your configured environment again. Also, you should have a drawing of your network.

Your task is adding routing information to your nodes in such a way, that every node of your network should be able to ping any other node of your network. The routes should be optimal, so the shortest path from node to node should be used. To implement routing, you'll have to use different types of routes as learned on the theory lesson.

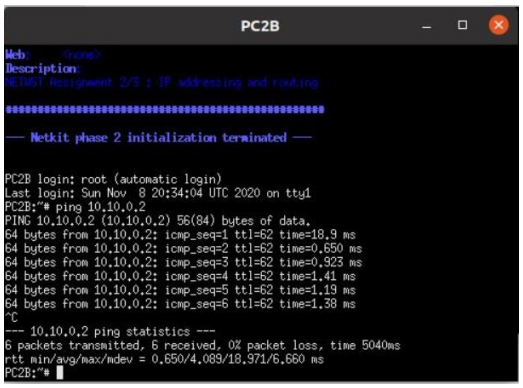
*Tip: Use the network drawing from the last week assignment (week 8) and first think about the way you're going to route. Use **tcpdump** and **traceroute** commands to debug your routing.*

Provide screenshots of the following pings:

1. PC1A to PC1B

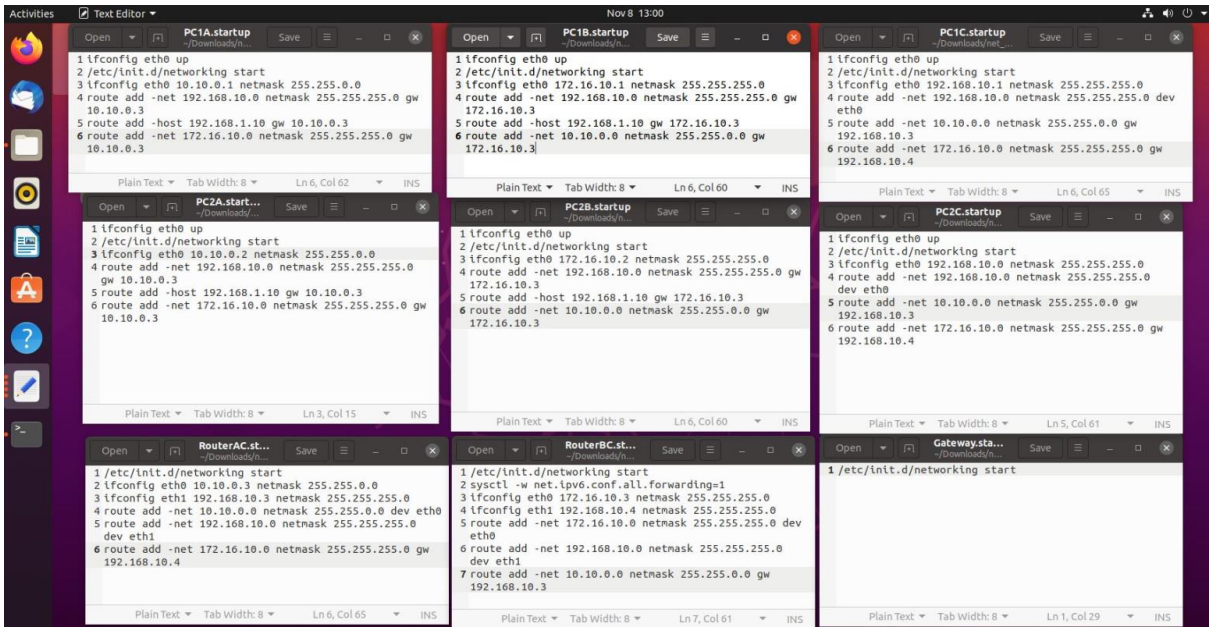


2. *PC2B to PC2A*



3. PC2A to PC1C

Give a list of all nodes where you had to adjust the routing tables and the screenshots of their configured routing tables.



WEEK 10 – TCP/UDP

Task 1: TCP in Netcat

To do this assignment we will use the Netcat tool which is provided in the Netkit. Netcat makes it possible to create and use TCP/UDP connections. If you want more info about Netcat you can consult Internet. To make this assignment we will reuse the net_routing lab from the previous assignments. Let's start a chat session by connecting 2 netcat instances via a TCP connection.

To listen to the TCP connections, go to one of your simulated nodes (e.g. PC1A) and issue the following command:

nc –l –p <port_nr>

This will make netcat listen to port number that you have specified in port_nr and accept connections.

Note: Any port number would be ok, as long as it is not used by another application.

To establish a TCP connection you can issue the following command from another simulated node (e.g. PC1C)

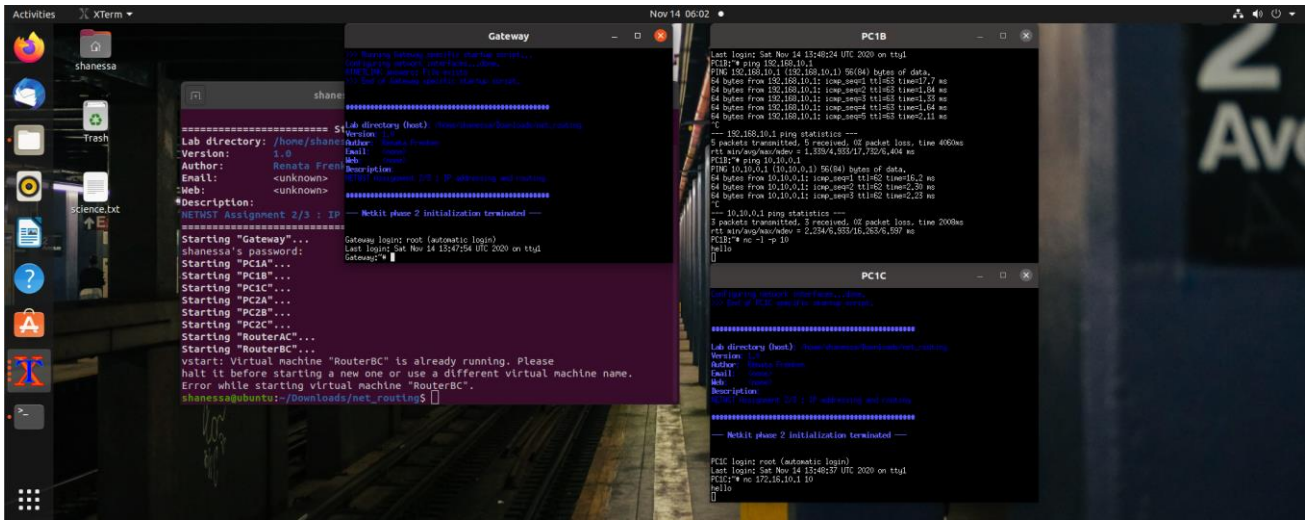
- **nc <IP address of the "listening" node> <port_nr of the "listening node">**

This will make a TCP connection with the listening netcat instance.Now you can chat from one netcat instance to the another. Try it out!

Your task:

- Netcat can also be used to copy the contents of a file from one place (file, folder, computer) to another. Find out how and try it out.

Provide screenshots of the sending and receiving command.



Let's now build a basic one-page webserver using netcat.

As a first step create a textfile 'response.txt' with following content:

HTTP/1.1 200 OK

Content-Type: text/html;

Content-Length: 12

Connection: close

Hello World!

This is a proper HTTP response.

You are going to simulate HTTP server and HTTP client (browser) using netcat. You are going to use Gateway node for running HTTP client (browser), see section below about how to install links browser on Gateway before you start the exercise. You are going to use any node of your network (e.g. PC2C) to simulate HTTP server.

Your task:

Construct an appropriate netcat command to listen to a port on your HTTP server (e.g. PC2C) and send the contents of the file response.txt to the HTTP client (Gateway) when a connection is made to this port. That is roughly what a webserver does too.
You can test it by entering the following URL in links browser on Gateway.

http://<IP_ADDRESS_OF_WEBSERVER>:<port_nr>

If everything works well the links should show the "Hello World" webpage.

Links installation:

Before starting this task, you have to install links text-mode web browser. Follow these steps:

- Go to your lab's Gateway directory. Create etc subdirectory and create there a resolv.conf text file with the following contents:
  nameserver 8.8.8.8
  search localdomain

- Start your lab. Now you should be able to connect to the Internet, so do the following installation on the Gateway node:
  apt-get install links

Now the links web browser should be available on your Gateway node. Watch out, once you stop your Gateway node, you have to do 'apt-get install links' command again to install links.

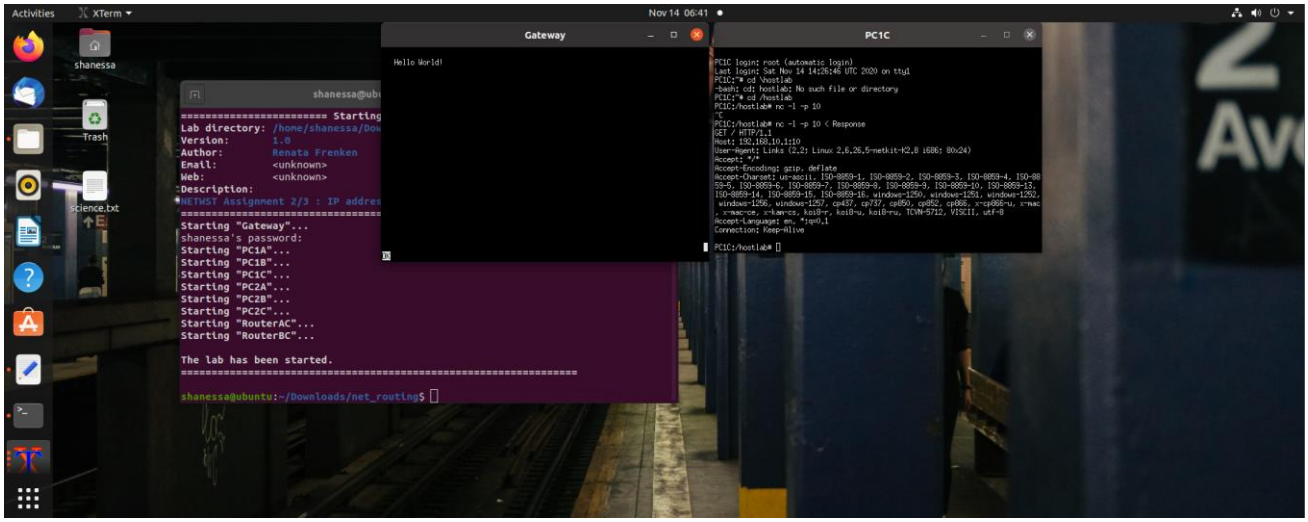
Note1: You can start links browser by issuing this command:

links

To be able to enter the URL in links browser press "G".

Note2 : You can put the response.txt file in your net_routing directory **before** start of the lab (lstart command). **After** starting the lab, you can find this file in the /hostlab directory of your node.

Provide a screenshot of the netcat command you used and of the links browser output.



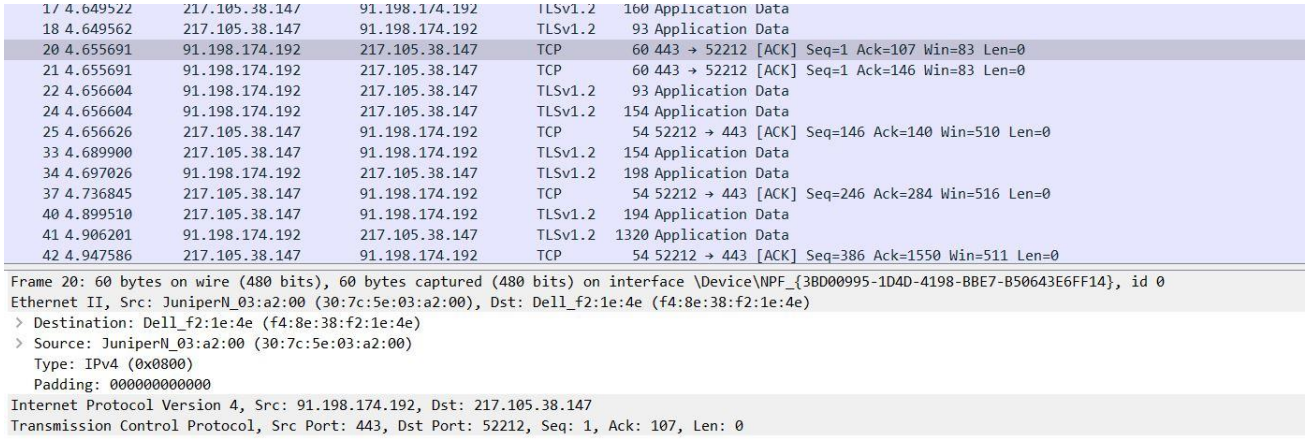
Task 2: Find 2 TCP uses

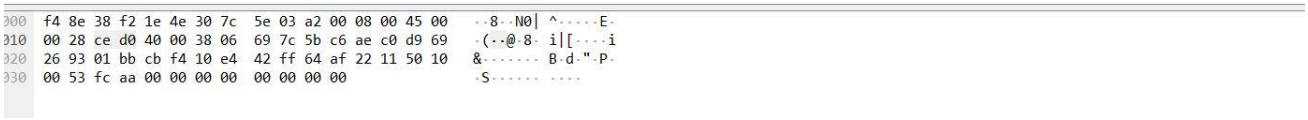
Think about two different scenarios for TCP use that you can simulate (you can do this on your own PC, so you don't need Ubuntu for this). Start a Wireshark trace for both scenarios.

Describe the chosen scenarios and a proof of TCP use in them by attaching a Wireshark trace showing TCP packets.

Scenario 1 : HTTPS Traffic

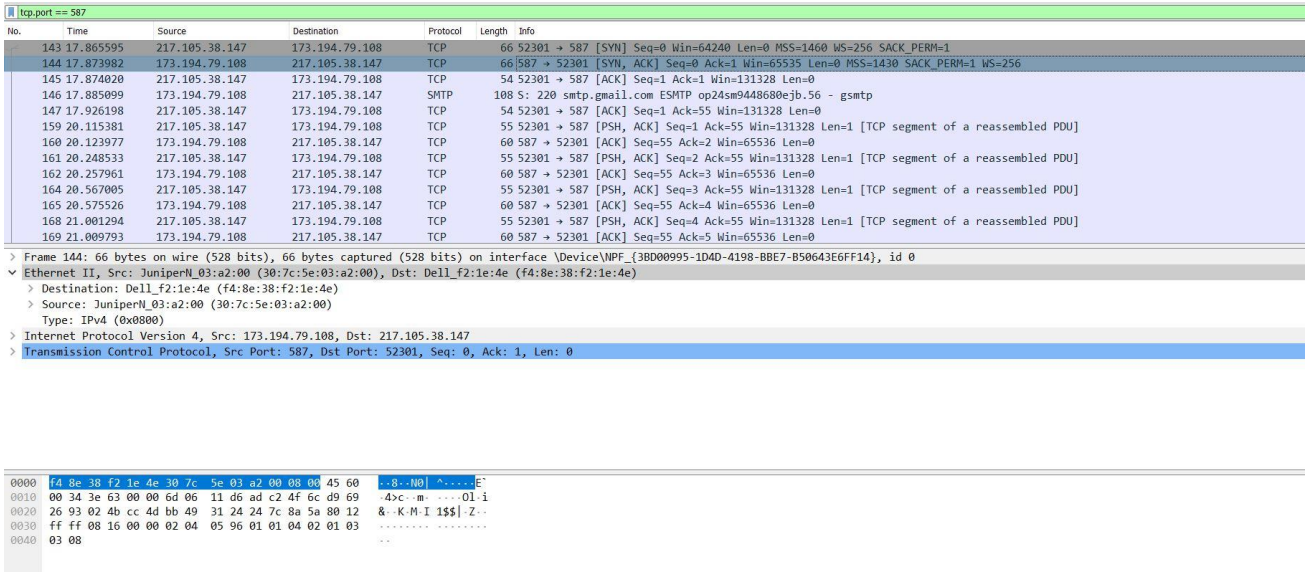
We capture HTTPS Traffic through opening multiple websites for a couple of seconds. We found traffic to the port 443. HTTPS is used for secure communication over computer networks and is widely used in the internet.





Scenario 2: SMTP Mail Server

We connect to the Google SMTP Mail Server using the Windows Telnet client to generate the traffic. The SMTP is a widely used communications protocol for email transmission. The TCP port was port 587.



Choose one of the 2 scenarios traces and browse it in the Wireshark. Select ‘Statistics > Flow Graph’ and then choose flowtype ‘TCP flow’ to draw a Sequence Diagram of the TCP message interaction that you see in Wireshark.

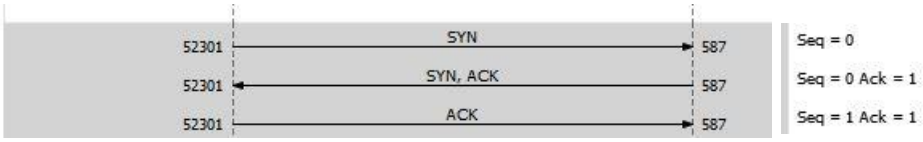
SMTP Sequence Diagram:

The Beginning:

The client (my computer) establishes a connection to the server, using the SYN (Synchronize sequence number) which informs server that client is ready to communicate.

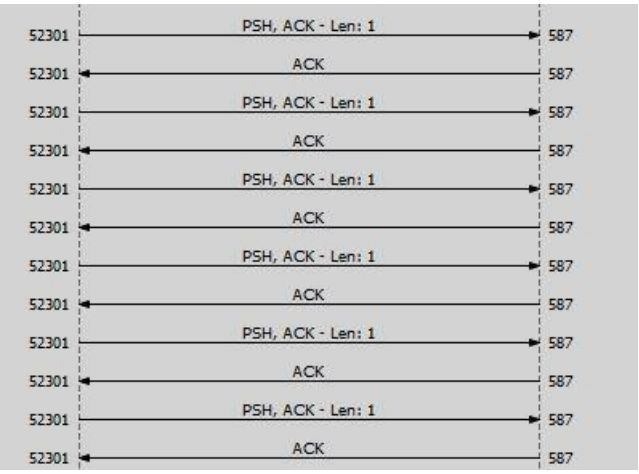
Then, the server responds with SYN + ACK signal, ACK (Acknowledgement) signifies the OK signal from the server side, and SYN signifies which sequence number it will try to start the transfer with.

The final ACK from the client side signifies the OK signal from the client side, finishing the “Three Way Handshake” and establishing a reliable connection between the two.



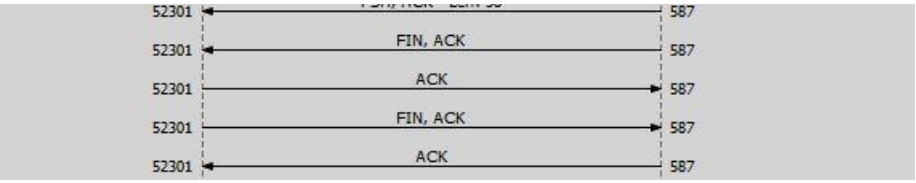
The LEN values signifies the length of the TCP payload in bytes. The SEQ value signifies the sequence number for communication, and the ACK value indicates that the server has acknowledged the client’s SYN flag and will proceed with the transfer.

The Middle:



In the middle, data transfer is done by continuous transfers and acknowledgement between the two parties. The PSH is the PUSH flag, that signifies the sending of data even when the buffer is not full during transfers!

In the End:



In the end, the FIN (FINISH) flag is established. This flag signifies the end of data transfer and the closing of the connection.

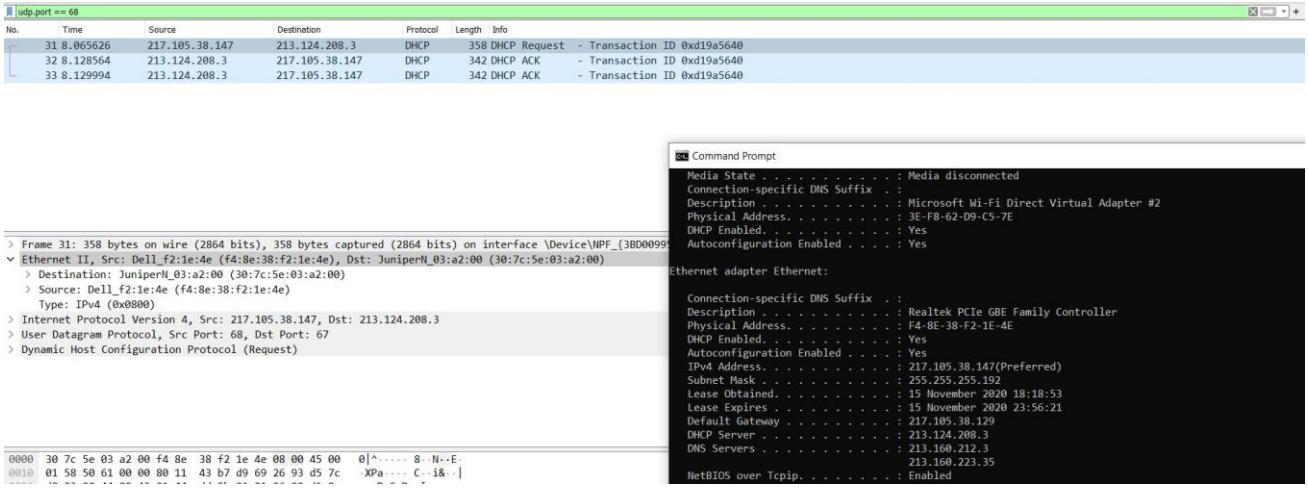
Task 3: Find 2 UDP uses

Think about two different scenarios for UDP use that you can simulate (you can do this on your own PC, so you don't need Ubuntu for this). Start a Wireshark trace for both scenarios.

Describe the chosen scenarios and a proof of UDP use in them by attaching a Wireshark trace showing UDP packets.

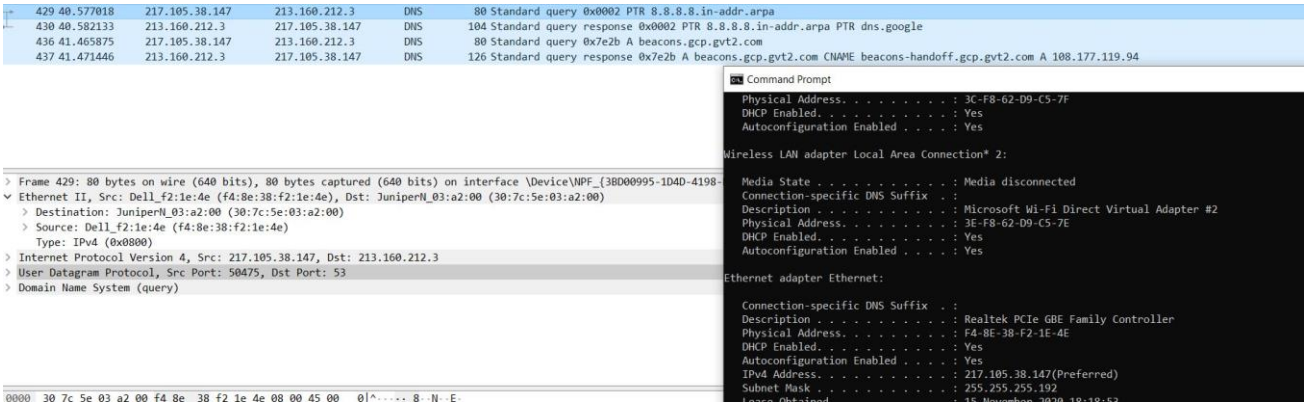
Scenario 1: Renew DHCP:

Using the ipconfig /renew command in the terminal, my DHCP assigned IP address is renewed. This will generate UDP traffic, particularly in port 68. Dynamic Host Configuration Protocol is a network management protocol that dynamically assigns an IP address to devices.



Scenario 2: Reverse DNS Lookup (rDNS)

By using the command nslookup 8.8.8.8, the Domain Name System protocol, which uses UDP, looks up the aforementioned IP address. DNS is a naming system that acts as a "finder" which translates domain names to the numerical IP addresses needed for locating networks under many different networking protocols. Reverse DNS Lookup generates traffic particularly in port 53.



Conclusion

I won't talk that much for the conclusion, on WEEK 7 we learned about networking basics. We need to install netkit and wireshark following the guide. In wireshark there's IP Address, MAC Address, Port, Host, etc. We learn the meaning and how it's work from IP Address, MAC Address, Port, Host on week 7.

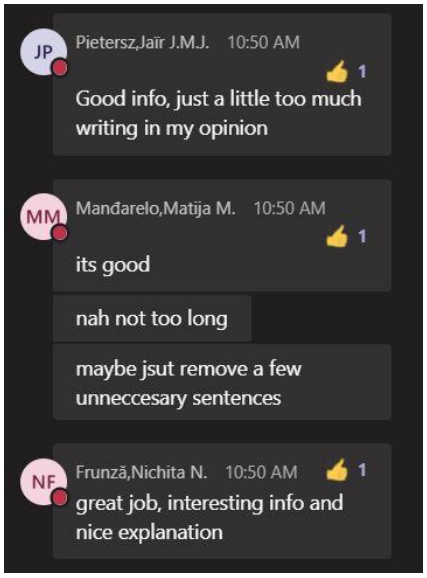
On WEEK 8 , we learn how to connect one pc to another using ifconfig, ping, and ARP. We used ifconfig to connect one pc to other pc and after we connected the pc we can type ping command to send request packets into a specific IP address and check whether the IP address responds back. If it does, that means that the network is connected to that address. And to discover MAC addresses on an associated IP address we can use ARP. We can use its cache to generally find records for IP and MAC addresses used in a network. After that we also made our own network using netkit and upload it to GitLab.

On WEEK 9 we learned about IP Routing. There's static routing and dynamic routing, Static routing provides better security since it is not possible for the router to "learn" unknown networks. Dynamic routing provides ease of use since its automatic nature does not require constant maintenance (like adding of networks, router traffic configurations, etc). For the task We added a route for every PC to connect to another corresponding PC from the routers through route add -net and We made the routers connect to each corresponding PCs in its domain (ex: Router A to PC1A, PC2A)

On WEEK 10, we learned about TCP and UDP. TCP (Transmission Control Protocol) is a connection-oriented protocol, where the source network (user) and the destination network (server) constantly acknowledge each other to ensure the data is fully delivered. It is reliable and less error prone, but slower and data-heavy! UDP (User Datagram Protocol) is a connectionless oriented protocol, where the server sends packets in bulk with minimal acknowledgement when the user requests it. Without constant data checks and error checking, this protocol is more prone to errors, but it is much faster and lighter than TCP. TCP have 2 scenario, first scenario is HTTPS. HTTPS is one of the most widely used application protocols in the internet. It provides secure encrypted network connection through the use of digital certificates. By opening multiple websites and using Wireshark, we sniffed its TCP connection and found it using port 443. And the second scenario is SMTP. The Simple Mail Transfer Protocol (SMTP) is a communication protocol for electronic mail transmission. We used telnet in Windows command prompt to generate a connection to the Gmail mail server, then we used Wireshark to sniff the connection. We discovered that SMTP uses port 587.
UDP also have 2 scenario, first scenario is reverse DNS lookup. The Domain Name System (DNS) is the phonebook of the Internet. Humans access information online through domain names, like nytimes.com or espn.com. Web browsers interact through Internet Protocol (IP) addresses. DNS translates domain names to IP addresses so browsers can load Internet resources. We used wireshark on a reverse DNS lookup operation and discover it uses UDP on port 53. And the second scenario is DHCP IP Address renewal. Dynamic Host Configuration Protocol is a network management protocol that dynamically assigns an IP address to devices. It uses UDP. We used the Windows command ipconfig/renew to renew a computer network's IP address and used Wireshark to sniff the operation. We discovered that DHCP uses UDP and uses port 68.

Personal Refraction

I realized that I've been using the internet for granted without understanding what is behind the scenes. I never expect that would be this hard, and I think I learn so many things from this assignment. And based on the presentation my group got a few feedback from the other students.



Yas, I also think our presentation is too long and talk about some unnecessary sentences, next time I'll try to make it shorter but still clear and understandable for others.

I think that's it for my personal refraction. Thank You.