

## Workshop 1 (3 weeks)

### Table of Contents

<b>Workshop 1 (3 weeks)</b> .....	<b>1</b>
<b>Objectives</b> .....	<b>1</b>
<b>Overview</b> .....	<b>1</b>
<b>Workshop Preparation: (before you arrive at the lab)</b> .....	<b>2</b>
<b>Tasks and Questions:</b> .....	<b>2</b>
<b>Part 1 – Emulating a LAN using GNS3.</b> .....	<b>3</b>
Step 1 - Create a basic local area network (LAN) with an ethernet switch in GNS3. ....	3
Step 2 – Use Wireshark to observe the packets on the network.....	5
Saving Machine States in Emulation .....	7
<b>Part 2 – Network Address Analysis using Python</b> .....	<b>9</b>
Step 1 – Complete all the tasks in the provided Jupyter notebook. ....	9
<b>Part 3 – Packet Capture and Analysis</b> .....	<b>10</b>
Step 1 - Use webterm and NAT to connect to the Internet from within GNS3 .....	10
Step 2 – Basic packet capture with modified links .....	12
Step 3 – Data Analysis of captured packets.....	12
<b>Workshop Report Guidelines:</b> .....	<b>15</b>

### Objectives

- Learn GNS3 basics and emulate a LAN in GNS3.
- Learn Wireshark basics and capture packets.
- Analyse packet flows, identify network protocols and gain hands-on experience with the layered network architecture.
- Practical experience with network addressing, IPv4 and MAC addresses.

#### Common objectives of all workshops:

- Gaining hands-on experience and learning by doing!
- Understanding how theoretical knowledge gained in lectures relates to practice.
- Observe how networks and systems operate in practice and develop curiosity for gaining further theoretical knowledge.

### Overview

Wireshark is a well-known *network packet analyser*. A network packet analyser captures network packets and displays captured packet data as detailed as possible. Therefore, you can think of it as a measuring device used to examine what's going on inside a network cable, just like a multimeter is used to examine what's going on inside an electric cable. In the past, such tools were either very expensive, proprietary, or both. Wireshark is free, open source, and one of the most popular packet analysers available today.

GNS3 is a powerful and realistic network emulator that is used by hundreds of thousands of network engineers worldwide to emulate, configure, test, and troubleshoot virtual and real networks. It emulates real computing and network hardware such as ethernet switches, DHCP servers, NAT, routers, terminals, and user devices.

We will use both GNS3 and Wireshark to gain hands-on experience with the network protocol stack, network layers, and encapsulation. We have discussed a basic overview of these in the lectures and will continue to do so in the upcoming lectures. **Note** that the learning order does not always have to be first lecture, then workshop. It can well be a lecture (overview), workshop, and lecture (details).

## Workshop Preparation: (before you arrive at the lab)

We recommend that you prepare before coming to workshops and learn much more in workshops!

We will give you a lot of time to finish the tasks but those are the bare minimums. Just like in the lectures, the topics we cover in the workshops are quite deep and we can only do so much in two hours. There is much more to learn and being prepared for the workshop is one of the best ways to gain more knowledge! For example, there are a few questions in each workshop which you can answer beforehand.

**Self-learning** is one of the most important skills that you should acquire as a student. Today, self-learning is much easier than it used to be thanks to a plethora of online resources.

For this workshop, start by exploring the resource mentioned in the preparation steps below.

### Workshop Preparation Steps:

1. *Common steps for all workshops:* read the Workshop Manual beforehand!
2. Review relevant lecture slides on layering, protocol stack, and encapsulation.
3. Read the relevant online documentation of the software we use (Wireshark, GNS3, Networkx etc.)

#### **Did you know?**

- Wireshark is a serious software tool! Did you know that there is an annual SHARKFEST conference, "Wireshark University", and an official Wireshark Certified Network Analyst (WCNA) Program?
- GNS3 is used in companies all over the world including Fortune 500 companies.

## Tasks and Questions:

Follow the procedures described below and answer the workshop questions for your Workshop Report and oral quiz which you will prepare according to the instructions. Keep your answers short and legible!

**The goal is to learn**, NOT blindly follow the procedures in the fastest possible way!

## Part 1 – Emulating a LAN using GNS3.

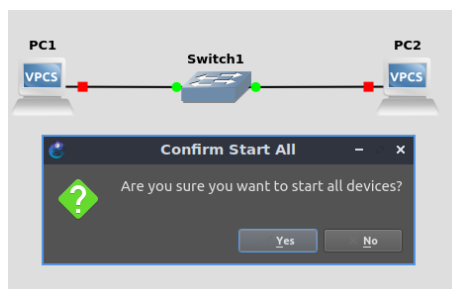
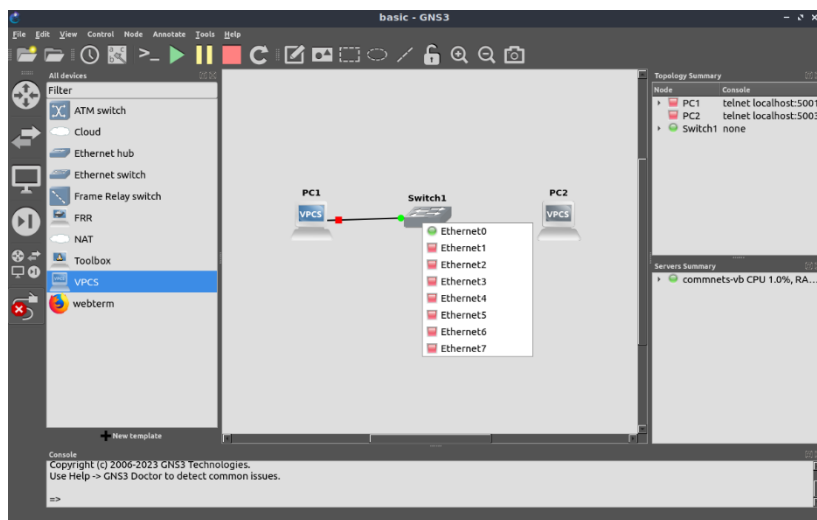
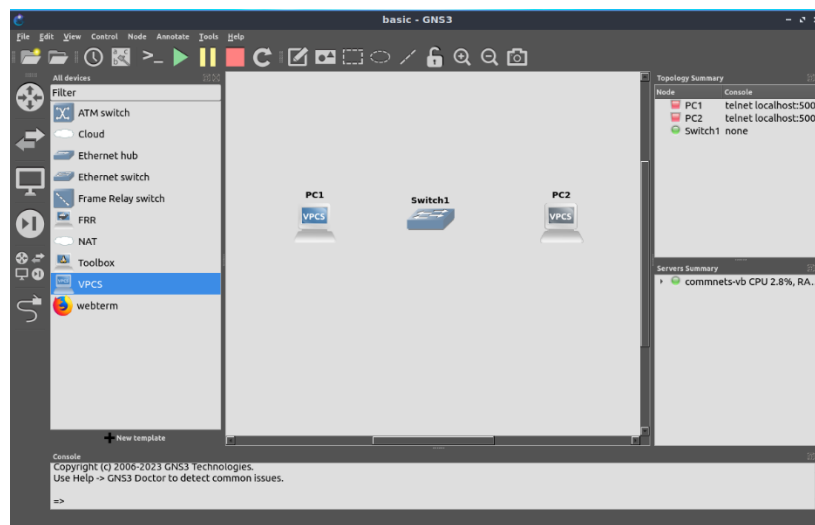
Step 1 - Create a basic local area network (LAN) with an ethernet switch in GNS3.

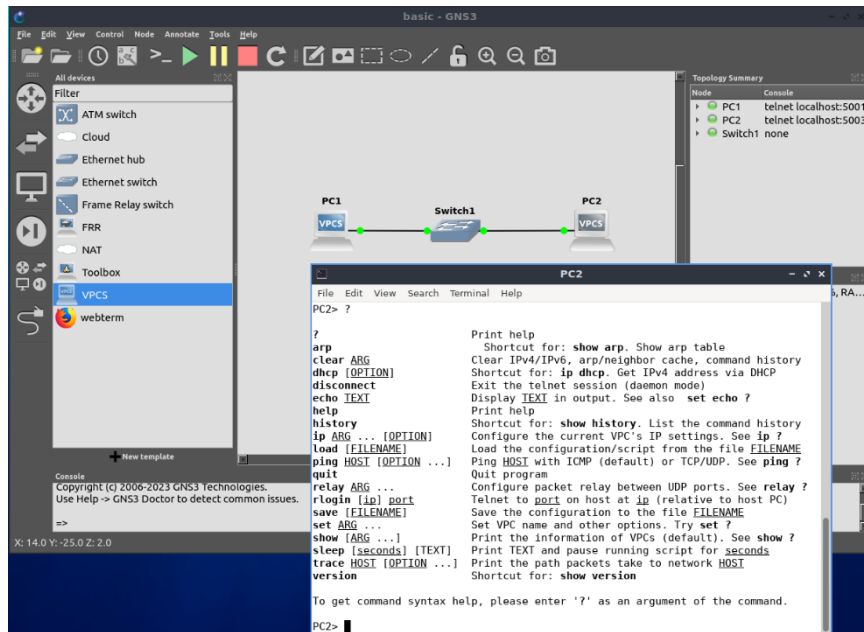
This is a very basic networking task but is valuable in exposing you to important networking concepts and fundamentals.

See the instructions <https://docs.gns3.com/docs/getting-started/your-first-gns3-topology>

and a relevant video <https://www.classcentral.com/classroom/youtube-gns3-start-here-if-you-are-new-to-gns3-80203/625113b5771d2>

The screenshots below show some of the steps (but not all)

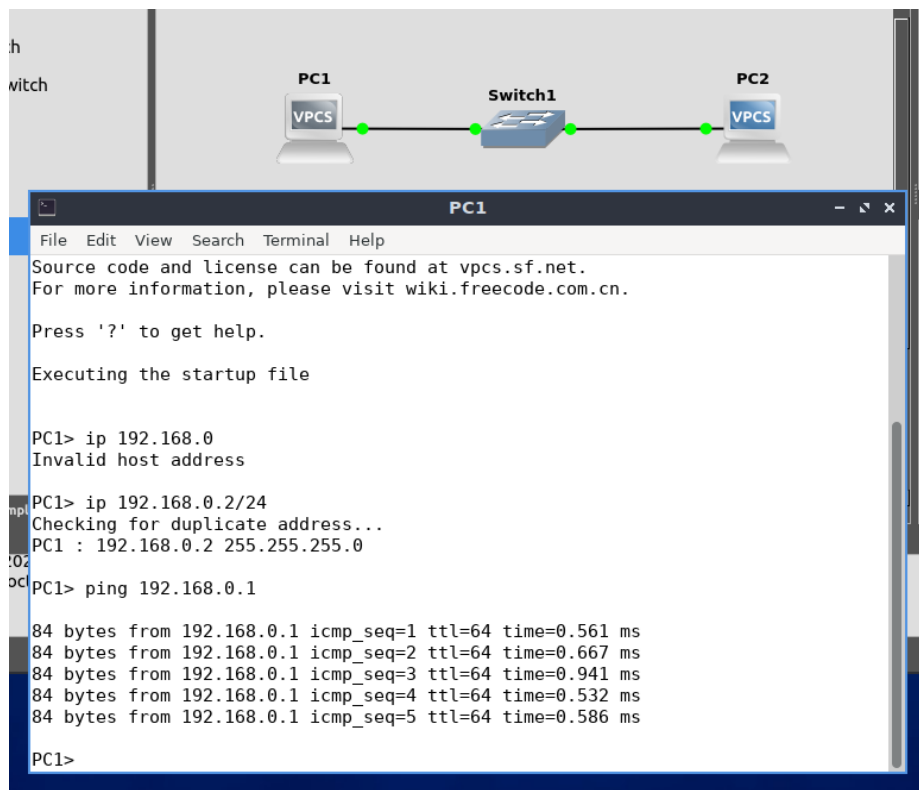




```
PC2>
PC2> ip 192.168.0.1/24
Checking for duplicate address...
PC2 : 192.168.0.1 255.255.255.0
```

```
PC2> show
```

NAME	IP/MASK	GATEWAY	MAC	LPORT	RHOST:PORT
PC2	192.168.0.1/24	0.0.0.0	00:50:79:66:68:01	10006	127.0.0.1:10007
	fe80::250:79ff:fe66:6801/64				



In this first step, you should achieve the following tasks:

- Create the wired LAN by connecting the terminals (VPCS) to the ethernet switch and start all devices. Remember, this is an emulator, which is more realistic than a simulator, so each device has an independent identity and needs to be started independently.
- Manually give IP addresses to your VPCSs. These are not real IP addresses, only local ones. Remember the lectures on IP addressing.
- Find the MAC address of each VPCSs.
- Ping is a well-known networking tool to say “hello” to a computer on the network.

**Reflect on the questions below and write your brief answers to the report**, maybe along with a couple of screenshots, to illustrate what you have done. You are expected to answer such questions during the oral examination at the end of each workshop and show your working system.

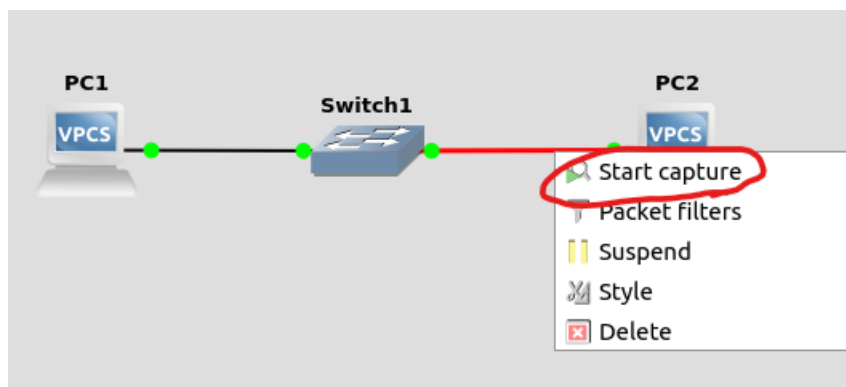
**Question 1.1.** Reflect on IP vs MAC addresses. What is the difference? Did you assign a MAC address to the VPCSs in the emulation? Where do they come from?

**Question 1.2.** How does the switch know where to send the packages? Would it be able to manage it if you had connected 3 or 4 VPCSs to the switch? If yes, how?

Step 2 – Use Wireshark to observe the packets on the network.

A great advantage of GNS3 is that it comes with Wireshark built in. Let’s see what things look like.

Right-click on a link and choose “start capture” to start Wireshark.



First, nothing is happening. Ping PC1 or PC2 again to see the ping packets captured by Wireshark.

*As a side note, observe that nothing is visible on the main GNS3 window. That is the issue with communications and networks. You cannot touch it or see it (unlike mechanical things). It requires abstract thinking (like mathematics) and a bit of imagination. Fortunately, our packet analyser shows us what is going on within those virtual wires!*

When you look at the Wireshark window, there is a wealth of information that was hidden behind the scenes, even in the case of a simple ping!

Carefully study the panels of Wireshark:

- The top panel shows the captured packets with time stamps, source, destination, protocol, and info notes.
- The middle panel shows detailed information about individual packets (blue highlighted here) and protocols associated with them.
- The bottom panel shows raw data in hexadecimal format and ASCII representation for convenience (each number is a byte).

The image shows a Wireshark packet capture window titled "Capturing from - [Switch1 Ethernet1 to PC2 Ethernet0]". The interface includes a menu bar (File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, Help) and a toolbar. A display filter box at the top contains "Apply a display filter ... <Ctrl-/>". The packet list shows several ICMP Echo (ping) request and reply packets. The selected packet (No. 3) is an ICMP Echo (ping) request from 192.168.0.2 to 192.168.0.1. The packet details pane shows the structure of the ICMP Echo request, including the type (8), code (0), checksum (0xe92e), and sequence number (1). The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Private_66:68:00	Broadcast	ARP	64	Who has 192.168.0.1? Tell 1
2	0.000169	Private_66:68:01	Private_66:68:00	ARP	64	192.168.0.1 is at 00:50:79:66:68:01
3	0.001401	192.168.0.2	192.168.0.1	ICMP	98	Echo (ping) request id=0x3
4	0.001527	192.168.0.1	192.168.0.2	ICMP	98	Echo (ping) reply id=0x3
5	1.003430	192.168.0.2	192.168.0.1	ICMP	98	Echo (ping) request id=0x3
6	1.003831	192.168.0.1	192.168.0.2	ICMP	98	Echo (ping) reply id=0x3
7	2.005358	192.168.0.2	192.168.0.1	ICMP	98	Echo (ping) request id=0x3
8	2.005568	192.168.0.1	192.168.0.2	ICMP	98	Echo (ping) reply id=0x3
9	3.007066	192.168.0.2	192.168.0.1	ICMP	98	Echo (ping) request id=0x3

Frame 3: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface -, id 0  
 Ethernet II, Src: Private\_66:68:00 (00:50:79:66:68:00), Dst: Private\_66:68:01 (00:50:79:66:68:01)  
 Internet Protocol Version 4, Src: 192.168.0.2, Dst: 192.168.0.1  
 Internet Control Message Protocol  
 Type: 8 (Echo (ping) request)  
 Code: 0  
 Checksum: 0xe92e [correct]  
 [Checksum Status: Good]  
 Identifier (BE): 14044 (0x36dc)  
 Identifier (LE): 56374 (0xdc36)  
 Sequence Number (BE): 1 (0x0001)  
 Sequence Number (LE): 256 (0x0100)  
 [Response frame: 4]  
 Data (56 bytes)

0020 00 01 08 00 e9 2e 36 dc 00 01 08 09 0a 0b 0c 0d ... .6. ....  
 0030 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d .....  
 0040 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d .. !"# \$% &'()\*+,-  
 0050 2e 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d ./012345 6789;,<=  
 0060 3e 3f >?

Type (icmp.type), 1 byte Packets: 12 · Displayed: 12 (100.0%) Profile: Default

You can use the “apply a display filter” box on the top to filter different types of packets. Try this by typing “ARP” or “ICMP” there. This is especially useful when a lot of traffic is present (which is usually the case, unlike the sterile emulation environment we have created!)

Finally, let’s see what the ARP table looks like at the VPCS that we used for pinging the other one.

```

PC1
File Edit View Search Terminal Help

To get command syntax help, please enter '?' as an argument of the command.

PC1> show arp

arp table is empty

PC1> ping 192.168.0.1

84 bytes from 192.168.0.1 icmp_seq=1 ttl=64 time=0.262 ms
84 bytes from 192.168.0.1 icmp_seq=2 ttl=64 time=0.709 ms
84 bytes from 192.168.0.1 icmp_seq=3 ttl=64 time=0.409 ms
84 bytes from 192.168.0.1 icmp_seq=4 ttl=64 time=0.617 ms
84 bytes from 192.168.0.1 icmp_seq=5 ttl=64 time=0.350 ms

PC1> show arp

00:50:79:66:68:01 192.168.0.1 expires in 95 seconds

PC1> show arp

arp table is empty

PC1>

```

Reflect on the questions below and write your brief answers to the report.

**Question 2.1.** Which protocols do you observe in Wireshark during ping? Focusing on ping packets, which layers do you see in action? What does Wireshark tell us about them? Can you identify, for example, the MAC layer and find the MAC addresses? Which fundamental characteristics of the layered architecture do you observe in action? Elaborate based on what you have learned in lectures.

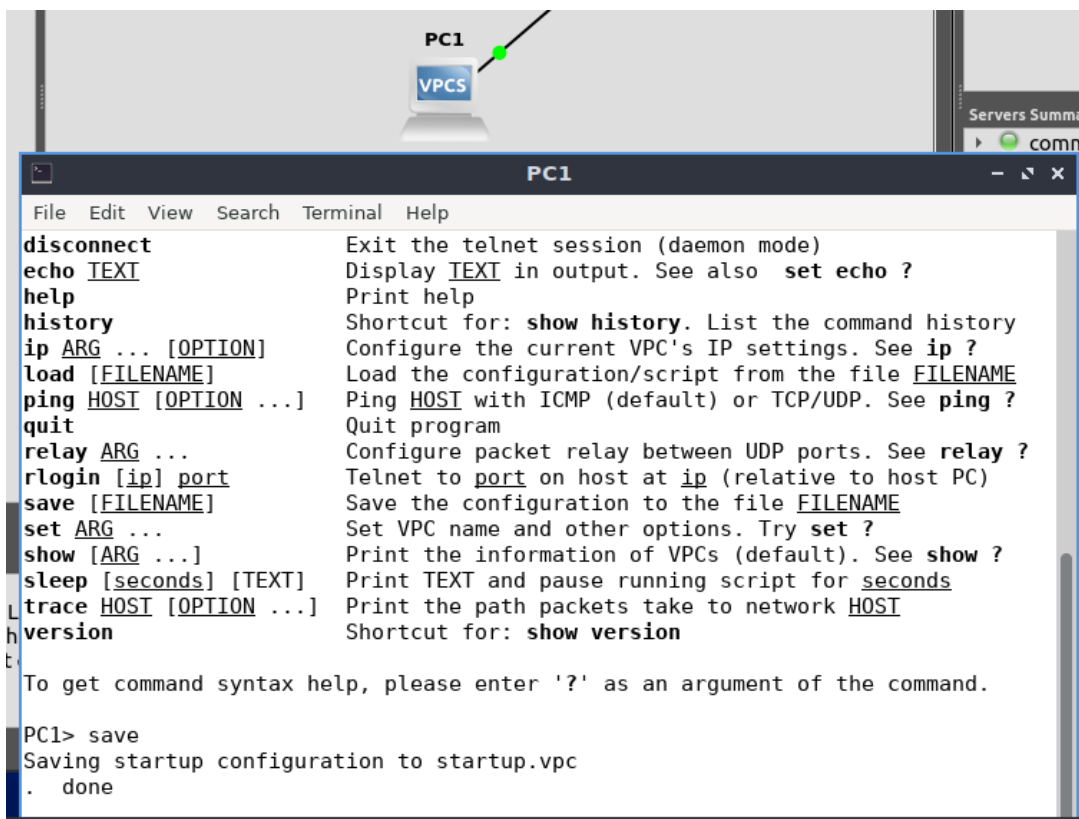
**Question 2.2.** Now focus on ARP protocol and tables. What do you observe about ARP? Comment on how it works. Discuss based on what you have learned in lectures and what you find on the internet, e.g., [https://en.wikipedia.org/wiki/Address\\_Resolution\\_Protocol](https://en.wikipedia.org/wiki/Address_Resolution_Protocol)

*Note that, maybe unsurprisingly, you can find excellent information on the internet and Wikipedia about internet protocols. Go get information online without hesitation (but don't believe everything you read before cross-checking)!*

## Saving Machine States in Emulation

Since GNS3 is an emulator, most devices on it are individual, separate entities. Therefore, *when you save a project in GNS3, it does not automatically save the state of the devices (unlike in simulators)*. Fortunately, there are methods for saving the state of the devices as described below:

**VPCS:** use the `save` command to save IP configuration. This way, next time you load the GNS3 project, it will retain its IP address/configuration.



```
PC1
VPCS

File Edit View Search Terminal Help
disconnect      Exit the telnet session (daemon mode)
echo TEXT       Display TEXT in output. See also set echo ?
help            Print help
history         Shortcut for: show history. List the command history
ip ARG ... [OPTION] Configure the current VPC's IP settings. See ip ?
load [FILENAME] Load the configuration/script from the file FILENAME
ping HOST [OPTION ...] Ping HOST with ICMP (default) or TCP/UDP. See ping ?
quit           Quit program
relay ARG ...   Configure packet relay between UDP ports. See relay ?
rlogin [ip] port Telnet to port on host at ip (relative to host PC)
save [FILENAME] Save the configuration to the file FILENAME
set ARG ...     Set VPC name and other options. Try set ?
show [ARG ...]  Print the information of VPCs (default). See show ?
sleep [seconds] [TEXT] Print TEXT and pause running script for seconds
trace HOST [OPTION ...] Print the path packets take to network HOST
version         Shortcut for: show version

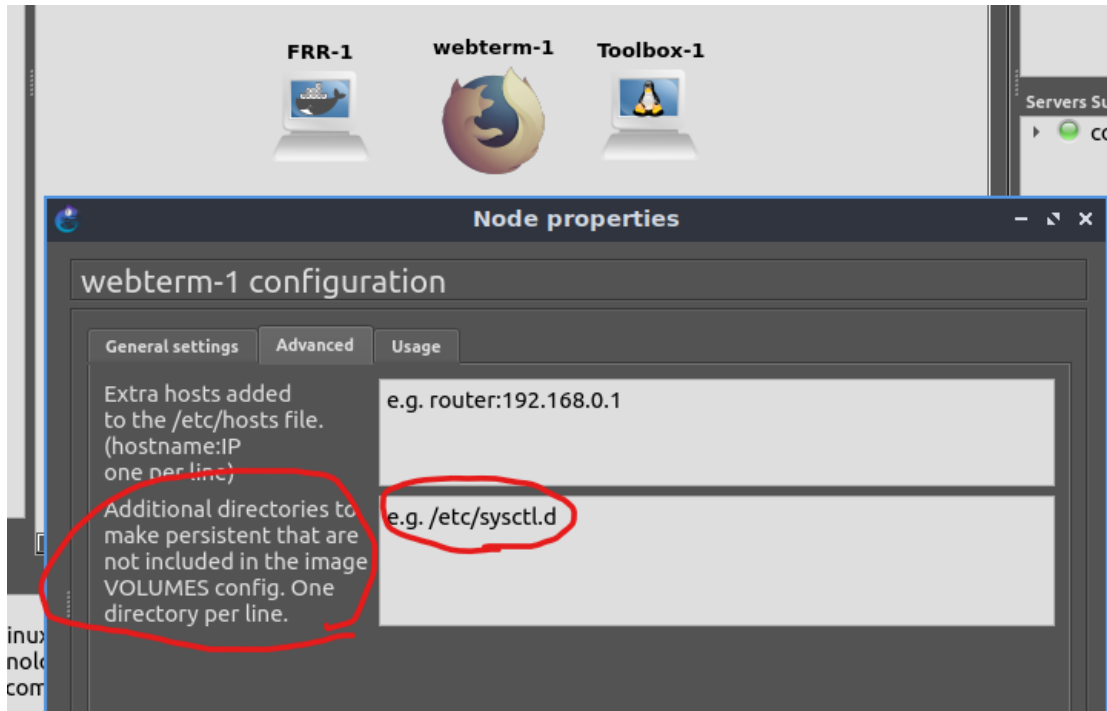
To get command syntax help, please enter '?' as an argument of the command.

PC1> save
Saving startup configuration to startup.vpc
. done
```

### Docker-based Devices such as FRR router or Webterm or Toolbox:

These are all Linux devices running over Docker. Their configurations are often stored under `/etc` folders in Linux filesystem. Docker provides multiple methods for persistency but that requires learning more about Docker containers. Fortunately, GNS3 makes our lives easy by providing a nice configuration option as shown below.

For example, in the case of Webterm, you can simply write `/etc` or `/etc/network` to the box provided to ensure that your Webterm device remembers the network configuration when you load the project next time (instead of repeating steps).





## Part 2 – Network Address Analysis using Python.

At this point, we switch gears a bit and will use the Python libraries to investigate network addressing. For this task, you will use the Anaconda installation on your own (or lab) computer (not the virtual machine). The Jupyter Notebook you are given contains all the tasks and you can write your answers directly there. Your report will be a simple printout of that notebook (which you can do from your browser's print function).

Step 1 – Complete all the tasks in the provided Jupyter notebook.

Follow the instructions in the Jupyter Notebook. Now you have a better idea of how addressing is done on the Internet.

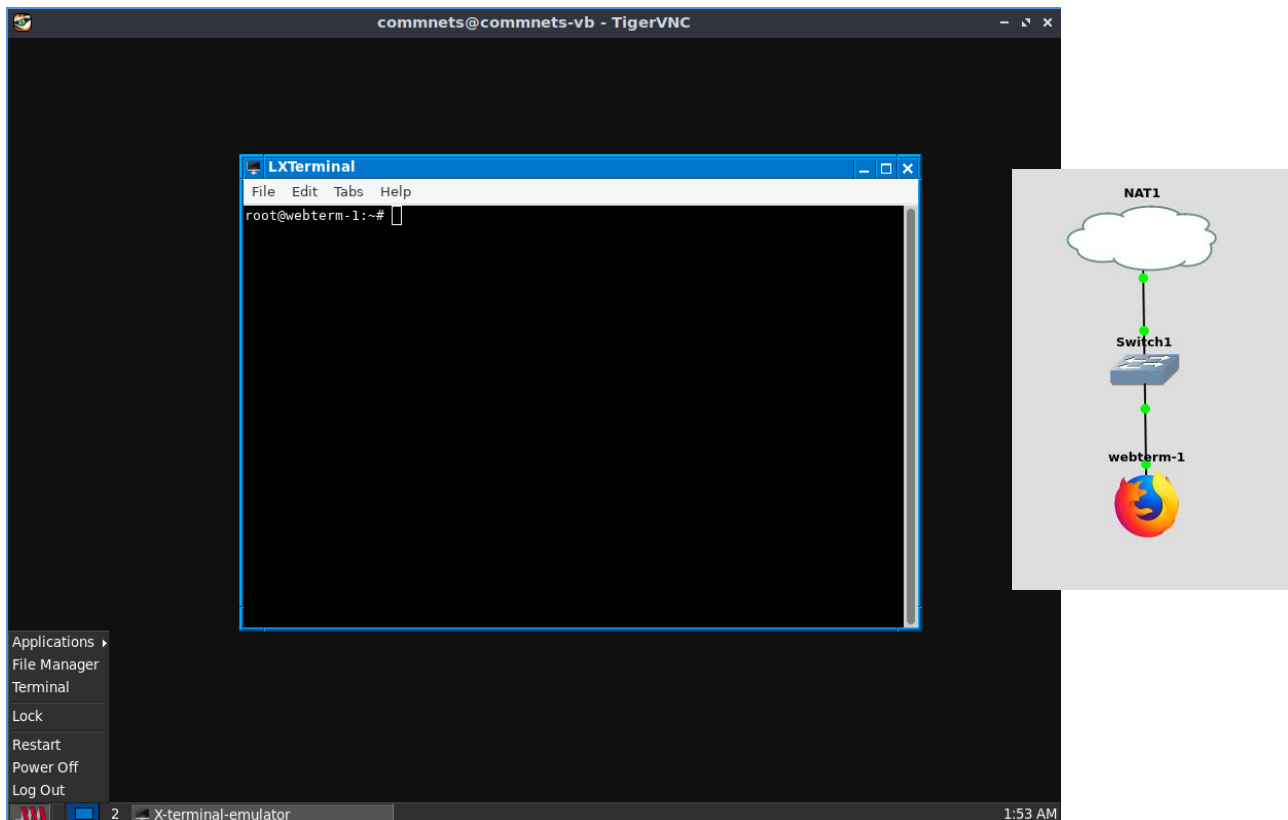
## Part 3 – Packet Capture and Analysis.

Next, we will use a browser on a device within GNS3 to connect to a server in the outside world and capture packets using Wireshark. Then, we will analyse them under various scenarios by modifying the link within GNS3.

### Step 1 - Use webterm and NAT to connect to the Internet from within GNS3

Here is what the setup looks like. **NAT** device creates the connection to the outside world and also runs a DHCP server, which we will use to get an IP address for our **webterm** device (which is a whole computer itself but has only a terminal and Firefox browser installed).

After starting the webterm, double-click on it and start the terminal there.



This is actually a Linux device, and we need to configure its network interface for it to get its IP address (we could have done this manually as well).

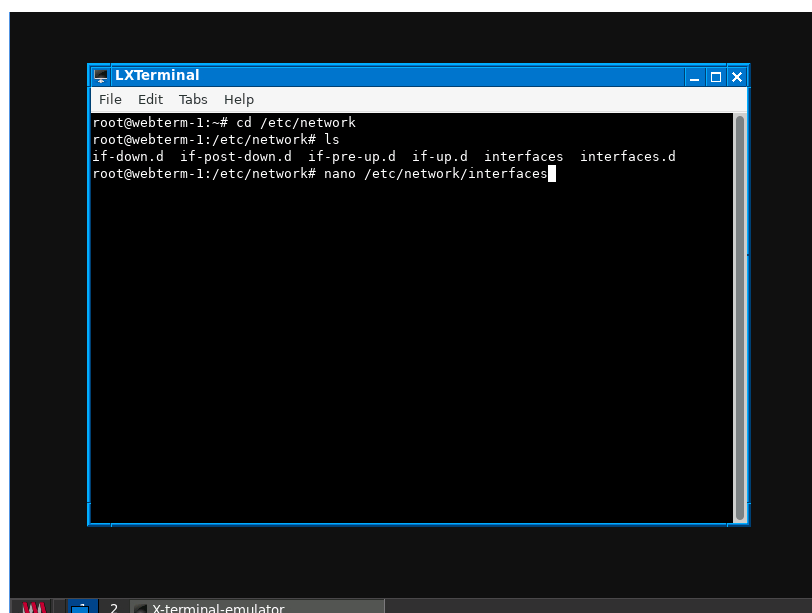
#### Run the command:

`nano /etc/network/interfaces`

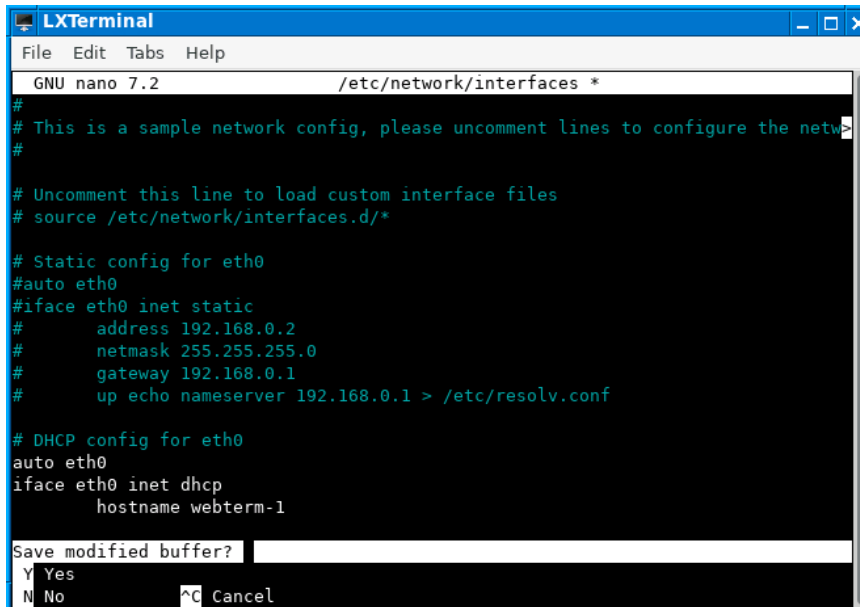
Nano is an old but useful text editor, especially when you are on a Linux terminal like this.

If you really want to get historical, you can use vi(m) instead. Vi(m) is one editor that is always available in Linux terminals!

<https://www.howtogeek.com/42980/the-beginners-guide-to-nano-the-linux-command-line-text-editor/>

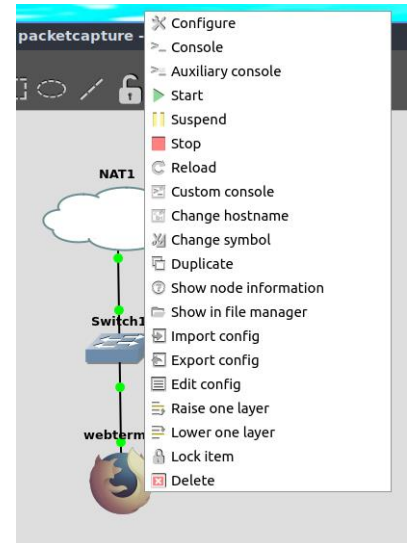


Within nano, uncomment the shown lines and press CTRL and X at the same time to exit. When it says, "Save modified buffer?" choose Yes if all is fine and don't change the filename.

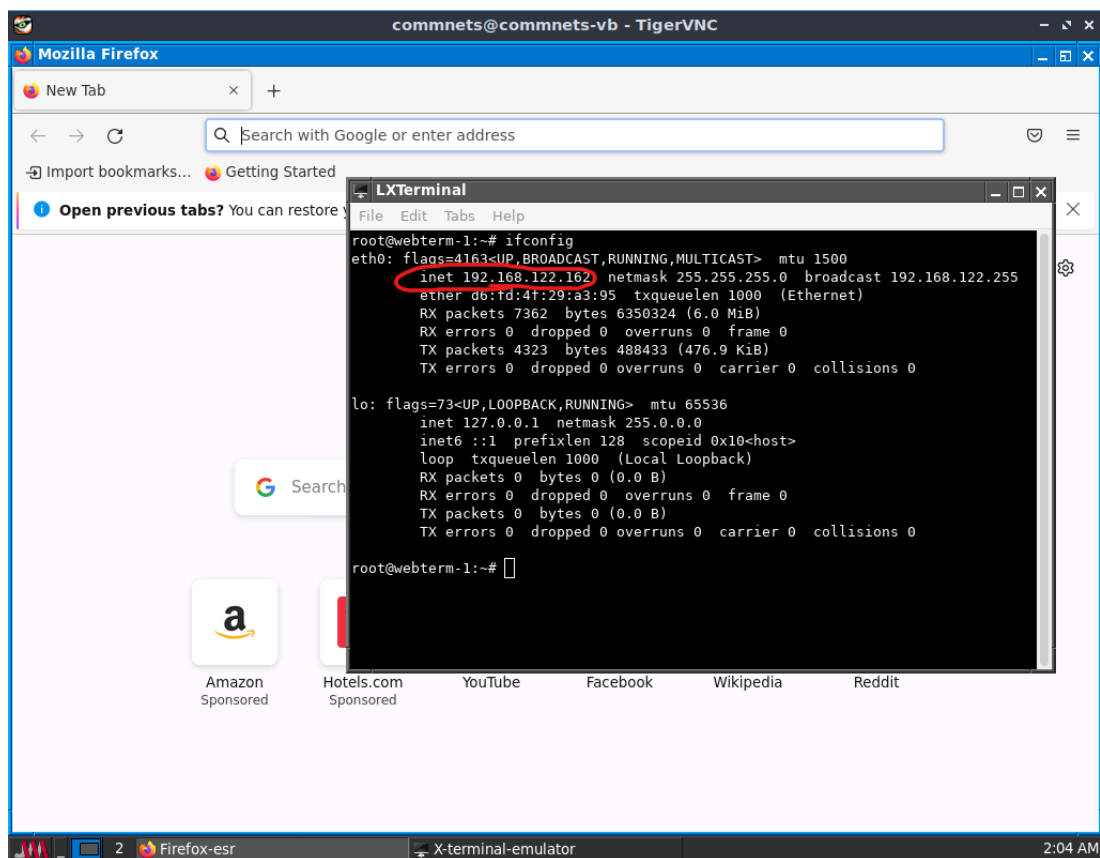


```
GNU nano 7.2 /etc/network/interfaces *
#
# This is a sample network config, please uncomment lines to configure the network
#
# Uncomment this line to load custom interface files
# source /etc/network/interfaces.d/*
#
# Static config for eth0
#auto eth0
#iface eth0 inet static
#    address 192.168.0.2
#    netmask 255.255.255.0
#    gateway 192.168.0.1
#    up echo nameserver 192.168.0.1 > /etc/resolv.conf
#
# DHCP config for eth0
auto eth0
iface eth0 inet dhcp
    hostname webterm-1

Save modified buffer? [Y] Yes [N] No [Ctrl-C] Cancel
```



As the final step, restart the webterm from GNS3 (stop and start webterm). Now, it will get the IP address. To see if it has worked, double-click on it, open the terminal inside and use **command** `ifconfig`



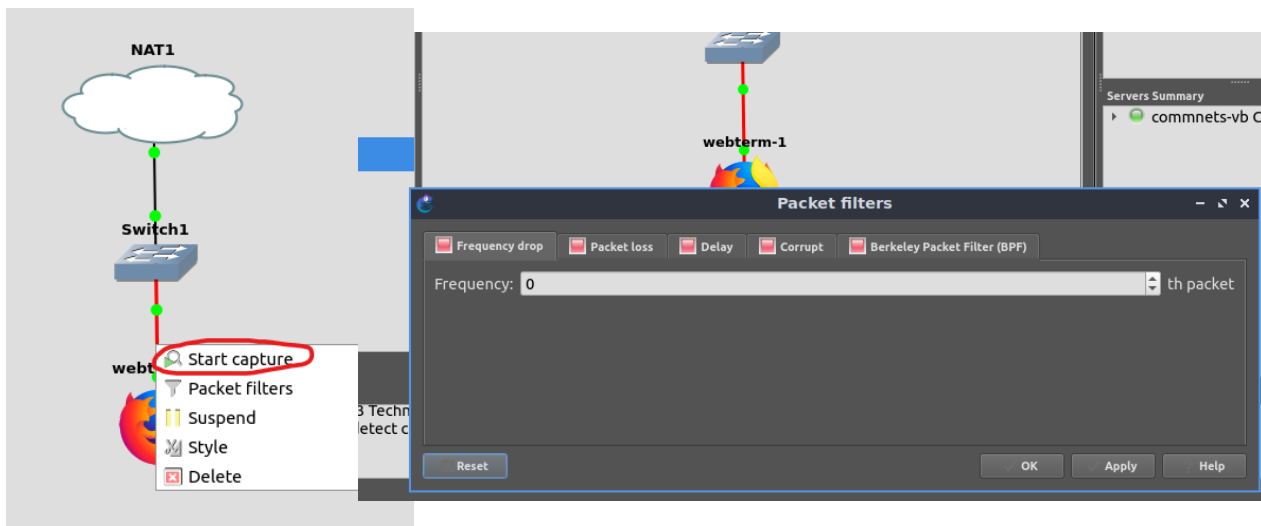
Now you should be able to connect to the internet using the Firefox browser within the webterm device!

## Step 2 – Basic packet capture with modified links

Let's capture packets using Wireshark as before (see figure left below)! Once capture starts, connect for example to [www.wikipedia.org](http://www.wikipedia.org)

Notice that now, there is much more going on as Firefox connects to Wikipedia web servers, as you can observe in Wireshark. We will learn about all those network layer protocols later. For now, what we want is to analyse the traffic patterns of the packets.

Before going deeper into this, however, there is an advantage of using GNS3 for this task. If you select **packet filters** in the right-click menu of any link in GNS3, you can modify the link characteristics (see figure below right)! Read more about this at <https://docs.gns3.com/docs/using-gns3/beginners/link-control/>



## Step 3 – Data Analysis of captured packets

Capture the packets while watching an online video. Possible options include ABC iview or YouTube. We will not worry about decoding the https packets (all traffic on the internet is encrypted with TLS these days – you will learn about this later in the subject).

Here are detailed steps as a suggestion:

1. Start the packet capture. Then, go to the website and select a video that is at least 2 minutes long.
2. Enter "tcp" as the capture filter option.
3. Watch for a minute or so and stop the capture.
4. Look for the IP address the video is coming from. Which Source IP address occurs most in your capture? Click on the packets you suspect most (of being video service servers) and use the "Menu->Analyse->Follow->TCP stream" to check focus on this data connection.
5. You will see something like "tcp.stream eq [a number]" appear in the Display filter on the toolbar automatically. *This number indicates that this is the xth TCP or UDP stream found in the trace.* You have now filtered the video packets and displayed only them (but others are still in the capture file). You can modify the filter to show only the incoming packets to your computer: "tcp.stream eq [a number] and ip.dst == xxx.xxx.xx.xx".
6. Ensure that only the target packets are displayed (using Step 5), then save these packets by choosing "Menu -> File -> Export Specified Packets". Note that in the screenshot of the file saving dialogue window below.
7. Choose "Menu -> File -> Export Packet Dissections" and save it as a CSV file.
8. Close the original capture (it may be a good idea to save the entire capture as backup).

You may, however, adopt the steps above to your own style as you prefer.



**Question 3.1.** What does the CSV file contain? Does it contain the packet contents?

Open the CSV file in MATLAB or Python or Excel, and check it.

Now, we will use analyse the interarrival times of the packets in the capture. You can use MATLAB or Python (e.g. with NumPy/SciPy, Pandas, matplotlib) for this.

If you use MATLAB, the recent versions have a powerful data import functionality. Open MATLAB and use the import wizard at "Menu -> File-> Import data". Create a new variable that contains only the time column.

The **MATLAB** functions you may use in analysing the packet interarrival times include:

*diff, find, prctile, hist, histc, bar, plot, hold, expfit, exppdf*

In Python, **NumPy** functions such as *histogram, diff, where*, as well as *scipy.optimize.curve\_fit* along with matplotlib, and pandas (e.g. *pandas.read\_csv*) may be useful.

In the following, each question should be answered for packets received from the server. In your report, do not simply give the final numerical answer to questions. Please include your calculations (e.g. MATLAB/Python commands you actually used).

**Question 3.2** Compute the sequence of packet inter-arrival times (IAT), i.e. the 'gap' times between packets. Give simple statistics about this time series: minimum, maximum, average, and standard deviation. In addition, calculate the instantaneous and average rate of the recorded flow.

**Question 3.3** 'Clean' the time series by selecting the subset of inter-arrival times smaller than the 95th percentile. Use this subset as the time series for the next questions. Comment on the statistics of this cleaned series: minimum, maximum, average, and standard deviation.

**Question 3.4** Plot the time series and histogram of the cleaned packet inter-arrival times (IATs). Add a title, and label the X and Y axis with the **correct** units. Include the figure in your report.

**Question 3.5** Using the curve fitting (e.g. MATLAB *expfit* function or *scipy.optimize.curve\_fit*), try to fit an exponential law to the distribution of the (inter-arrival) time series.

**Question 3.6** Plot the packet IAT distribution and superimpose the exponential distribution obtained with curve fitting. Include the figure in your report. Would you agree with the following statement for the packets captured?

"the packet arrival time series can be accurately modelled by a Poisson process."

Argue the case for or against the statement above. Note that the statement may well be wrong.

**Question 3.7 Repeat** Questions 3.2-3.6 with two additional traces in which you change link characteristics (introduce delays, random packet drops). Now, expand your discussion based on your observations on all these different data sets.

## Workshop Report Guidelines:

*You should complete the workshop tasks and answer the questions within the respective session!*

It is mandatory to follow all of the submission guidelines given below:

1. All figures produced and programs written should be uploaded to the right place in LMS by the announced deadline.
2. Filenames should be “ELEN90061 Workshop **W**: StudentID1-StudentID2 of session Day-Time”, where **W** refers to the workshop number, **StudentID1-StudentID2** are your student numbers, and **Day-Time** is your session day and time, e.g. Tue-14.
3. Submit your report as 3 separate files as follows:
  - A pdf file converted e.g. from a word file that contains the answers to Parts 1 and 3.
  - A pdf file printed directly from the Jupyter notebook as output of Part 2.
  - A zip file that contains all the codes (e.g. Matlab m files or Python scripts for Parts 1 and 3) **and** the Jupyter notebook ipynb file.
4. You should **NOT** zip all your files into one. You **MUST** submit your reports as separate PDF to pass the plagiarism check. Hence, we require 3 separate files.
5. Answers to questions, simulation results and diagrams should be embedded to your reports.
6. One report submission per group.
7. Check with your demonstrator in case of any questions.