Lab

1

Introduction to the Internet Lab

GNS3/Docker Version

- Installation of a virtual machines for Cisco routers and a Docker container for Linux PCs
- Setting up a network topology
- Saving and transferring data
- Navigating your way around Linux
- Working with the protocol analyzer *Wireshark*

It is assumed you have successfully completed the installation of VirtualBox and GNS3.

Updated: September 2020

Table of Content

Study Material for Lab 1

1. Read Appendix B to learn about the Linux file system and Linux commands.

2. **Man Pages:** The PCs run the Linux operating system. This assignment asks you to review some Unix commands. Man pages exist on every lab machine. You can also find the manual pages ("man pages") online at

   http://man7.org/linux/man-pages/

Step 1: For each of the following commands, type the name of the command as a search term.
Step 2: Read the man pages of the following commands:
`man`, `pwd`, `ls`, `more`, `mv`, `cp`, `rm`, `mkdir`, `rmdir`, `chmod`, `kill`, `ping`

3. **Wireshark:** Read about the *Wireshark* network analyzer at the website

   https://www.howtogeek.com/104278/how-to-use-wireshark-to-capture-filter-and-inspect-packets/

Step 3: There are numerous websites and videos that explain the operation of *Wireshark* (e.g., search for *"Wireshark 101"* on YouTube.com).
Find a few of these sources and learn about the syntax of display filter expressions in *Wireshark.*

Prelab 1

1. What will happen if you type "`man man`" in Linux?

2. How can you use the command "`ls`" to find out about the size of file `/etc/lilo.conf`?

3. What happens if you have two files with names *file1* and *file2* and you type "`mv file1 file2`"? Which option of "`mv`" issues a warning in this situation?

4. What is the command that you issue if you are in directory "`/`" and want to copy the file `/mydata` to directory `/labdata`?

5. What is the command that you issue if you are in directory "`/`" and want to copy all files and subdirectories under directory `/mydirectory` to directory `/newdirectory`?

6. What happens if you type the command "`rm *`" in a directory?

7. What is the command that you issue if you want to delete all files and directories under the directory `/mydirectory`?

8. Write the syntax of a Wireshark display filters to show packets with the following properties:

   a. IPv4 datagrams with source IPv4 address equal to 10.0.1.12.

   b. ICMP messages with source or destination IPv4 address equal to 10.0.1.12.

   c. IPv4 datagrams containing TCP segments with source or destination IP address equal to 10.0.1.12.

   d. IPv4 datagrams containing TCP segments with source or destination port number 23.

9. Write the syntax of a Wireshark display filters to show packets with the following properties:

   a. IP datagrams with a destination IP address equal to 10.0.1.50 and frame sizes greater than 400 bytes

   b. ICMP messages with source or destination IP address equal to 10.0.1.12 and frame numbers between 15 and 30.

   c. TCP segments with source or destination IP address equal to 10.0.1.12 and using port number 23.

Lab 1: Introduction to the Internet Lab

In the this and all other labs, icons in the page margin ask you do perform certain tasks. There are three different icons.

Icons in Lab instructions

Step 5: If you find the icon, you are asked to take a screen capture of a window on your desktop.

Step 6: The floppy disk icon tells you that you need to save data from the console of a PC or a router.

Step 7: The notepad symbol indicates an assignment or question to be included in the lab report.

In Part 6, the lab provides options for saving data. The following exercise shows how to take a screen capture.

Exercise. Screen captures

You can use the screenshot facilities of your computer to take a snapshot of all or parts of your desktop.

- **Windows PC**:
  - Hit ⊞-`Shift-S` and follow the instructions to select a snapshot. When the snapshot is completed, it is copied to the clipboard and a window appears in the lower right corner of your desktop. Clicking on the window opens the “Snip & Sketch” tool from where you can save the snapshot to a file. Alternatively, you can paste the content of the clipboard to a Word document. Alternative methods to take screenshots are discussed at https://www.cnet.com/how-to/8-ways-you-can-take-screenshots-in-windows-10/
- Mac OS:
  - `Shift-Command-4` allows you to select an area on the screen by dragging the mouse. After you release the mouse, a file with the screenshot on the desktop.
  - `Shift-Command-5` provides options for saving the entire desktop or a window on the desktop.
  - Screenshots are generally saved on the desktop.

Part 1. Importing a Cisco Router IOS image into GNS3

Cisco IOS in GNS3

To run a Cisco router in GNS3, we must install the IOS system software. This requires to having a file with a Cisco IOS image. There are two methods to install Cisco IOS for GNS3:

Step 8: If you have access an IOS image file you can install the IOS image installed on that router.

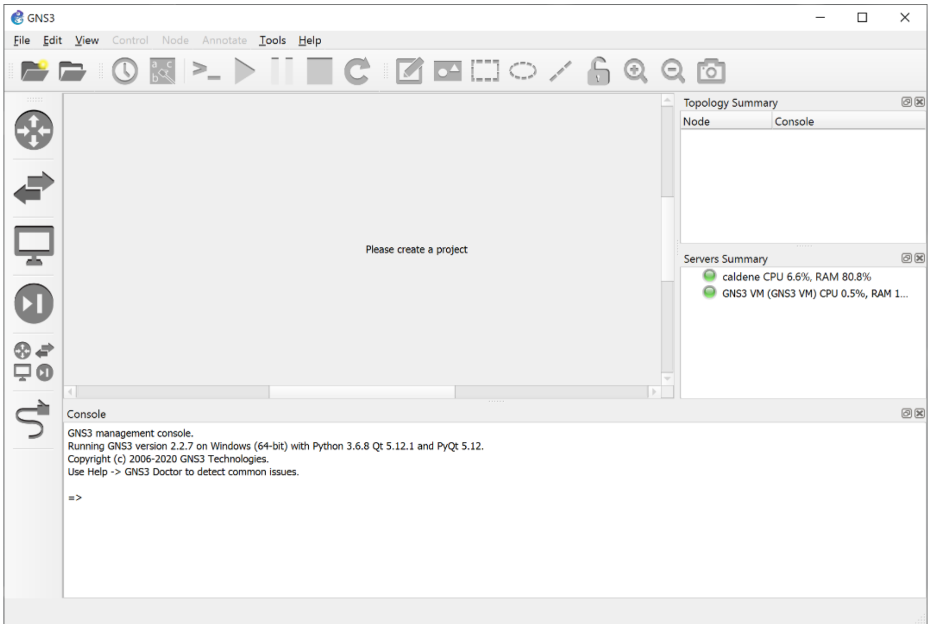
Step 9: You can purchase the Cisco Virtual Internet Routing Lab Personal Edition (VIRL PE) which provides images of virtualized routers. See: https://www.youtube.com/watch?v=jhh2_PP9JLU

The filename of the Cisco IOS image for a Cisco C3640 router has the format "c3640-xxxxx-xxxxxx.bin". The instructions assume that you have access to such an image file.

Before starting the lab, make sure that you have completed the software installation of VirtualBox and GNS3.

Exercise 1. Installing a Cisco Router in GNS3

Step 1: Start the GNS3 application. You should see a window as shown in Figure 1.1. Wait until the "Server Summary" pane shows a green status for both the local server (showing the name of your computer) and for GNS3 VM.



caldene CPU 6.6%, RAM 80.8%
GNS3 VM (GNS3 VM) CPU 0.5%, RAM 1...

```
GNS3 management console.
Running GNS3 version 2.2.7 on Windows (64-bit) with Python 3.6.8 Qt 5.12.1 and PyQt 5.12.
Copyright (c) 2006-2020 GNS3 Technologies.
Use Help -> GNS3 Doctor to detect common issues.

=>
```

Figure 1.1. GNS3 application.

Step 2: Go to:

- **Windows PC**: Select “Edit→Preferences”.
- **Mac OS**: Select “GNS3→Preferences”.

In the left-hand pane of the window that opens, click on “IOS routers” (if you do not see it click on “Dynamips”), and select “New” as shown in Figure 1.2.

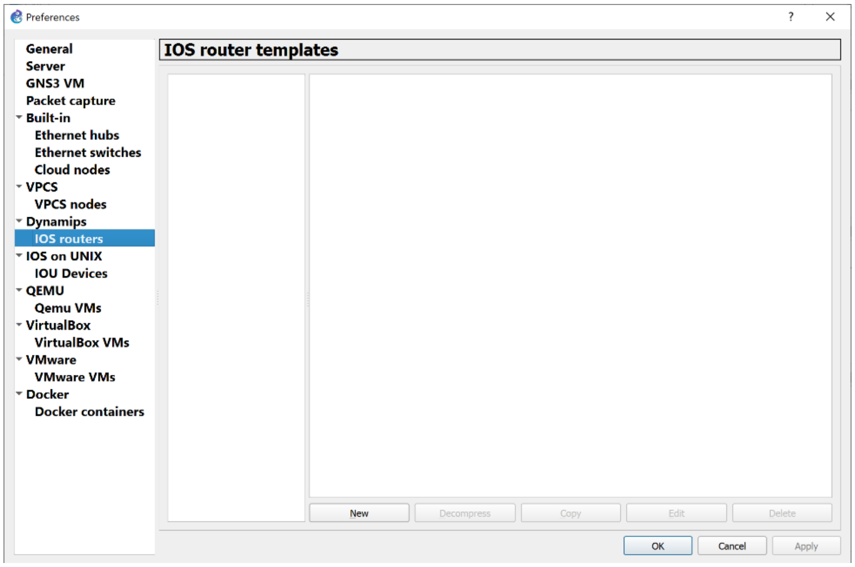
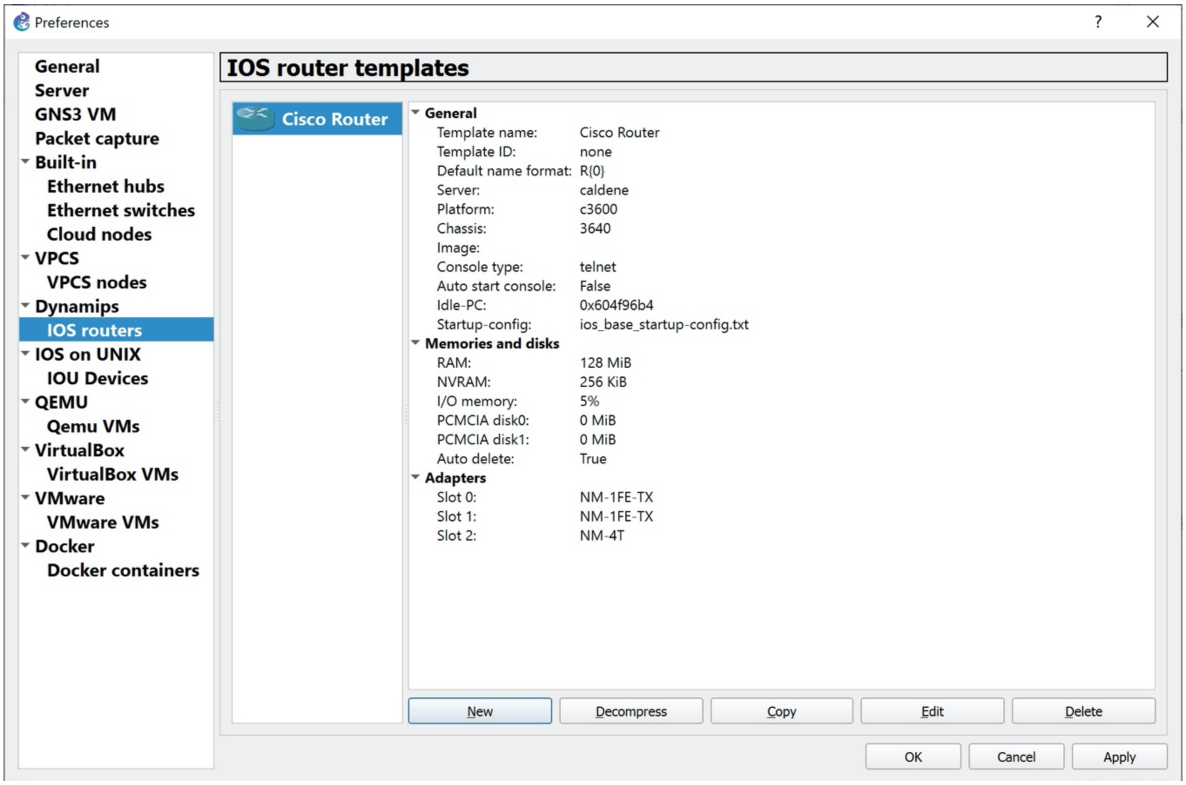


Figure 1.2. Creating a router template.

Step 3: You proceed through a series of windows, where you make selections as shown in the table below. Click “Next” or “Finish” after each selection.

Window title	Instructions
Server	• Select “Run this IOS router on my local computer”.
IOS image	• Select "Browse" to select the Cisco IOS image file (“c3640-xxxx-xxxx.bin”) on your computer.
Name and platform	• Change “Name” to *Cisco Router.*
Memory	• Set “Default RAM” to *128 MiB.*
Network adapters	• For “slot 0” and “slot 1”, select *NM-1FE-TX* from the dropdown menu. For “slot 2”, select *NM-4T*. • The selection NM-1FE-TX sets up a single Ethernet interface, and NM-4T sets up four serial interfaces.
Idle-PC	• Click on “Idle-PC finder” and wait for a minute. If the Idle-PC finder hangs for a longer time close the pop-up window. If the Idle-PC finder does not find a value continue with “Finish”.

When selecting “Finish”, the results of the router configuration are displayed in a window as shown in Figure 1.3.



IOS router templates

General

Template name:	Cisco Router
Template ID:	none
Default name format:	R{0}
Server:	caldene
Platform:	c3600
Chassis:	3640
Image:	
Console type:	telnet
Auto start console:	False
Idle-PC:	0x604f96b4
Startup-config:	ios_base_startup-config.txt

Memories and disks

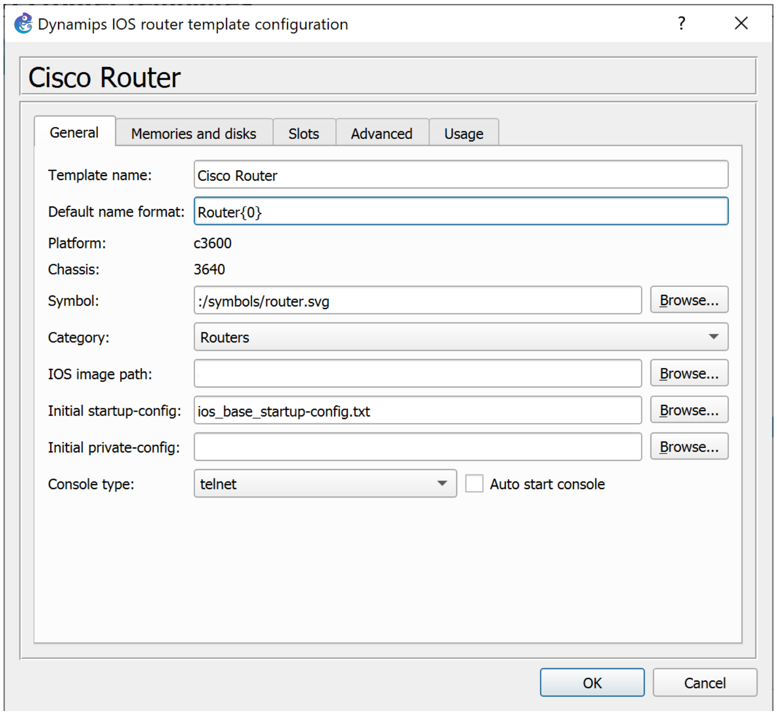
RAM:	128 MiB
NVRAM:	256 KiB
I/O memory:	5%
PCMCIA disk0:	0 MiB
PCMCIA disk1:	0 MiB
Auto delete:	True

Adapters

Slot 0:	NM-1FE-TX
Slot 1:	NM-1FE-TX
Slot 2:	NM-4T

Figure 1.3. Cisco router configuration in GNS3.

Step 4: As a last step, change the default name format of the routers. Select “Edit” in the window in Figure 1.3. You will see the “General” tab of the router configuration. Change the Default name format to “Router{0}” as shown in Figure 1.4 and click “OK.”



Dynamips IOS router template configuration

Cisco Router

General | Memories and disks | Slots | Advanced | Usage

Template name: Cisco Router

Default name format: Router{0}

Platform: c3600

Chassis: 3640

Symbol: :/symbols/router.svg

Category: Routers

IOS image path:

Initial startup-config: ios_base_startup-config.txt

Initial private-config:

Console type: telnet

□ Auto start console

Figure 1.4. Cisco router template configuration (General tab).

Step 5: You again see the window in Figure 1.3. Select “Apply” and then “OK.”

This completes the installation of the Cisco router in GNS3.

Part 2. Configuring and accessing a Cisco Router

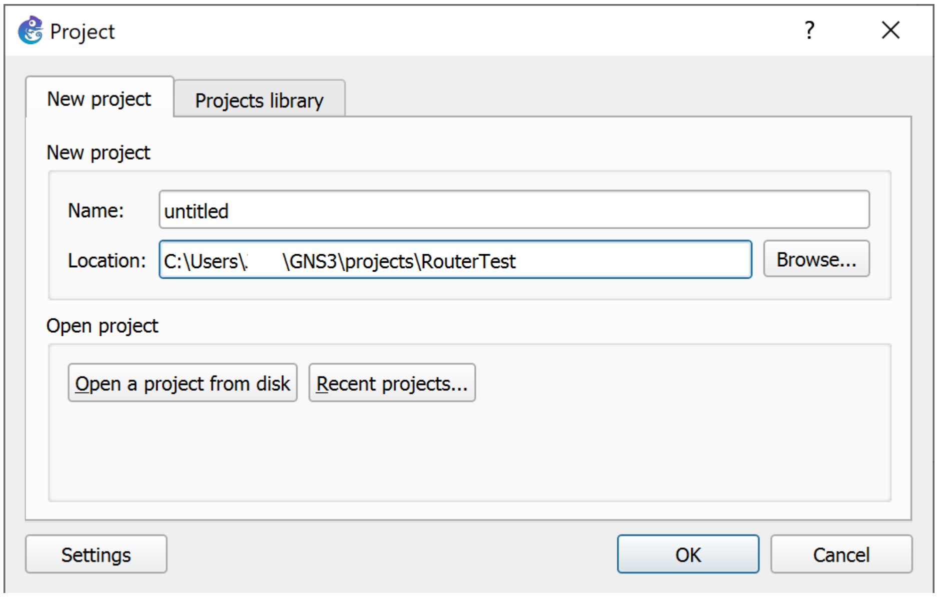
In Part 1, you added a Cisco router to GNS3. Here, you learn how to instantiate such a router.

Exercise 2. Creating and accessing a Cisco router

Step 1: Quit and restart GNS3.

Step 2: If you see a window that asks to create a new project (Figure 1.5) insert a project name (e.g., “RouterTest”) and click “OK.“

Step 10: You can also create a new project in the GNS3 application window (shown in Figure 1.1) by clicking on the folder icon in the upper left-hand corner (📂) or choose “New blank project” by selecting “File→New Blank Project.”



Name: untitled

Location: C:\Users\ \GNS3\projects\RouterTest

Figure 1.5. Creating a GNS3 project.

Step 3: Refer to Figure 1.1. On the icons on the left-hand side, select the second icon from the bottom (“Browse All Devices"), which displays the installed network devices (as shown in Figure 1.6). The icon with label *Cisco Router* refers to the Cisco router image that you installed in Part 1. Select this icon and drag it to the project pane in the center. Then you see the router in the project pane as shown in Figure 1.7.

Numbering Cisco routers:
The router is assigned the label “Router1”. When you create another router by dragging it to the project pane, the router is labeled “Router2”. And so forth.

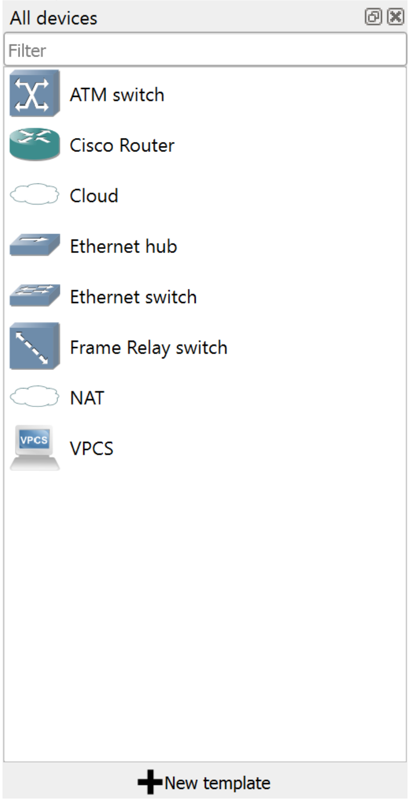
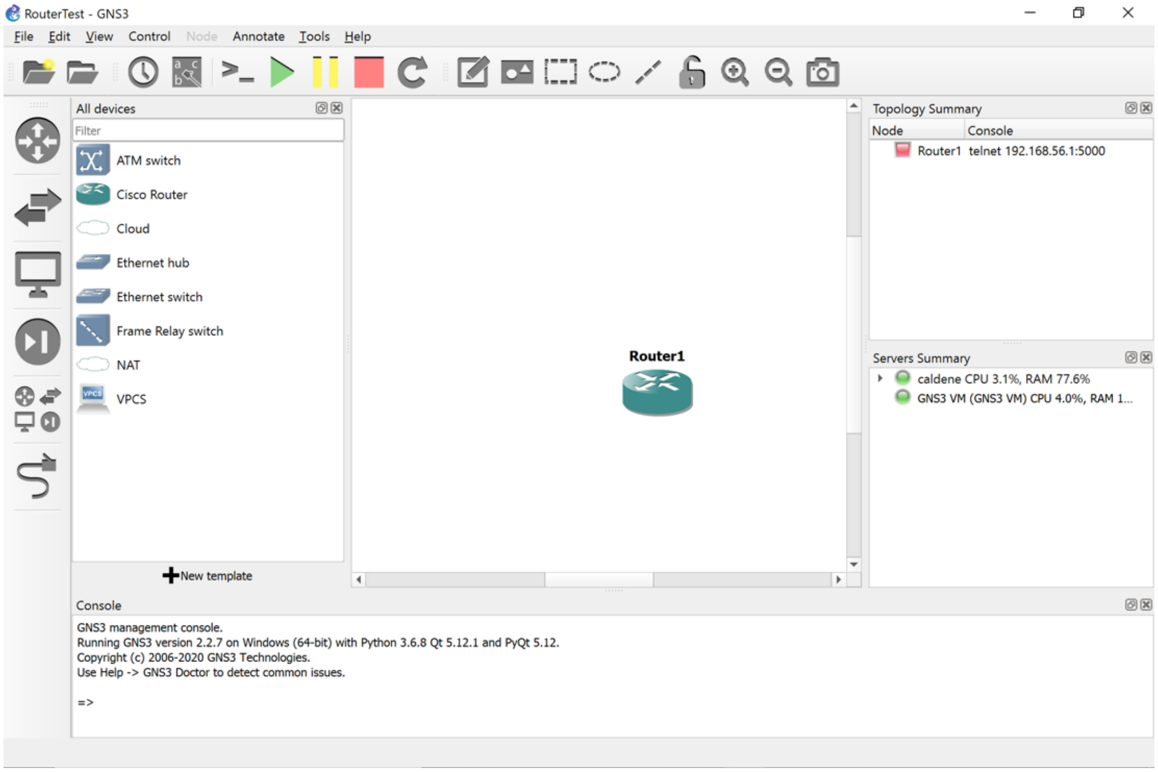


Figure 1.6. Display of installed devices in GNS3.



Node	Console
Router1	telnet 192.168.56.1:5000

```
caldene CPU 3.1%, RAM 77.6%
GNS3 VM (GNS3 VM) CPU 4.0%, RAM 1...
```

```
GNS3 management console.
Running GNS3 version 2.2.7 on Windows (64-bit) with Python 3.6.8 Qt 5.12.1 and PyQt 5.12.
Copyright (c) 2006-2020 GNS3 Technologies.
Use Help -> GNS3 Doctor to detect common issues.

=>
```

Figure 1.7. Adding a router to a GNS3 project.

Step 4: Right-click on *Router1* and select "Start".

Step 5: Right click on *Router1* again and select “Auto Idle-PC”. The system will choose the best value.

Auto Idle-PC

We recommend that you select “Auto Idle-PC” every time you add a router to a project. Also, when you start a saved project, select “Auto Idle-PC” for each Cisco router.

Step 6: Right-click on *Router1* and choose "Console". This opens a console window to *Router1*. Press the Return key a few times. You will see a command prompt (Privileged EXEC prompt) from *Router1*:

```
Router1#
```

When you see this prompt, you can type Cisco IOS commands. If the prompt does not appear, press the Return key another few times.

Step 7: Now you can run commands on the router. To see a list of the available commands, type

```
Router1# ?
```

Run a command that lists the configuration of the router

```
Router1# show running-config
```

Step 8: You can now proceed directly to the next part. If you want to do this at a later time, then (1) halt the execution of Router1 by right-clicking on the router and selecting “Stop”, and (2) exit GNS3 by selecting “File→Quit.”

Autosave
GNS3 auto saves all changes to a project. There is no need to explicitly save a project before exiting GNS3.

Part 3. Setting up a Network Configuration with Routers

In this part of the lab, you will set up a network configuration between routers as shown in Figure 1.8.

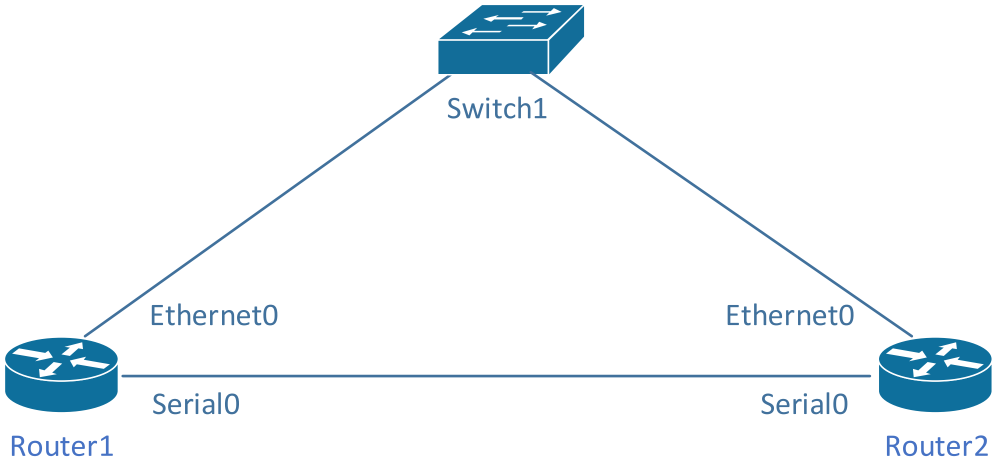


Figure 1.8. Network configuration with Routers.

Exercise 3-a. Connecting two Cisco routers by an Ethernet link

In this exercise, you connect two Cisco routers by an Ethernet link.

Step 1: If you continue directly from Part 2, proceed to Step 2. Otherwise, start GNS3, create a new or open an existing project, and add *Router1* to the project pane.

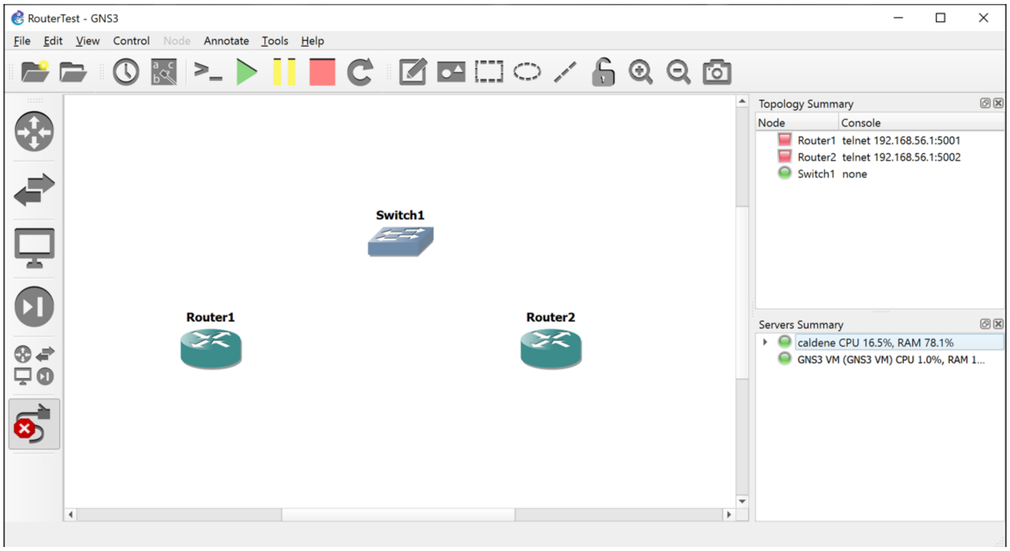
Step 2: Add a second router (*Router2*) to the project following Steps 3–5 from Exercise 2-a.

Step 3: Add an Ethernet switch to the project by selecting and dragging the icon with label “Ethernet switch” into the project pane. If you are queried for a server (where the switch is emulated), select “GNS3 VM”. Arrange the icons as shown in Figure 1.9.

Step 4: In the left pane of the GNS3 window, select the icon at the bottom (“Add a link”). Once selected, the icon changes and shows white `X’ in a red circle (see in Figure 1.9).

Step 5: Click on *Router1* in the project pane. This displays the available network interfaces at Router1 (see Figure 1.10(a)). Select interface “FastEthernet0/0”. Then, click on *Switch1* to display the available network interfaces (see Figure 1.10(b)). Select *Ethernet0*.

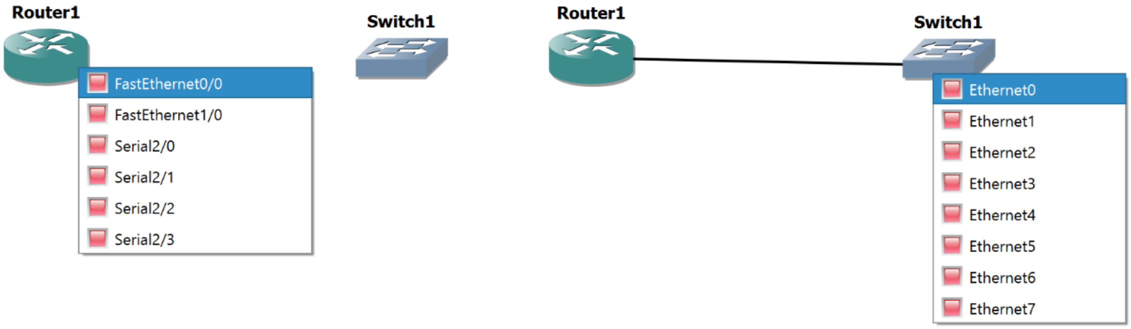
> **“Add a link” is a toggle**
>
> The “Add a link” button is a toggle switch, which puts GNS3 into a mode where links are added. When you are done with adding links, push the button again to disable the mode.



Node	Console
Router1	telnet 192.168.56.1:5001
Router2	telnet 192.168.56.1:5002
Switch1	none

caldene CPU 16.5%, RAM 78.1%
GNS3 VM (GNS3 VM) CPU 1.0%, RAM 1...

Figure 1.9. GNS3 project with two Cisco routers and one Ethernet switch.



(a) Start link at Router1.

(b) Complete link at Switch1.

Figure 1.10. Adding a link in GNS3.

Step 6: Follow the instructions from Step 5 to set up a link between Router2 (FastEthernet0/0) and Switch1 (Ethernet1).

Step 7: Follow Steps 3–5 to set up a link between the *Serial2/0* interfaces of Router1 and Router2. Now the network topology in the GNS3 window looks as shown in Figure 1.8.

Naming conventions of network interfaces

The naming convention for network interfaces of Cisco routers in the lab manual is different from the names used by the Cisco router installed in Part 1 of this lab. In the lab manual, we use *Ethernet0, Ethernet1, …* for the Ethernet interfaces, and *Serial0, Serial1, …* for the serial WAN interfaces. In GNS3, the installed Cisco router uses *FastEthernet0/0, FastEthernet0/1,* … and *Serial2/0, Serial2/1*. The naming conventions are summarized in Table 1.1.

Table 1.1. Naming conventions of router interfaces.

Lab manual		GNS3
Ethernet0	→	*FastEthernet0/0*
Ethernet1	→	*FastEthernet0/1*
Serial0	→	*Serial2/0*
Serial1	→	*Serial2/1*

Deleting a link in GNS3
Delete a link by right-clicking on the link and selecting “Delete” (see Figure 1.11).

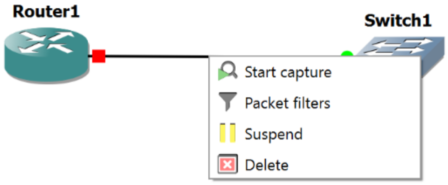


Figure 1.11. Deleting a link in GNS3.

Exercise 3-b. IPv4 configuration

Step 11: The network interfaces of the Cisco routers in the lab do not have pre-configured IP addresses. Here, you run the commands to assign an IPv4 addresses to interfaces of Router1 and Router2 as shown in Table 1.2.

Table 1.2. IP Addresses of routers.

Routers	IP Address of Ethernet0	IP Address of Serial0
Router1	10.0.1.1 / 24	10.0.2.1 / 24
Router2	10.0.1.2 / 24	10.0.2.2 / 24

Commands for IPv4 address configuration
The commands for the IP configuration of routers will be covered in Lab 3, and the IP configuration of PCs is covered in Lab 2. For Lab 1, there is no expectation that you have a grasp of the available commands. Just type the commands as they are given here. If you make a mistake, add a new router and start over.

Step 1: Now start *Router1* and *Router2*. This can be done individually for each router by right-clicking on the icon of the router and selecting “Start”. An alternative is to click on the "Start/Resume all devices" button, which is indicated in the window in Figure 1.9 by a large green triangle toward the top of the window. Clicking this button starts all devices in the project pane.

Step 2: Open console windows for *Router1* and Router*2*. This can be done individually for each router by clicking on the icon of a router and selecting “Console”. If you click on the “Console connect all nodes” button (seen on the left of the green triangle in Figure 1.9) you open consoles for all started devices.

> **Consoles in GNS3 on Windows systems**
>
> - The consoles in GNS3 on a Windows system use the Solar-PuTTY application. By default, multiple consoles are shown as tabs in a single window. To display consoles in separate windows, right-click on a tab for that console and select “Detach.”
> - The default font size of Solar-PuTTY in GNS3 is quite small. Instructions for changing colors and font sizes are given in the video available at this link.

Step 3: In the console of Router1, enter the following commands.

```
Router1# config term
Router1 (config)# interface FastEthernet0/0
Router1 (config-if)# ip address 10.0.1.1 255.255.255.0
Router1 (config-if)# no shutdown
Router1 (config-if)# interface Serial2/0
Router1 (config-if)# ip address 10.0.2.1 255.255.255.0
Router1 (config-if)# no shutdown
Router1 (config-if)# exit
Router1 (config)# exit
```

Run the same commands on Router2, where you use the IPv4 addresses 10.0.1.2 (to replace 10.0.1.1) and 10.0.2.2 (to replace 10.0.2.1)

Step 12: The netmask 255.255.255.0 adds the FastEthernet0/0 interfaces of Router1 and Router2 to the 10.0.1.0/24 subnet and the Serial2/0 interfaces to the 10.0.2.0/24 subnet.

Exercise 3-c. Issuing ping commands

After connecting the Cisco routers and configuring the IP addresses, Router1 and Router2 can communicate with each other. The following steps verify that the two Cisco routers are properly connected and configured. The test consists of running the *ping* command between Router 1 and Router2. A successful *ping* command confirms that a host or router is reachable via the Internet Protocol. The *ping* command sends an ICMP Echo Request datagram to an interface, and expects an ICMP Echo Reply datagram in return.

Ping on Cisco Routers
On a Cisco router, a ping command issues five messages. After the messages are sent, the console displays how many replies where received. Receiving at least one reply makes the ping successful. Usually, the success rate is either 0% or 100%.

Step 1: In the console window of Router1, issue ping commands to Router2. The command

```
Router1# ping 10.0.1.2
```

pings the FastEthernet0/0 interface of Router2, and the command

```
Router1# ping 10.0.2.2
```

pings the Serial2/0 interface of Router2.

Step 2: Take a screen of the console output of the ping commands.

Lab Questions/Report

1. Include the screen capture from Exercise 3-c.

Part 4. Importing a Linux PC into GNS3

The PCs for the labs are based on the Ubuntu Linux distribution and is made available as a Docker container.

Exercise 4-a. Installing an Ubuntu Linux PC

Step 1: Download the configuration file for the Ubuntu Docker container. In a web browser, go to https://raw.githubusercontent.com/Internet-lab/gns3/master/DockerPC.gns3a and save the displayed file as "DockerPC.gns3a" on your computer. (Use copy and paste if your browser does not have a "Save as…" option. Double check that the saved file does not have a ".txt" or similar extension.

Step 2: If GNS3 is not running, start the GNS3 application and wait until the status light of the GNS3 VM (in the Servers Summary pane) has turned green.

Step 3: In GNS3, select "File→Import Appliance".

Step 4: In the window that opens, navigate to the "DockerPC.gns3a" file, which you created in Step 1, and select "Open".

Step 5: In the "Install PC appliance" window, click "Next".

Step 6: In the next window, select "Finish". In the "All devices" pane in the GNS3 application window, you now see a new icon, which is labelled "PC" (see Figure 1.12).

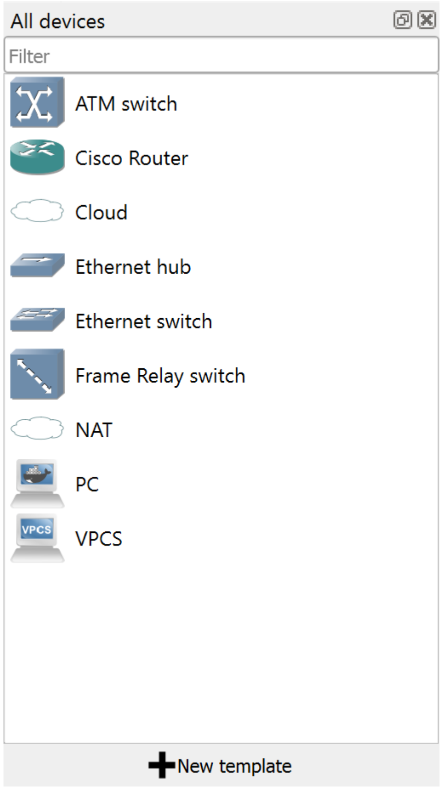


Figure 1.12. Display of installed devices in GNS3.

Exercise 4-b. Adding a Linux PC to a GNS3 project

Step 1: Quit and restart GNS3.

Step 2: Create a new GNS3 project (e.g., “Lab1”) following the instructions from Exercise 2-a, or open an existing project.

Step 3: In the leftmost pane of the GNS3 application window, select the second icon from the bottom (“Browse All Devices"). From the list of displayed devices, select the icon with label “PC” and drag it to the project pane in the center.

> **Numbering PCs**
>
> The first PC created is assigned the label *PC1*. If you add additional PCs, they are labelled *PC2*, *PC3*, and so forth.

Step 4: Right-click on *PC1* and select "Start".
When you start PC1 for the first time after installation, there will be a delay, where the installed Docker container downloads software. After that, adding a PC does not incur this delay.

Step 5: Right-click on *PC1* again and choose "Console". This opens a console window to *PC1*. You see the command prompt

```
PC1$
```

Step 6: Type the command

```
PC1$ ifconfig
```

and save a snapshot of the output. Note that no IP addresses are assigned to interfaces.

Step 7: You can now proceed directly to Part 5.

> **Accessing PCs in future labs**
>
> From now on lab instructions for accessing any PC, e.g. *PC2*, only say
> “On *PC2*, ….”
> Then, follow the instructions as given above to create and access a PC.

Lab Questions/Report

1. Include the screen capture from Exercise 4-b.

2. As much as you can, interpret the output in the screen capture (Limit yourself to 5 observations).

Part 5. Setting up a Network Configuration with PCs

In this part of the lab, you will set up a network configuration as shown in Figure 1.13.

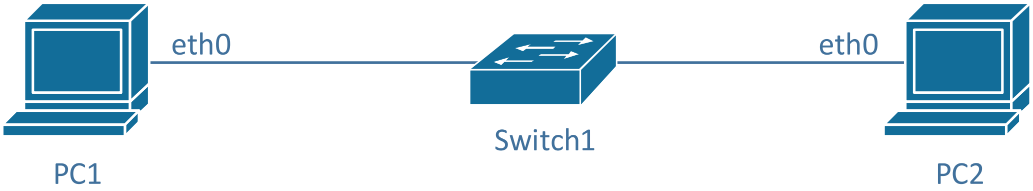
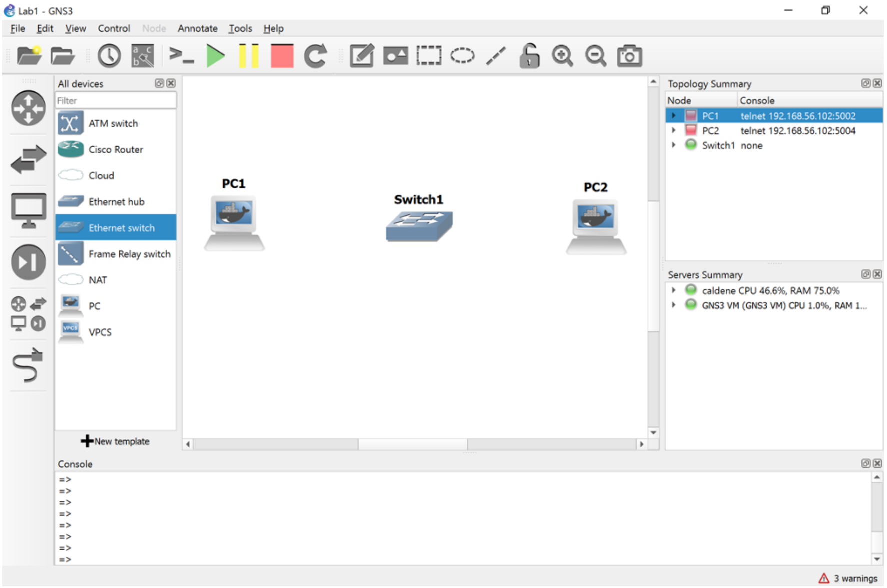


Figure 1.13. Network configuration with PCs.

Exercise 5-a. Creating a network configuration in GNS3

Step 1: If you continue from Part 4, add a second PC to the project pane. Otherwise, add two PCs (PC1 and PC2) to the project pane.

Step 2: Add an Ethernet switch to the project by selecting and dragging the icon with label "Ethernet switch" into the project pane. If you are queried for a server (where the switch is emulated), select GNS3 VM. Arrange the icons as shown in Figure 1.14.



Topology Summary

Node	Console
PC1	telnet 192.168.56.102:5002
PC2	telnet 192.168.56.102:5004
Switch1	none

Servers Summary

caldene CPU 46.6%, RAM 75.0%
GNS3 VM (GNS3 VM) CPU 1.0%, RAM 1...

```
=>
=>
=>
=>
=>
=>
=>
=>
```

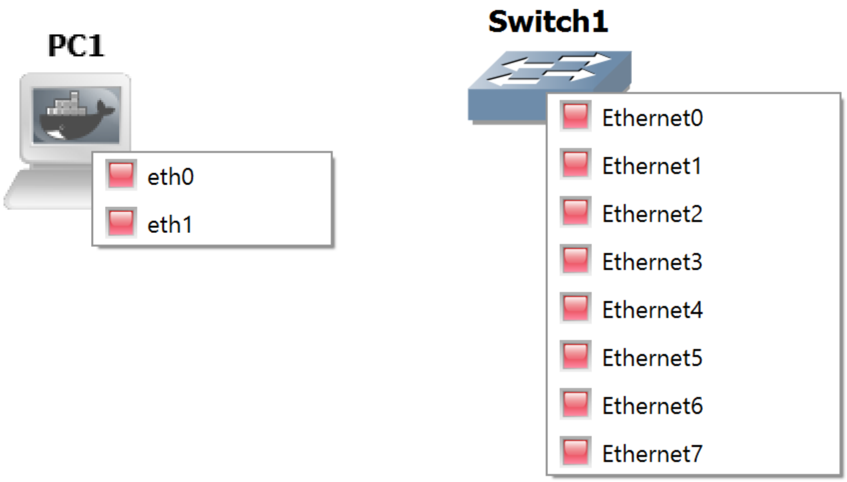
Figure 1.14. Two PCs and an Ethernet hub.

Step 3: In GNS3, add links that connect *PC1* and *PC2* to the Ethernet switch. The PCs have two Ethernet interfaces (eth0, eth1) as shown in Figure 1.15a. The switch has eight interfaces (Ethernet0, …, Ethernet7), as seen in Figure 1.15b. Refer to Steps 3–5 of Exercise 3-a for instructions for creating a link.

- Add a link between the eth0 interface lick on *PC1* and select interface *eth0.* Then, click on *Switch1* and select one of the available interfaces.

- Add a link between the eth0 interface lick on *PC2* and select interface *eth0.* Then, click on *Switch1* and select one of the available interfaces.

The project pane now looks as shown in Figure 1.16. The green and red dots on the links indicate the status of the interfaces. The red dots next to *PC1* and *PC2* show that the interfaces are not active. The dots turn green once you "Start" the PCs. Since Ethernet switches in GNS3 are started when they are created, all interfaces that are attached to a link show a green dot.



PC1

Switch1

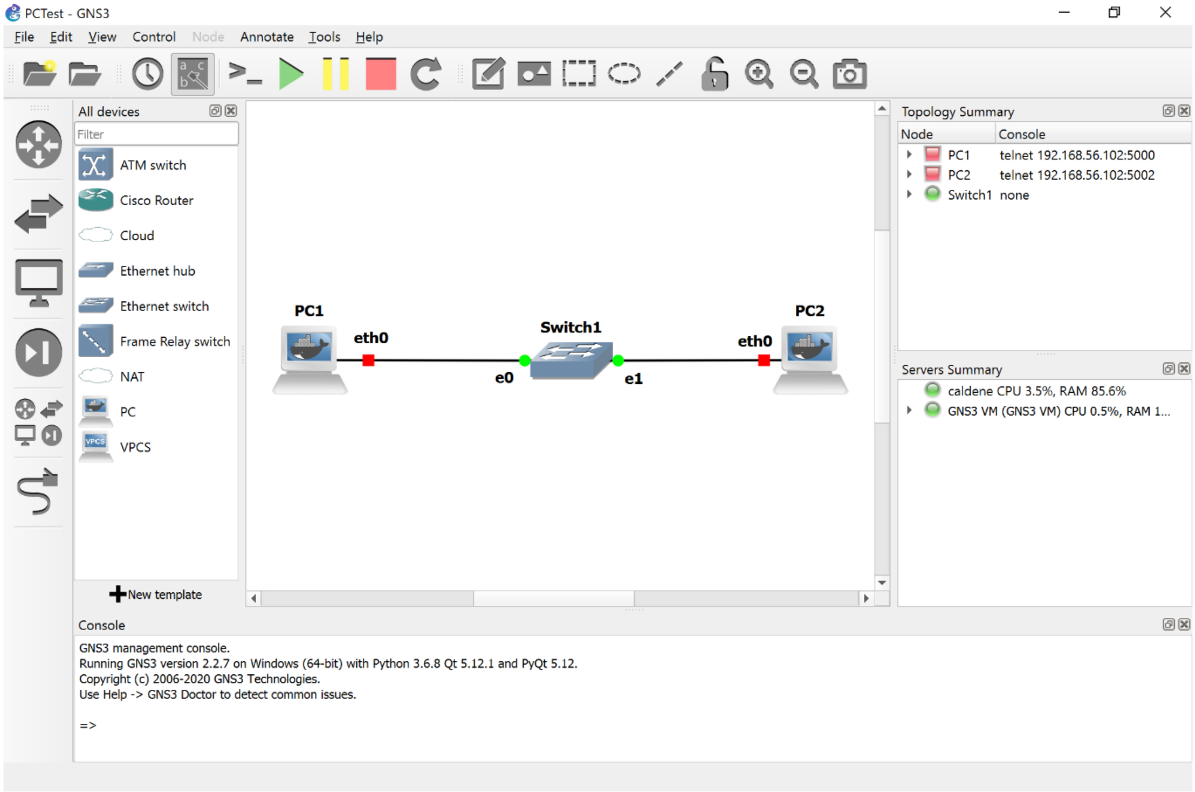
Figure 1.15. Network interfaces of PCs and Ethernet switch.

Step 4: Now start *PC1* and *PC2*. If they are booted up, the red status lights of PC1 and PC2 in the Topology Summary pane (seen in the upper right corner of Figure 1.16) turn green.

Step 5: Open a console for *PC1* and *PC2*.

Displaying connected interfaces

- By selecting "View→Show/Hide interface labels" ( ), the project pane displays the labels of connected interfaces (as in Figure 1.16).



Node	Console
PC1	telnet 192.168.56.102:5000
PC2	telnet 192.168.56.102:5002
Switch1	none

caldene CPU 3.5%, RAM 85.6%
GNS3 VM (GNS3 VM) CPU 0.5%, RAM 1...

```
GNS3 management console.
Running GNS3 version 2.2.7 on Windows (64-bit) with Python 3.6.8 Qt 5.12.1 and PyQt 5.12.
Copyright (c) 2006-2020 GNS3 Technologies.
Use Help -> GNS3 Doctor to detect common issues.

=>
```

Figure 1.16. Two PCs connected by an Ethernet switch.

Exercise 5-b. IPv4 configuration

Step 3: The network interfaces of the PCs in the lab do not have pre-configured IP addresses. You set up an IP address at a PC by typing commands in the console window of that PC. When the PC is rebooted, the configured IP addresses will be lost.

Step 4: The IPv4 addresses of the PCs are shown in Table 1.3. The notation 10.0.1.11/24 means that the IP address is 10.0.1.11 and the network prefix is 24 bits long. A network prefix of 24 bits corresponds to a netmask set to 255.255.255.0. With the above configuration, the eth0 interfaces of both PC1 and PC2 are added to the 10.0.1.0/24 subnet.

Table 1.3. IP Addresses of PCs.

PCs	IP Address of eth0
PC1	10.0.1.11 / 24
PC2	10.0.1.22 / 24

Step 1: Using the console of PC1 (from Step 5 in the last exercise) configure an IPv4 address for the eth0 interface of PC1 by typing

```
PC1$ sudo ip addr flush dev eth0
PC1$ sudo ip addr add 10.0.1.11/24 dev eth0
```

The first command removes any previous IP configuration, and the second command sets the new IP address.

Step 2: Configure the IP address at PC2 (eth0) by entering the following command in the console of PC2:

```
PC2$ sudo ip addr flush dev eth0
PC2$ sudo ip addr add 10.0.1.22/24 dev eth0
```

Exercise 5-c. Issuing ping commands

Next you use the ping command to verify that the IP configuration was successful.

Ping on Ubuntu

- On Ubuntu, ping continues to send packets until you interrupt the command with the `Ctrl-c` key.
- You can limit the number of messages sent with the `-c` option. For example, `ping -c5 10.0.1.22` issues five messages to 10.0.1.22.
- When using ping on the Linux PCs, we recommend to always send at least two ICMP Echo Request packets, since in some occasions, because of delays, the first ICMP Echo Request does not trigger a response by the receiver.

Step 1: On PC1, issue a ping command to PC2 by typing

```
PC1$ ping -c2 10.0.1.22
```

Take a snapshot that shows the output of the command.

Step 2: On PC1, type the command

```
PC1$ ifconfig
```

and save a snapshot of the output. Note the differences to the previous snapshot from Part 4.

Lab Questions/Report

1. Provide the screenshots from Exercise 5-c.
2. Include the observed differences (minimum 2) between the outputs of the `ifconfig` command from Part 4 and Exercise 5-c.

Exercise 5-d. Ethernet connectivity without a switch

Step 1: If you only connect two PCs you do not need an Ethernet switch. You can simply connect two ports directly.

Step 2: Do the following:

a. Delete the links of PC1 and PC2 to the Ethernet switch (by right-clicking on the link and selecting “Delete”).

b. Delete Switch1 (by right-clicking on the switch and selecting “Delete”).

Step 3: Add a link that connects the *eth0* port of PC1 to the *eth0* port of PC2 (using the “Add a link” feature)**.**

Step 4: Repeat Step 1 in Exercise 5-c. You should get the same result.

Part 6. Saving and transferring data

Most lab exercises ask you to save data that is displayed on your computer or data that has been stored in a file. The purpose of this exercise is to become familiar with some methods to save and transfer data.

> **Saved data on PCs are not persistent**
>
> The data that you save on a (Docker) PC in a GNS3 project are lost when you quit GNS3.

Exercise 6-a. Saving command output to a file in Linux

Linux provides an easy way for redirecting the output of a command to a file via the redirection operators > and >>. For example, the command

```
PC1$ ls > fname
```

writes the output of the *ls* command to file *fname*. All previous data in the file is deleted. The command

```
PC1$ ls >> fname
```

appends the output of the *ls* command to file *fname*, without deleting any data. Cisco routers do not have such a command.

Exercise 6-b. Transferring data with copy/paste

Copy/paste to the clipboard is a convenient method for transferring data from a console window of a PC or router in GNS3 to an application (e.g., a word processor) on your computer.

- **Windows PC:** To copy to the clipboard, highlight text and then type *Ctrl- Shift-C* (alternatively, you can simply hit the *Enter* key). To paste from the clipboard, move the mouse at the desired position and right-click. (Note: The usual Ctrl-c and Ctrl-v may not work, since Ctrl-c in Linux is a signal to terminate a running command.)
- **Mac OS:** To copy to the clipboard, highlight text and then type ⌘-*C*. To paste from the clipboard, left click the mouse at the intended position and type ⌘-*V*.

Exercise 6-c. (optional) Connecting a PC in GNS3 to a remote computer

It is possible to connect PCs or routers in a GNS3 project to the Internet or a home network and transfer files to your local computer or a remote computer using the Secure File Transfer Protocol (SFTP). To take advantage of this option, you must have **either** an SFTP server running on your local computer, **or** you must have an account on a remote computer that runs an SFTP server.

The instructions below assume some knowledge of IP configuration issues. Skip the rest of the exercise if you get to a point where you cannot continue.

Step 1:

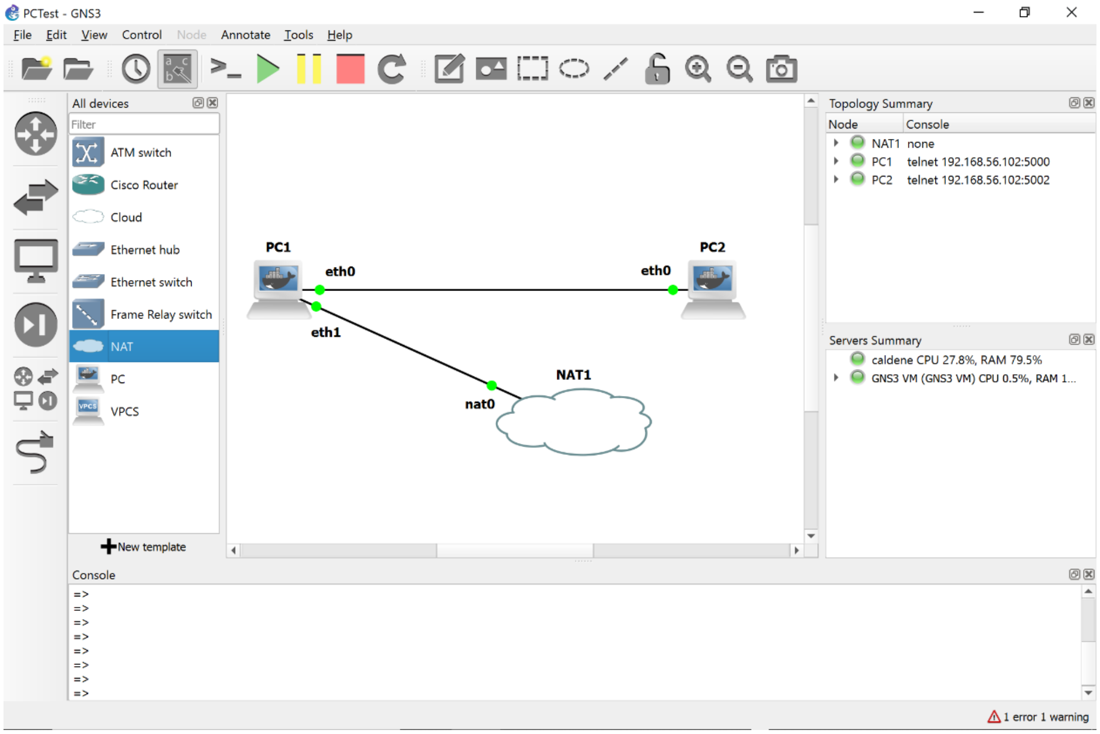
- **On Mac:** Start an SFTP server on your local computer by following the instructions at https://www.maciverse.com/how-to-turn-on-your-macs-sftp.html

- **On Windows:** You must install SFTP software from a third party. There are many free tools that provide an SFTP server on a Windows computer, e.g., the SFTP/SCP server from Solarwinds. Follow the instructions from the chosen provider for downloading and starting an SFTP server.

Step 1: Take note of the IPv4 address of your computer where GNS3 is running. This can be an IP address on your home network.

Step 2: Create a new GNS3 project or open an existing project. If it does not exist, create *PC1*. Start PC1.

Step 3: Drag the cloud with label "NAT" from the "All Devices" pane into the project pane. Connect interface *eth1* of *PC1* to the cloud icon (see Figure 1.17).



Topology Summary

Node	Console
NAT1	none
PC1	telnet 192.168.56.102:5000
PC2	telnet 192.168.56.102:5002

Servers Summary

caldene CPU 27.8%, RAM 79.5%
GNS3 VM (GNS3 VM) CPU 0.5%, RAM 1...

Figure 1.17. Connecting PC1 to a computer outside GNS3.

Step 4: Open a console on *PC1*. Type the command

```
PC1$ sudo dhclient -nw eth1
```

Ignore any warning messages. The command starts a DHCP client on interface *eth1*, which obtains an IP address on the network where your computer is located.

Step 5: Open a console on *PC1*, and start an SFTP client by typing

```
PC1$ sftp myaccount@remoteIP
```

Replace *‘myaccount’* with your login on your computer and replace *`remoteIP’* with an IPv4 address of your computer. You are prompted for your password. If the login is successful, you see the prompt

```
sftp>
```

If your computer is connected to the Internet, you can transfer files to any host on the Internet with an SFTP system where you have a user account. If you have an account on *server.myuniversity.edu* and your login is *myaccount,* type the commands

```
PC1$ echo nameserver 8.8.8.8  > /etc/resolv.conf
PC1$ sftp myaccount@server.myuniversity.edu
```

Step 6: Copy files from PC1 to your computer. Typing

```
sftp> put /etc/hosts
```

transfers the file /etc/hosts on PC1 to the home directory of your computer.

Step 7: Copy files from your computer to PC1. Type

```
sftp> get remotefile
```

to transfer the file *remotefile* (if it exists) from your computer to the the directory where you typed the command in Step 6.

Here are a few useful commands for navigating directories, where `local system’ refers to PC1 and `remote system’ refers to your computer.

- `pwd` Display the current directory on the remote system.
- `!pwd` Display the current directory on the local system.
- `cd` *myremotedir* Change to directory *myremotedir* on the remote system.
- `cd !`*mylocaldir* Change to directory *mylocaldir* on the local system.

Step 8: When you are done exit the SFTP client by typing

```
sftp> quit
```

Part 7. Using the Linux Operating System

Here you explore the Linux system by trying out commands that are typed in a console window.

Exercise 7-a. Linux commands

Step 1: Review the Linux commands discussed in Appendix B. If you are not familiar with Linux or other Unix-like systems, try out some Linux commands by performing the following tasks on PC1

1. Change to the home directory.
2. Create a directory `test` in the home directory.
3. Copy the file `/etc/hosts` to directory `test`.
4. Change to directory `test`.
5. Change the name of file `hosts` to `hostfile`.
6. List the content of directory `test`.
7. List the content of `hostfile`.
8. Remove all files in directory `test.`
9. Remove directory `test`.

Step 2: List the recent commands that you issued on your computer by typing

```
PC1$ history 10
```

If you issued more than 10 commands to complete the list of tasks, replace 10 by a larger number. If you used less, type a smaller number.

Step 3: Take a snapshot that shows the output of the command from the previous step.

Exercise 7-b. The Nano text editor

To modify configuration files on the Linux from a console window, we need a text editor that uses a command line interface. In the labs, we will use the nano text editor. The following is an exercise with the editor.

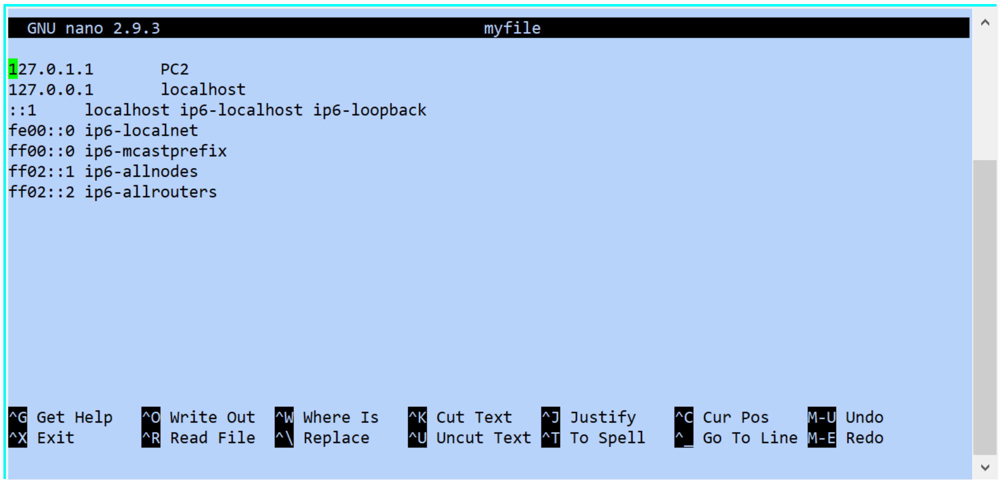
If you are familiar with the text editors *vi* or *vim*, you may use these instead for this exercise.

Step 1: From the home directory, copy the file */etc/hosts* to *myfile*.

Step 2: Open *myfile* with the *nano* text editor by issuing the command

```
PC1$ nano myfile
```

This will display a window as shown in Figure 1.18. The window shows the content of the file. You can make modifications by moving the curser with the arrow keys and typing or deleting content. At the bottom are keyboard shortcuts for more powerful commands. In the appendix you find a list of more shortcuts.



```
  GNU nano 2.9.3                                  myfile

127.0.1.1       PC2
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Figure 1.18. Nano text editor.

Step 3: Edit the content of the file as follows:

- Revert the order of lines (first line becomes last line, 2nd line becomes 2nd line from the bottom, etc.)
- Change all occurrences of “ff” to “gg”.
- Add a new line between lines 5 and 6, where you insert your student ID.

As much as possible, try to use keyboard shortcuts.

Step 4: Take a snapshot of the nano window when you are done.

Step 5: Exit the nano editor by typing Ctrl-X (^X).

Lab Questions/Report

1. Provide the snapshots from Exercise 7-a and Exercise 7-b.

Part 8. Locating Information on the Network Configuration

Linux has a large number of network configuration parameters, that specify IP addresses, host name, whether the system operates as an IP router, and much more. Configuration parameters are stored in the operating system kernel and can be modified by a user with sudo privileges. Kernel parameters can be accessed in different ways. Studying kernel parameters gives insights into the range of network configuration options available to you. Here, you learn about two methods to read and modify kernel parameters. One of them is the command `sysctl` and the other one accesses files in the */proc* directory.

Kernel parameters are initialized through configuration files, which are consulted when the system is booted up. Configuration files in Linux have evolved significantly. What complicates matters is that old versions of configuration files may co-exist with newer versions. This lab will expose you to a structure of configuration files that is not the most recent, but can be managed with moderate effort.

> **Configuration files on the Ubuntu Docker in GNS3**
> The PCs in GNS3 are installed as a Docker container, which resets the network and all other configurations each time the Docker container is started. Hence, changes to any configuration file are lost whenever you start a PC in GNS3.

Exercise 8-a. The `sysctl` command

Step 1: On PC1, list the complete set of kernel parameters by typing the command

```
PC1$ sudo sysctl -a
```

Record the output to a file. Browse the file for parameters whose meaning you can guess.

Step 2: Kernel parameters are organized in a hierarchical fashion with dots separating the hierarchy levels. We are mostly interested in the kernel variables for the network operation, which start with "net." On PC1, list these kernel variables by typing

```
PC1$ sudo sysctl -a | grep net | less
```

The above line has three commands which are linked by pipes ('|'), such that the output of one command becomes the input of the next command. The first command lists the kernel parameters, the second (`grep net`) filters the network relevant parameters, and the third (*less*) lets you navigate the output with arrow and space keys.

Step 3: You can change the value of a kernel parameter using the `-w` option of the `sysctl` command. Next, you use this command to change the host name of PC1 by overwriting the kernel *parameter kernel.hostname*. On PC1, run the commands.

```
PC1$ hostname
PC1$ sudo sysctl -w kernel.hostname=X1
PC1$ hostname
```

The first and third command display the hostname.

Exercise 8-b. The /proc file system

Another method to access the kernel parameters is through the /proc filesystem, which contains information on every detail of the currently running system. Kernel parameters are located in the directory /proc/sys.

Step 1: Set up a network as shown in Figure 1.13, with IPv4 addresses configured as in the Table 1.3 (see Exercise 5-b).

Step 2: Explore the directory */proc/sys/net/ipv4* which contains one file for each kernel parameter relevant to the configuration of IPv4. The name of each file corresponds to a kernel parameter and the contents of the file is the current value.

Step 3: The kernel parameter `icmp_echo_ignore_all` in *directory /proc/sys/net/ipv4* specifies whether a system replies to a ping, or not. A value of `icmp_echo_ignore_all=0` means that the system replies to pings (by sending an ICMP Echo Reply message), and `icmp_echo_ignore_all=1` means that it will not reply.

Step 4: On *PC1*, access the value of the kernel parameter with the command

```
PC1$ cat /proc/sys/net/ipv4/icmp_echo_ignore_all
```

Step 5: The value will be 0.

Step 6: Verify that *PC1* replies to a ping. On *PC2*, issue a ping to *PC1* with

```
PC2$ ping -c 3 10.0.1.11
```

Step 7: Now, change the value of `icmp_echo_ignore` on *PC1*. This can be done by editing the file and changing its content from 0 to 1. A more direct method to write the value 1 in the file is with the command

```
PC1$ sudo sh -c 'echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all'
```

Using sysctl
The same result can be obtained with the following sysctl command

```
PC1$ sudo sysctl -w net.ipv4.icmp_echo_ignore_all=1
```

Step 8: Now go back to *PC2* and issue another ping to *PC1* with the command

```
PC2$ ping -c 10.0.1.11
```

Step 9: Verify that the ping is not successful.

Step 10: On *PC1*, reset the value of *icmp_echo_ignore* to the original value with

```
PC1$ sudo sh -c 'echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_all'
```

Exercise 8-c. Configuration files

The above methods for changing the network configuration are only temporary. The changes disappear when the system is rebooted. Making changes to the network configuration permanent requires modifications of configuration files. When a Linux system is started, it initializes the values of the kernel parameters from configuration files.

The structure of configuration files in Linux has been undergoing many changes. In particular, the most recent method for the configuration of network parameters (using the netplan tool) involves an additional layer of abstraction that is not always intuitive. The configuration files below will generally work, or can be enabled, on recent versions of Linux.

> **Different Linux version, different configuration files**
> Configuration files can vary across different Linux distributions, e.g., Fedora Linux and Debian Linux. Sometimes, the organization of configuration files changes between different releases of the same Linux distribution.

> **Configuration files on Docker**
> The virtual labs use Docker images for the PCs. Here, each time a PC is booted is uses the same initial configuration files of the Docker image. Therefore, changes to the configuration file do not have an effect.

`/etc/sysctl.conf`

This file specifies kernel options, including those related to the network configuration, e.g., whether IPv6 is enabled or not.

`/etc/sysctl.d/`

This is a directory, which contains additional configuration files.

`/etc/hosts`

This file specifies the mapping between the symbolic names and IP addresses for network devices.

`/etc/hostname`

This file specifies the host name of the local system.

Step 1: View the content of the file `/etc/sysctl.conf` on PC1.

Lab Questions/Report

1. Provide the total number of number of kernel parameters saved in Exercise 8-a, as well as the number of kernel parameters related to network configuration.

2. Which kernel parameter determines whether a Linux PC performs IPv4 forwarding?

Part 9. Wireshark

Wireshark is a network protocol analyzer with a graphical user interface. Using *Wireshark*, you can interactively capture and examine network traffic, view summaries and get detailed information for each packet.

- **On Windows PC:** Wireshark is installed during the GNS3 installation process. (If for some reason, you don't have Wireshark installed, you can download it from https://www.wireshark.org/download.html. Follow the installation instructions.

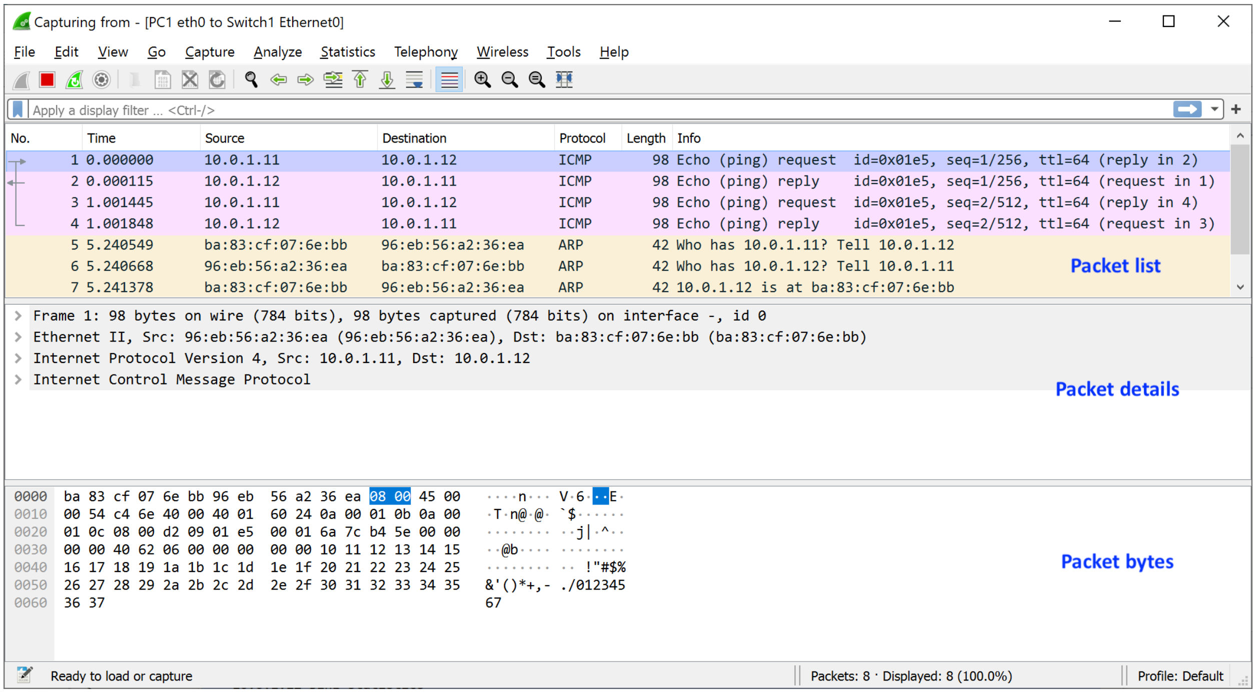
- **On Mac OS:** Wireshark was installed in Lab 0.

Exercise 9-b. Configure and run Wireshark

This exercise walks you through the steps of capturing and saving network traffic with *Wireshark*.

Step 1: Set up a network with two PCs as shown in Figure 1.13, with IPv4 addresses configured as in Table 1.3.

Step 2: **Start Wireshark:** Right-click on the link that connects PC1 and the Ethernet Switch and select "Start capture". Click "OK" in the pop-up window. *Wireshark* opens a new window as shown in Figure 1.19 and starts to capture packets.


Capturing from - [PC1 eth0 to Switch1 Ethernet0]
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Packet list
Packet details
Packet bytes

```
No.  Time      Source             Destination        Protocol Length Info
   1 0.000000  10.0.1.11          10.0.1.12          ICMP       98 Echo (ping) request  id=0x01e5, seq=1/256, ttl=64 (reply in 2)
   2 0.000115  10.0.1.12          10.0.1.11          ICMP       98 Echo (ping) reply    id=0x01e5, seq=1/256, ttl=64 (request in 1)
   3 1.001445  10.0.1.11          10.0.1.12          ICMP       98 Echo (ping) request  id=0x01e5, seq=2/512, ttl=64 (reply in 4)
   4 1.001848  10.0.1.12          10.0.1.11          ICMP       98 Echo (ping) reply    id=0x01e5, seq=2/512, ttl=64 (request in 3)
   5 5.240549  ba:83:cf:07:6e:bb  96:eb:56:a2:36:ea  ARP        42 Who has 10.0.1.11? Tell 10.0.1.12
   6 5.240668  96:eb:56:a2:36:ea  ba:83:cf:07:6e:bb  ARP        42 Who has 10.0.1.12? Tell 10.0.1.11
   7 5.241378  ba:83:cf:07:6e:bb  96:eb:56:a2:36:ea  ARP        42 10.0.1.12 is at ba:83:cf:07:6e:bb
```

```
> Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface -, id 0
> Ethernet II, Src: 96:eb:56:a2:36:ea (96:eb:56:a2:36:ea), Dst: ba:83:cf:07:6e:bb (ba:83:cf:07:6e:bb)
> Internet Protocol Version 4, Src: 10.0.1.11, Dst: 10.0.1.12
> Internet Control Message Protocol
```

```
0000  ba 83 cf 07 6e bb 96 eb  56 a2 36 ea 08 00 45 00   ····n··· V·6···E·
0010  00 54 c4 6e 40 00 40 01  60 24 0a 00 01 0b 0a 00   ·T·n@·@· `$······
0020  01 0c 08 00 d2 09 01 e5  00 01 6a 7c b4 5e 00 00   ········ ··j|·^··
0030  00 00 40 62 06 00 00 00  00 00 10 11 12 13 14 15   ··@b···· ········
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ········ ·· !"#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
0060  36 37                                               67
```

Ready to load or capture | Packets: 8 · Displayed: 8 (100.0%) | Profile: Default

Figure 1.19. *Wireshark* window.

Step 3: **Generate traffic:** In a console on PC1, run a ping command to PC2.

```
PC1$ ping -c2 10.0.1.22
```

Observe the output in the Wireshark window. Take a snapshot of the entire window.

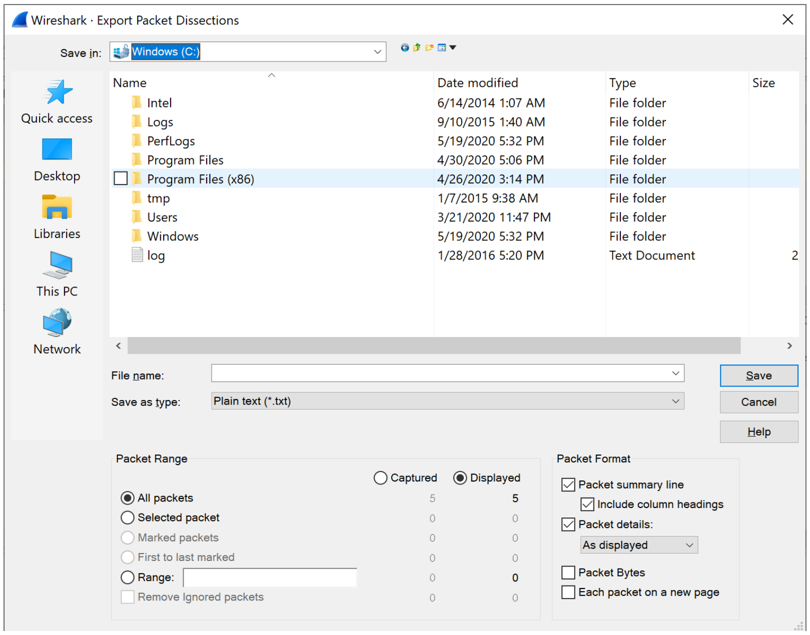
Step 4: Click and highlight a captured ICMP ping packet in the Wireshark window in the "Packet List" pane, and view the headers of the captured packet in the "Packet Detail" pane.

Step 5: **Stop the traffic capture:** In the project pane of GNS3, right-click on the link that connects *PC1* and *Switch1* and select "Stop capture". Alternatively, click on the stop icon (🟥) in the Wireshark window.

Exercise 9-c. Save captured traffic

Here you save the traffic that was captured in the previous exercise to a file. When saving the captured traffic, consider two options:

1. **Plain text file** (*.txt*): Saving captured traffic as a plain text file is useful when you want to include captured traffic in your lab report. Select "File→Export Packet Dissections" and then select "As Plain Text…". This will display a window as shown in Figure 1.20. By default, all displayed packets will be saved with detailed information on each packet. By unselecting "Packet details", only a summary line is saved for each packet.
2. **Packet capture format (***.pcapng***)**: When saving data in this format to a file, you can open the file with *Wireshark* for further analysis. To save packets in this format select "File→Save As…" and select the *pcapng* format. To open a file in packet capture format, select "File→Open" in *Wireshark*.



Save in: Windows (C:)

Name	Date modified	Type	Size
Intel	6/14/2014 1:07 AM	File folder	
Logs	9/10/2015 1:40 AM	File folder	
PerfLogs	5/19/2020 5:32 PM	File folder	
Program Files	4/30/2020 5:06 PM	File folder	
Program Files (x86)	4/26/2020 3:14 PM	File folder	
tmp	1/7/2015 9:38 AM	File folder	
Users	3/21/2020 11:47 PM	File folder	
Windows	5/19/2020 5:32 PM	File folder	
log	1/28/2016 5:20 PM	Text Document	2

File name:

Save as type: Plain text (*.txt)

Packet Range

	○ Captured	● Displayed
● All packets	5	5
○ Selected packet	0	0
○ Marked packets	0	0
○ First to last marked	0	0
○ Range:	0	0
□ Remove Ignored packets	0	0

Packet Format

☑ Packet summary line
☑ Include column headings
☑ Packet details:
As displayed
□ Packet Bytes
□ Each packet on a new page

Figure 1.20. Saving captured traffic as plain text file in *Wireshark*.

Saving plain text file on Mac
On some versions (not the most recent) of Wireshark on a Mac, the “Export Packet Dissections” and/or available file formats may be greyed out. If this is the case, go to the Wireshark window, write “IP” in the text field that displays “Apply a display filter …”, and enter the return key. Then, delete the typed text. After this, the greyed out parts seen earlier no longer show.

Saving Wireshark data:
Unless specifically requested to do so, do not include “Packet details” in lab reports. This helps with keeping the length of the lab reports reasonably small. If detailed information is required, you will be asked to save details of the captured traffic.

Step 1: **Save captured traffic**: Save the captured traffic from Step 4 of Exercise 9-b as a plain text file (*.txt*) twice, once with and once without “Packet details” selected. Inspect the two files to get a sense of the information saved. Then, save the captured traffic in packet capture format **(***.pcapng***)**.

Step 2: **Open a .pcapng file**: Quit and re-start *Wireshark*. Open the packet capture format file from Step 1. Verify that Wireshark displays the complete set of captured packets.

Lab Questions/Report

1. Provide the saved plain text file from Step 5, the one without the “Packet details”.

Using Wireshark in future labs

- In the following labs, instructions for capturing traffic will read *“Capture the traffic between PC1 and PC2 with Wireshark.”* With these instructions, you need to run Wireshark on either PC1 or PC2 using the instructions above.
- All future labs require that you know how to start and stop a traffic capture between PCs and/or routers, and that you know how to save traffic captures in *Wireshark*.

Exercise 9-d. Display filters in Wireshark

Display filters are used to display selected subsets of captured traffic. Refer to Appendix A for the syntax of display filters. Detailed information on filter capabilities and options of Wireshark is available in help tab in Wireshark (Select “Help→Manual Pages→Wireshark Filter).

Step 1: Start the Wireshark application on your computer, outside of the GNS3 application.

- **On Mac:** Go to the application folder and double-click on the icon of the Wireshark application ( ).
- **On Windows:** Go to the Start menu and locate the Wireshark application. Open it.

Step 2: In the window entitled "Welcome to Wireshark", select the WiFi interface of your computer. Then click on the icon ( ) to start capturing packets.

Step 3: Start a web browser and visit some webpage. After a minute, stop the traffic capture (🟥) in the *Wireshark* window.

Step 4: You will see that Wireshark has captured many packets from a large variety of applications and using different protocols. Here, display filters help with reducing the set of displayed packets, when you are only interested certain types of packets or protocols.

Step 5: **Set a display filter**: Display filters are typed in the text field below the row of icons (with initial display "Apply a display filter"). The syntax for display filters and examples are given in Appendix A. Clicking on the ribbon symbol (🔖) which is located next to the text field shows samples of display filters. Use display filters to selectively display the following types of packets:

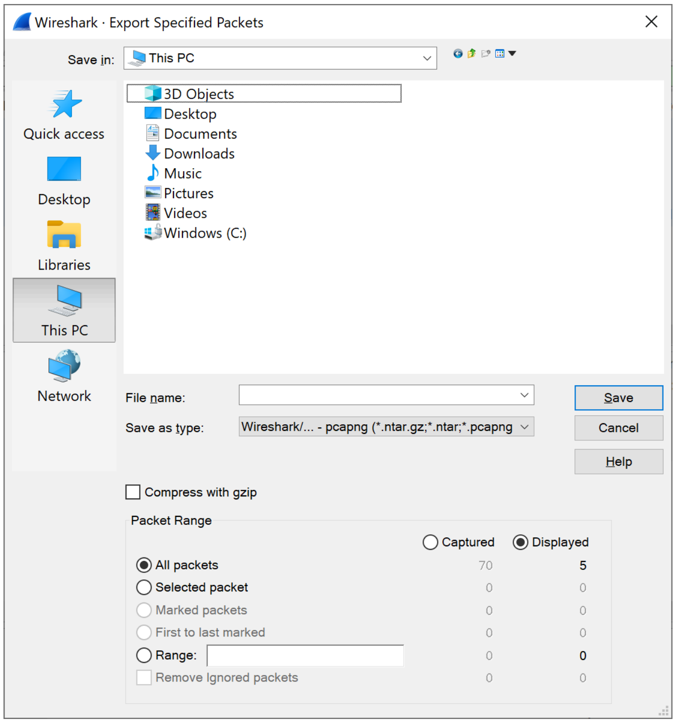
- Only ARP packets,
- only IPv6 packets,
- all packets of the DNS protocol,
- all TCP packets that arrive from or go to TCP port 80.
- all packets that are arrive at the IPv4 address of the WIFI interface of your computer.

Step 6: **Save filtered traffic**: Stop the traffic capture. Set the display filter to display only packets of the ARP protocol.

Step 7: **Save plain text file:** Select "File→ Export Packet Dissections" and then select "As Plain Text…". This displays a window as shown in Figure 1.20. Unselect "Packet details" and save the data.

Step 8: **Save *pcapng* file:** Select "File → Export Specified Packets". This displays a window as shown in Figure 1.21. Select ".pcapng" as file type and save to a file.

Step 9: Verify that the correct data has been saved by opening the .txt file with your favorite application, and opening the .pcapng file in Wireshark.



Save in: This PC

File name:

Save as type: Wireshark/... - pcapng (*.ntar.gz;*.ntar;*.pcapng)

☐ Compress with gzip

Packet Range

	○ Captured	● Displayed
● All packets	70	5
○ Selected packet	0	0
○ Marked packets	0	0
○ First to last marked	0	0
○ Range:	0	0
☐ Remove Ignored packets	0	0

Figure 1.21. Saving filtered traffic as .pcapng file in *Wireshark*.

Part 10. A Network Configuration with PCs and Cisco routers

This is the grand finale of Lab 1, where you put all pieces of this lab together. You set up a topology with routers and PCs, configure IP addresses, and capture traffic in the topology with Wireshark.

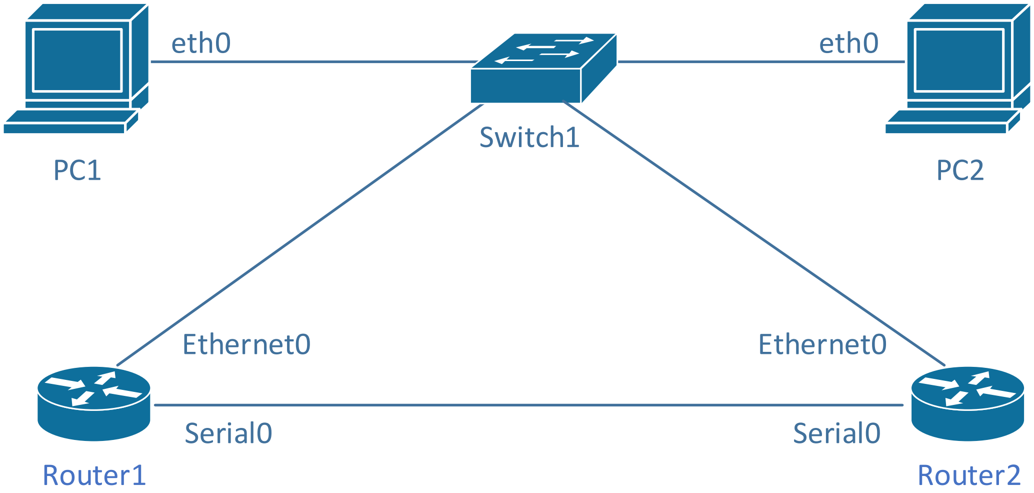


Figure 1.22. Network configuration for the grand finale of Lab 1.

This part of the lab serves as a check point if you are ready to move on to Lab 2. The instructions in the following exercises use the same level of detail (or lack thereof) that you will find in the upcoming labs. If you get stuck with some instructions, e.g., how to configure an IP address, refer to the detailed instructions in earlier parts of this lab.

Exercise 10. Creating and testing a network configuration in GNS3

Step 1: **Setup of topology**. Create a network topology as shown in Figure 1.22, where two PCs (PC1, PC2) and two routers (Router1, Rotuer2) are connected to an Ethernet switch. Also, the routers are connected by a serial WAN link.

Step 2: **Configure IPv4 addresses**. Configure the IP addresses of the Ethernet0 and Serial0 interfaces of Router1 and Router2 as shown in Table 1.2 and configure the eth0 interfaces of the PCs as shown in Table 1.3.

Step 3: **Start traffic capture:** Start a traffic capture for the traffic between PC1 (eth0) and the Ethernet switch.

Step 4: **Generate traffic:** Open a console window on PC1 and issue ping commands to the other devices in the topology:

```
PC1:~# ping -c2 10.0.1.22
PC1:~# ping -c2 10.0.1.1
PC1:~# ping -c2 10.0.1.2
PC1:~# ping -c2 10.0.2.1
```

All ping commands, except the last one, should be successful and you should observe the traffic between PC1 and the other devices. Why does the last ping fail?

Step 5: On Router 1, additionally perform the command

```
Router1# ping 10.0.2.2
```

Step 6: Take a snapshot of the pane in Wireshark that shows all captured packets from the packet list pane (see Figure 1.22). Enlarge the pane as necessary to get all packets that are captured.

Lab Questions/Report

1. Include the snapshot from Step 6.
2. Why does the last ping in Step 4 fail?

Appendix A: Display Filter Expressions in Wireshark

The following tables describe frequently used display filter expressions in *Wireshark*. If you need additional expressions, go to “Analyze Display → Filter Expressions”.

`ip.dst==10.0.1.2`	IP destination address field is 10.0.1.2
`ip.src==10.0.1.2`	IP source address field is 10.0.1.2
`ip.addr==10.0.1.2`	IP source or destination address field is 10.0.1.2
`ip.src==10.0.1.0/24`	IP source address matches the network address 10.0.1.0/24
`ip.dst==10.0.1.0/24`	IP destination address matches the network address 10.0.1.0/24
`ip.addr== 10.0.1.0/24`	IP source or destination address matches the network address 10.0.1.0/24

`tcp.dstport == 80 or`
`udp.dstport == 80`
	Destination port is 80 in TCP segment or UDP datagram

`tcp.srcport==80 or`
`udp.srcport==80`
	Source port is 80 in TCP segment or UDP datagram

`tcp.port==80 or udp.port==80`	Destination or source port is 80 in TCP segment or UDP datagram

`(tcp.srcport==80 and`
`tcp.dstport==80) or`
`(udp.srcport==80 and`
`udp.dstport==80)`
	Destination and source port is 80 in TCP segment or UDP datagram

`tcp.port==80`
`udp.port==80`
	Destination or source port is 80 in TCP segment
	Destination or source port is 80 in UDP datagram

`eth.len <= 200`	Packet size is not longer than 200 bytes

`icmp`
`tcp`
`udp`
`ospf`
	IP protocol field is either set to the number for ICMP or TCP or UDP or OSPF (see example below) or one can use the protocol name. (Since icmp, tcp, and udp are keywords in *tcpdump,* therefore an escape character (‘\’) must be placed in front of these keywords.)

`ip.proto==17`	IP protocol number is set to 17
`eth.dst==ff:ff:ff:ff:ff:ff`	Ethernet broadcast packet
`eth.dst[0]==1`	Ethernet multicast packet
`ip.dst==224.0.0.0/4`	IP multicast packet.
`ip` `arp`	Ethernet payload is IP or ARP

Table 1.4. Display filter expressions in *Wireshark*.

tcp[0] > 4	The first byte of the TCP header is greater than 4
ip[2] <= f	The third byte of the IP header does not exceed 15
udp[0:2] == 3:ff	The first two bytes of the UDP header are equal to 1023 Note: When selecting bytes from a header, each byte is written as a hexadecimal number, and bytes are separated by a colon
ip.hdr_len > 20	IP headers that are longer than 20 bytes, i.e., IP headers with options
ip.frag_offset == 0	IP packets where the fragment offset field is zero, i.e., unfragmented IP packets or the first fragment of a fragmented IP packet
tcp.flags.syn==1 or tcp.flags.fin==1	TCP headers with the SYN flag or the FIN flag set

Table 1.5. More display filter expressions in *Wireshark*.

Appendix B: Linux File System and Commands

The Linux File System

Like most operating systems, Linux organizes files as a hierarchical tree of directories. **Error! Reference source not found.** shows a snapshot of the directory hierarchy of Linux. The directory at the top of the hierarchy, denoted by `/', is called the *root directory*.

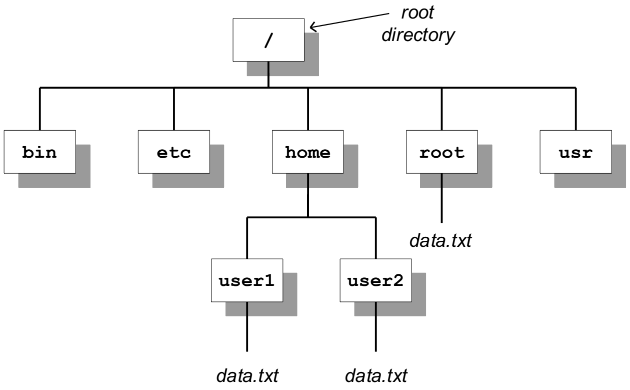


Figure 1.23. Linux directory hierarchy.

Pathnames: Each file and directory in a Linux file system is uniquely identified by a *pathname*. Pathnames can be absolute or relative. Absolute pathnames start at the root directory. The absolute pathname of the root directory is a slash (*/*). In the file hierarchy in Figure 1.23, the absolute pathname of directory *home* in the root directory is */home*, that of directory *user1* in */home* is */home/user1*, and the absolute pathname of file *data.txt* in */home/user1* is */home/user1/data.txt*.

Pathnames that do not begin with a slash are relative pathnames and are interpreted relative to a *current (working) directory*. For example, if the current directory is */home*, then the pathname *user1/data.txt* refers to the absolute pathname */home/user1/data.txt*.

When using relative pathnames, a single dot (.) denotes the *current directory* and two dots (..) denote the *parent directory*, which is the directory immediately above the current directory. With the parent directory, it is feasible to identify each file with relative pathnames. In Figure 1.23, if the current directory is */home/user1*, the relative pathname .. refers to directory */home,* the pathname *../..* refers to the root directory, and the pathname *../user2/data.txt* refers to the file */home/user2/data.txt.*

User accounts: In addition to regular user accounts, Linux systems have an administrator account, called *root*. Changes to the network configuration and other system wide settings require root privileges. In recent years, the use of the root account has been discouraged. Instead, regular user accounts can be assigned administrator privileges by adding the user to the *sudo* group. In the lab, the PCs have the account with name "labuser", which has been added to the *sudo* group.

Home directories: Each Linux account – with the exception of the *root* account – has a *home directory*, that is located in */home*. So, */home/user1* is the home directory for an account with login *user1*. The home directory of the root account is */root*. When a new terminal window is created, the current directory in the terminal window is the home directory.

File permissions: Each file and each directory has an owner. A regular user only owns the home directory and all files created by the user. The root is the owner of all other files on the system. In Linux, each file has a set of access permissions. The permissions are *read ("r"), write ("w"),* and *execute ("x"),* and give, respectively, permission to read the contents of a file, modify the file, or execute the file as a program. Permissions are set by the owner of a file. Linux specifies access permissions separately for the owner of the file, a user group which is associated the file, and the set of all users. So, the owner of a file can set the permissions so that all users can read the files, but only the owner can modify the file. The root and *sudo* users can ignore access permissions and can even change the ownership of files. Since the exercises in the Internet Lab are done from the *labuser* account, which has sudo privileges, access permissions are not important for the Internet Lab. The downside of not having to worry about access permissions is that there is no protection against accidentally deleting or corrupting files.

Linux Devices and Network Interfaces

The software abstraction through which the Linux kernel accesses networking hardware is that of a *network interface.* For example, when assigning an IP address to an Ethernet interface cards, one manipulates the configuration parameters of the network interface which represents the Ethernet card. In the Internet Lab, the names of network interfaces for are *eth0* for the first Ethernet interface card and *eth1* for the second Ethernet interface card. There is a special network interface, called *loopback interface*, with name *lo*. The loopback interface is not connected to a real device, but is a virtual interface, which allows a Linux system to send messages to itself.

Linux shell and commands

The command line interface of the Linux operating system is called a *shell.* A shell is a program that interprets and executes Linux commands which are typed in a window terminal. Whenever you create a new terminal window, a shell is started. The shell displays a prompt at which the user can type commands. The prompt can be as simple as

```
%
```

or the prompt can be set to provide additional information. For example, in the terminal in **Error! Reference source not found.**, the prompt

```
39: PC1@/root =>
```

displays the name of the computer and the current directory. Throughout this book, we use the prompts "%", or "PC1%" if we want to indicate that this is a shell prompt at PC1. When you type a command at the prompt, and press the enter key, the shell interprets the command, and, if it is a valid Linux command, executes the command. A shell is terminated by typing *exit* at the command prompt. If the shell is running in a terminal window, the terminal window disappears.

Next, we review some basic Linux commands that are typed in at a shell prompt. Commands in Linux have a common format: a command name, which may be followed by a set of options and arguments. For example, in the command `*ls –l data.txt*', `*ls*' is the command, `*-l*' is an option which further specifies the command, and `*data.txt'* is an argument. Options are generally preceded by a **–** (dash), and multiple options can be specified in a single command.

Since all files in Linux are organized as a tree of directories, you need to become familiar with navigating and manipulating the directory tree. The following are commands that relate to Linux directories.

Directory commands:

pwd

Prints the absolute path of the current directory.

cd *dirpath*
cd

Changes the current directory to the relative or absolute pathname of the directory *dirpath*. If no directory is given, the command changes the current directory to the home directory. For example, the command **'cd /usr/bin'** changes to directory /usr/bin, the command `cd ..' changes to the parent directory, and the command `cd' without a parameter changes to the home directory.

mkdir *dirname*

Creates a new directory with name *dirname* in the current directory. For example, the command `mkdir xyz' creates a subdirectory in the current directory with name *xyz*.

rmdir *dirname*

Deletes the directory *dirname* from the current directory. A directory cannot be deleted when it still contains files or subdirectories. Thus, before deleting a directory, all files and subdirectories must be deleted first.

Before discussing the commands to list and manipulate files, we introduce the *wildcard characters* * (star) and ? (question mark). The wildcard character * matches any sequence of zero or more characters, and **'?'** matches any single character. Wildcard characters are useful to describe multiple files in a concise manner. For example, the text string *A*.txt* matches all file names that start with an *`A'* and end with *`.txt',* e.g., *ABC.txt, A.txt,* and *Ab.txt* The text string *A?.txt* matches all filenames that are two characters long and start with *`A',* e.g., *Ab.txt* and *A1.txt.*

File commands:

ls
ls *dirname*

Lists information about files and directories in the current directory. If the command has a directory name as argument, then the command lists the files in that directory. The ls command has several options. The most important is `ls -l', which includes extensive information on each file, including, the access permissions, owner, file size, and the time when the file was last modified.
For example, `

- 'ls /' lists all files and directories in the root directory;
- `ls AB*' lists all files and directories in the current directory that start with AB;

- `ls -l ..’ prints detailed information on each file and directory in the parent directory of the current directory.

mv *fname newfile*
mv *fname dirname*

The first command renames a file or directory with name *fname* as *newfile*. If the destination file (*newfile*) exists, then the content of the file is overwritten, and the old content of *newfile* is lost. The second command moves a file or directory with name *fname* to directory *dirname*.
For example,

- `mv data.txt text.txt’ simply renames file data.txt, and
- `mv * /home/labuser’ moves all files from the current directory to directory /root (and gives an error message if the current directory is /home/labuser).

cp *fname newfile*
cp *fname* dirname

Copies the content of file *fname* to *newfile*. If a file with name *newfile* exists, the content of that file is overwritten. If the second argument is a directory, then a copy of *fname* is created in directory *dirname*.
For example,

- `cp *.txt /tmp’ creates a copy of all files that end with `.txt’ in directory /tmp.

rm *fname*

Removes a file. Once removed, the file cannot be recovered.
For example,

- `rm *’ removes all files in the current directory.

Prevent overwriting files
By default, Linux does not issue a warning when a file is overwritten or when a file is removed. If add the option `-i’, Linux asks for confirmation before overwriting or deleting files.

Examples:

```
$ cp -i file1 file2
```

gives a warning if file2 already exists and request confirmation to overwrite its content.

```
$ rm -i file1
```

requests to confirm deletion of file *file1*.

An important thing to have in mind is that Linux does not have an undo command that reverses the effects of a previously issued command.

In many lab exercises you need to modify the content of configuration files. For modifying files, the following commands are helpful.

On the Linux PCs used on GNS3, we recommend the editor *nano*. There is an exercise with *nano* in Part 7 of Lab 1. To edit the file /etc/hosts with *nano*, simply type

```
PC1$ nano /etc/hosts
```

To modify the file simply click on a location in the text window and type text. To save the changes, press Ctrl-x.

Many lab experiments ask you to save the output of a command to a file. The following commands show how to redirect the output of a terminal window to a file.

Redirecting the output of programs:

cmd > fname

The output of the command *cmd* is written to file *fname*. The file is created if it doesn’t already exist, and its contents is overwritten if the file exists.
For example,

- The command `ls > mlist.txt’ writes a listing of the current directory into file *mylist.txt*.

cmd >> fname

The >> operator appends the output of command *cmd* to the end of file *fname*.
For example,

- The command, `ls >> mylist.txt’ appends a listing of the current directory to file mylist.txt.

In Linux, each console window can run multiple commands at the same time. Also, it is possible to stop a command temporarily and resume it at a later time. In each terminal window, one command can be run as a *foreground process* and multiple command can be run as *background processes*. When a command is issued from the prompt, say

```
PC1$ nano
```

the command nano is started in the foreground. When a command is running in the foreground, no shell prompt is displayed until the command is finished. The same command can be run in the *background* by adding an & (ampersand) at the end of the command, by typing

```
PC1$ nano &
```

If a command is executed in the background, the shell prints a prompt for the next command without any delay. Using background commands, you can run multiple commands from a single terminal window.

You can switch a command that is running in the foreground to the background and vice-versa. Switching a command from the foreground to the background is done as follows:

```
PC1$ nano
```

Then type `Ctrl-z` to stop the editor from running. You will see the command prompt of the shell. If you then type

```
PC1$ bg
```

the nano command resumes in the background. To switch a command from the background to the foreground, type

```
PC1$ jobs
```

The command *jobs* lists all commands that are currently running in the background or are stopped, e.g., with `Ctrl-z`. The command

```
PC1$ jobs
```

resumes the first command from the list in the foreground. The following are a set of Linux commands that control the execution of commands.

Control of commands:

`Ctrl-c`

Pressing *Ctrl-c* terminates the command running in the foreground.

`Ctrl-z`

Pressing the *Ctrl-z* stops the command running in the foreground.

cmd **&**

Executes the command *cmd* in the background.

`jobs`

Lists all background and stopped commands of the current user, and assigns a number to each command.

`fg` ***%n***
`fg`

Resumes the n-th command of the user (as listed by the command jobs). If no number is given, the command refers to the command that was last running, started, or stopped.

`bg` ***%n***
`bg`

Resumes the n-th command of the user that is stopped or running in the background. If no number is given the command refers to the command that was last running, started, or stopped.

`kill` ***`%n`***

Terminates the *n*-th command (listed by `jobs’).