Department of Computer Science and Engineering

Islamic University of Technology (IUT)

A subsidiary organ of OIC

Laboratory Report

CSE 4616: Wireless Networks Lab

Name: Md Farhan Ishmam
Student ID: 180041120
Section: CSE-1
Semester: Sixth
Academic Year: 2021

Date of Submission: 06-Dec-21

Title: Introduction to GNS3 and Wireshark

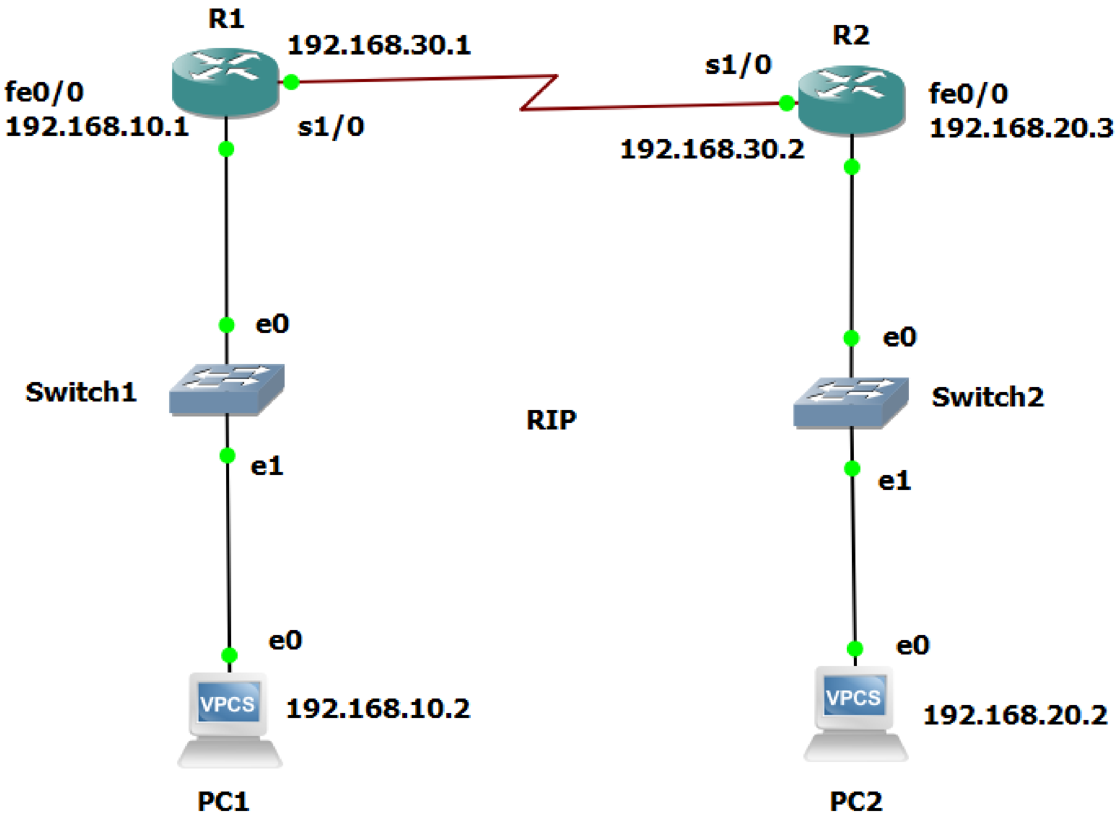
Objective:

1. Installing GNS3 client on windows
2. Adding, installing and importing Cisco IOS in GNS3
3. Introduction to Wireshark
4. Configuring RIP in a simple network topology using GNS3

Devices/ software Used:

Device: Windows PC
Software: GNS3 2.2.27.0

Diagram of the experiment:



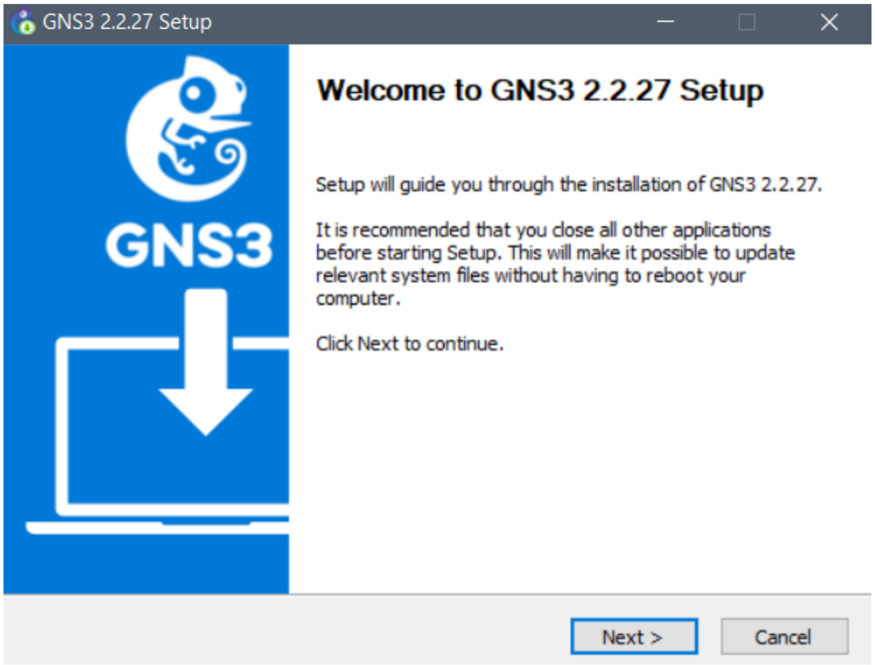
Working Procedure:

Step 1: Installing GNS3

- As per instructions in the lab manual, we install GNS3 by downloading it from

  https://www.gns3.com/software/download

- Then by double clicking on the installer, we go through the necessary steps of the setup. The screenshot of the installer is shown below:



Welcome to GNS3 2.2.27 Setup

Setup will guide you through the installation of GNS3 2.2.27.

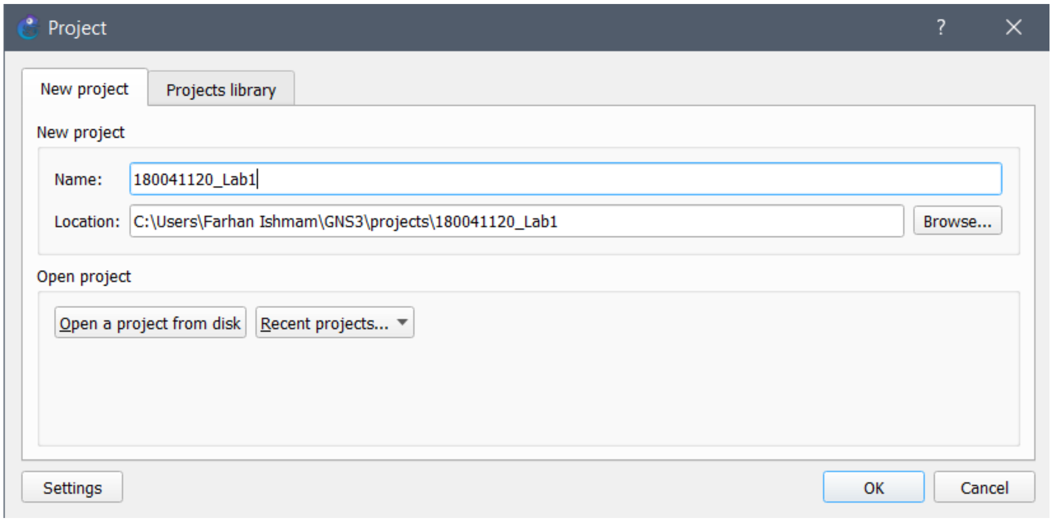
It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

- Finally, we run the GNS3 executable file after the installation is complete by double clicking on the desktop shortcut.

Step 2: Creating a project and loading IOS

- To create a new project we go to File> New Blank Project. Then after renaming the project and selecting the location, we click on ok. The screenshot is given below:



New project

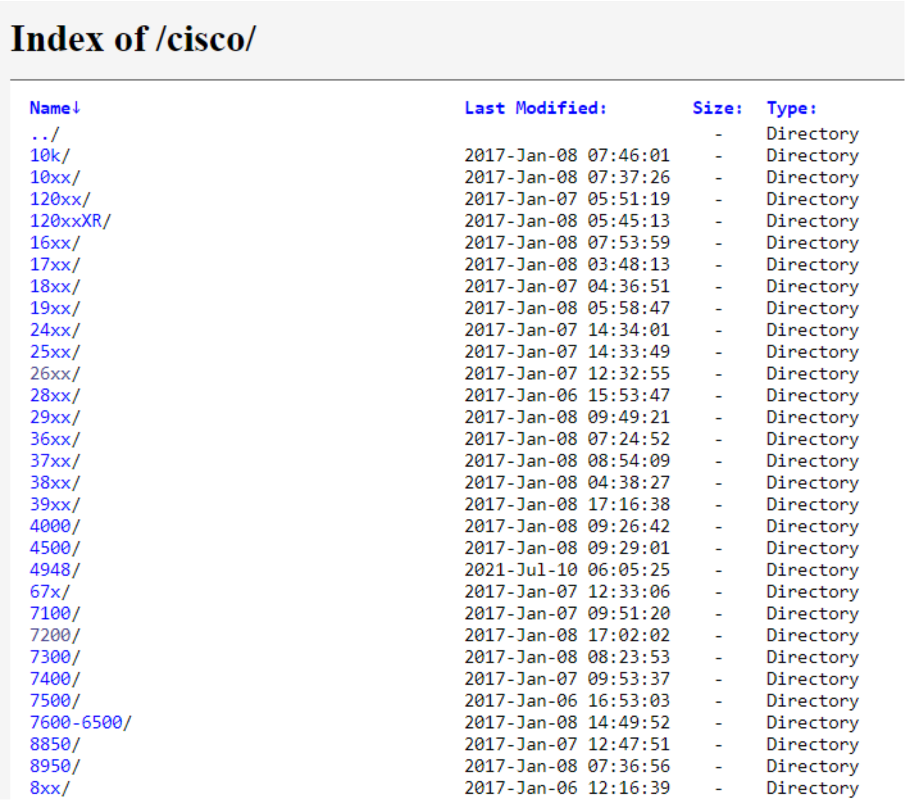
Name: 180041120_Lab1

Location: C:\Users\Farhan Ishmam\GNS3\projects\180041120_Lab1

- After creating the project, we download the required IOS files from

  http://tfr.org/cisco/

  A screenshot of the site is given below:

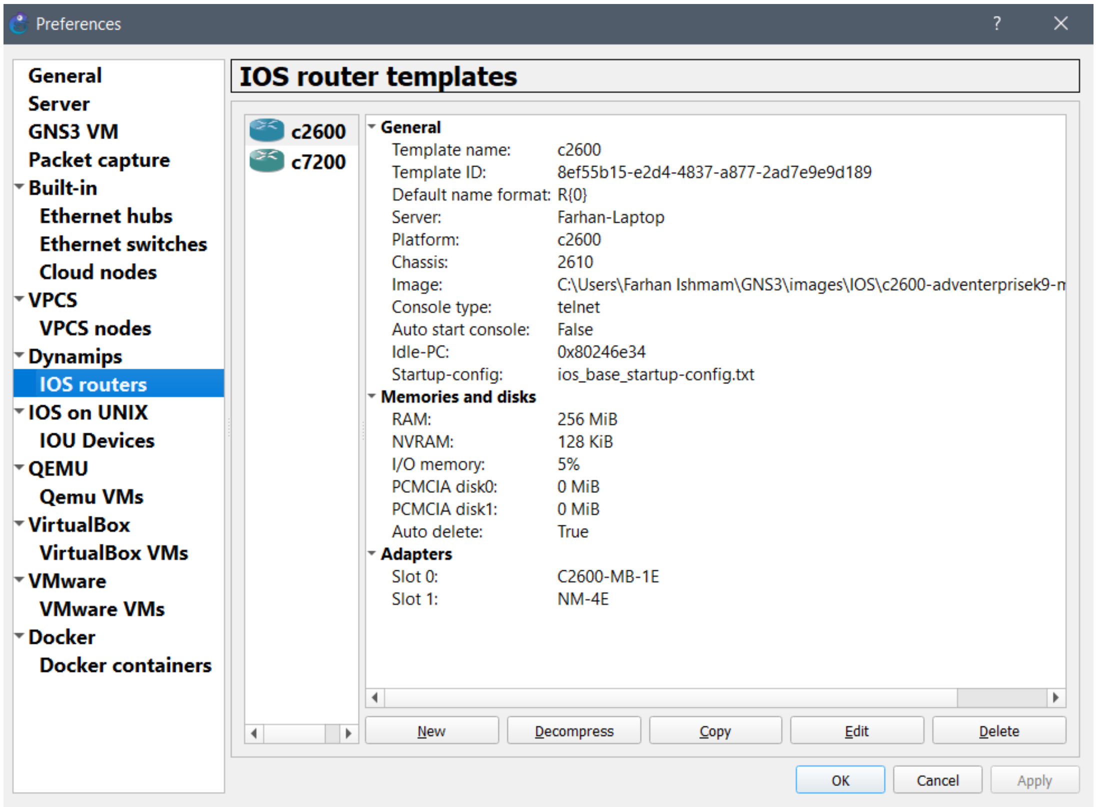


Index of /cisco/

Name↓	Last Modified:	Size:	Type:
../		-	Directory
10k/	2017-Jan-08 07:46:01	-	Directory
10xx/	2017-Jan-08 07:37:26	-	Directory
120xx/	2017-Jan-07 05:51:19	-	Directory
120xxXR/	2017-Jan-08 05:45:13	-	Directory
16xx/	2017-Jan-08 07:53:59	-	Directory
17xx/	2017-Jan-08 03:48:13	-	Directory
18xx/	2017-Jan-07 04:36:51	-	Directory
19xx/	2017-Jan-08 05:58:47	-	Directory
24xx/	2017-Jan-07 14:34:01	-	Directory
25xx/	2017-Jan-07 14:33:49	-	Directory
26xx/	2017-Jan-07 12:32:55	-	Directory
28xx/	2017-Jan-06 15:53:47	-	Directory
29xx/	2017-Jan-08 09:49:21	-	Directory
36xx/	2017-Jan-08 07:24:52	-	Directory
37xx/	2017-Jan-08 08:54:09	-	Directory
38xx/	2017-Jan-08 04:38:27	-	Directory
39xx/	2017-Jan-08 17:16:38	-	Directory
4000/	2017-Jan-08 09:26:42	-	Directory
4500/	2017-Jan-08 09:29:01	-	Directory
4948/	2021-Jul-10 06:05:25	-	Directory
67x/	2017-Jan-07 12:33:06	-	Directory
7100/	2017-Jan-07 09:51:20	-	Directory
7200/	2017-Jan-08 17:02:02	-	Directory
7300/	2017-Jan-08 08:23:53	-	Directory
7400/	2017-Jan-07 09:53:37	-	Directory
7500/	2017-Jan-06 16:53:03	-	Directory
7600-6500/	2017-Jan-08 14:49:52	-	Directory
8850/	2017-Jan-07 12:47:51	-	Directory
8950/	2017-Jan-08 07:36:56	-	Directory
8xx/	2017-Jan-06 12:16:39	-	Directory

From this site, we click on the IOS file we want to download. In this project, we are using a 7200 router, so we click on that directory and pick an IOS file.

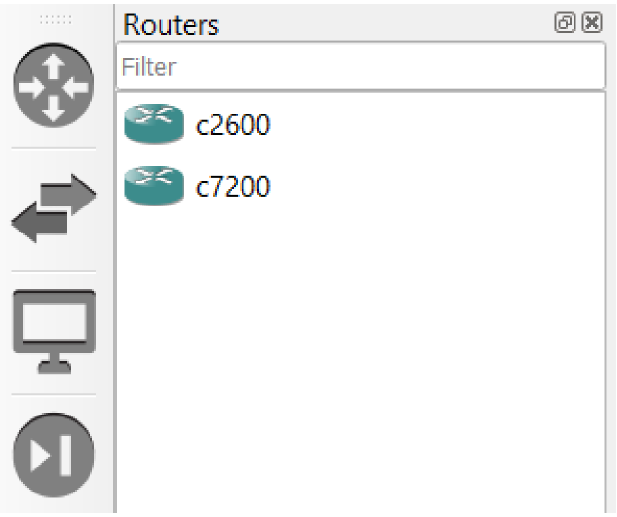
- After downloading the IOS file, we load it by going to Edit>Preferences. Then we click on IOS routers and click on New. While adding, we won't decompress the image file, and select the necessary slots. Then we allocate memory and find an idle PC for the router. A screenshot of the preferences tab after adding the router is given below:



IOS router templates

```
General
  Template name:          c2600
  Template ID:            8ef55b15-e2d4-4837-a877-2ad7e9e9d189
  Default name format:    R{0}
  Server:                 Farhan-Laptop
  Platform:               c2600
  Chassis:                2610
  Image:                  C:\Users\Farhan Ishmam\GNS3\images\IOS\c2600-adventerprisek9-m
  Console type:           telnet
  Auto start console:     False
  Idle-PC:                0x80246e34
  Startup-config:         ios_base_startup-config.txt
Memories and disks
  RAM:                    256 MiB
  NVRAM:                  128 KiB
  I/O memory:             5%
  PCMCIA disk0:           0 MiB
  PCMCIA disk1:           0 MiB
  Auto delete:            True
Adapters
  Slot 0:                 C2600-MB-1E
  Slot 1:                 NM-4E
```

- Finally, we click on Apply, and select the router from the Routers panel at the left side of the screen. A screenshot of the routers panel is given below:



Step 3: Creating the Network Topology

- We select the necessary routers from the routers panel. In our case, we are using c7200. In slot-1, we select PA-4T+ for serial communication. Two such routers are created.

- For selecting the switches, we go to the switches panel and select Ethernet Switch. Then we take two such switches.

- For the end devices, we select VPCS and take two of them.

- We connect all the devices by clicking on the connection tab. Then we select the device we want to connect along with the interface (fast ethernet0/0, serial0/0 etc.) Similarly, we select the other device of the connection with its interface and complete the connection. In this way, we connect all the devices according to the diagram.

- Finally, we add notes and start all the devices.

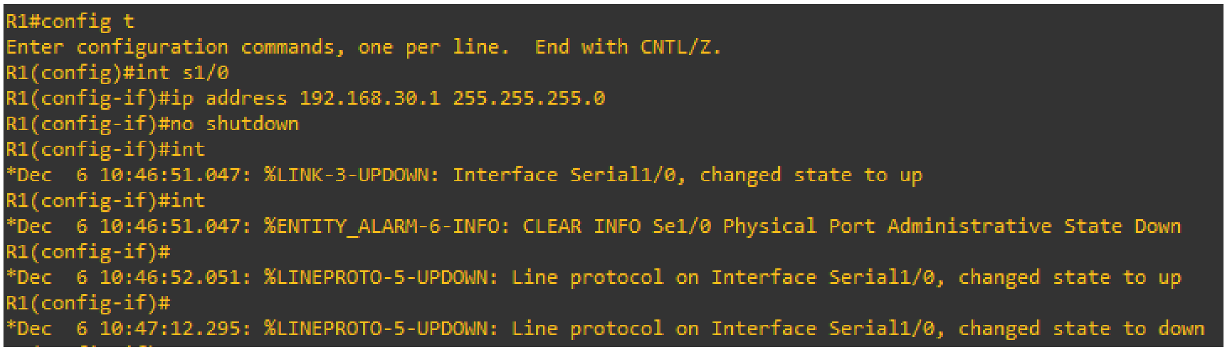
Step 4: Configuring interfaces for R1 & R2

- We configure an interface by selecting the router, and then selecting the interface. Then we assign an ip address to the interface and change the state to up.

- The following commands are used in **router R1** for **interface s1/0**:

```
R1(config)#interface s1/0
R1(config-if)#ip address 192.168.30.1 255.255.255.0
R1(config-if)#no shutdown
```

The screenshot of the command is given below:

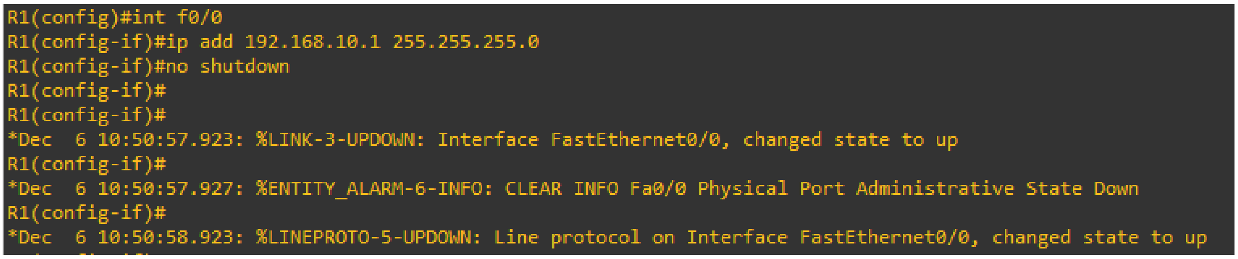


```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#int s1/0
R1(config-if)#ip address 192.168.30.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#int
*Dec  6 10:46:51.047: %LINK-3-UPDOWN: Interface Serial1/0, changed state to up
R1(config-if)#int
*Dec  6 10:46:51.047: %ENTITY_ALARM-6-INFO: CLEAR INFO Se1/0 Physical Port Administrative State Down
R1(config-if)#
*Dec  6 10:46:52.051: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed state to up
R1(config-if)#
*Dec  6 10:47:12.295: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed state to down
```

- The following commands are used in **router R1** for **interface f0/0**:

```
R1(config-if)#interface f0/0
R1(config-if)#ip address 192.168.10.1 255.255.255.0
R1(config-if)#no shutdown
```

The screenshot of the command is given below:

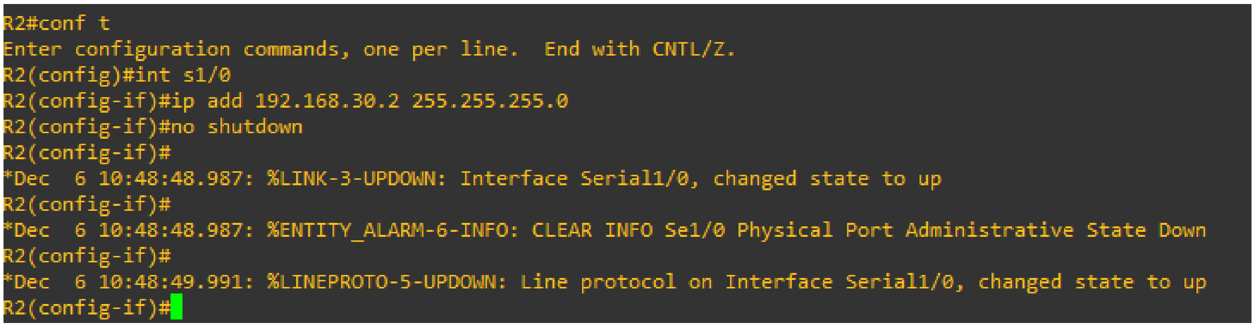


```
R1(config)#int f0/0
R1(config-if)#ip add 192.168.10.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#
R1(config-if)#
*Dec  6 10:50:57.923: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
R1(config-if)#
*Dec  6 10:50:57.927: %ENTITY_ALARM-6-INFO: CLEAR INFO Fa0/0 Physical Port Administrative State Down
R1(config-if)#
*Dec  6 10:50:58.923: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
```

- The following commands are used in **router R2** for **interface s1/0**:

```
R2(config)#interface s1/0
R2(config-if)#ip address 192.168.30.2 255.255.255.0
R2(config-if)#no shutdown
```

The screenshot of the command is given below:

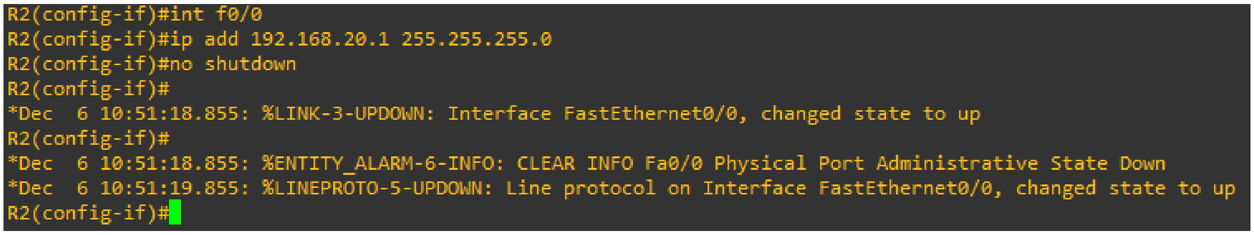


```
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#int s1/0
R2(config-if)#ip add 192.168.30.2 255.255.255.0
R2(config-if)#no shutdown
R2(config-if)#
*Dec  6 10:48:48.987: %LINK-3-UPDOWN: Interface Serial1/0, changed state to up
R2(config-if)#
*Dec  6 10:48:48.987: %ENTITY_ALARM-6-INFO: CLEAR INFO Se1/0 Physical Port Administrative State Down
R2(config-if)#
*Dec  6 10:48:49.991: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed state to up
R2(config-if)#
```

- The following commands are used in **router R2** for **interface f0/0**:

```
R2(config-if)#interface f0/0
R2(config-if)#ip address 192.168.20.1 255.255.255.0
R2(config-if)#no shutdown
```

The screenshot of the command is given below:



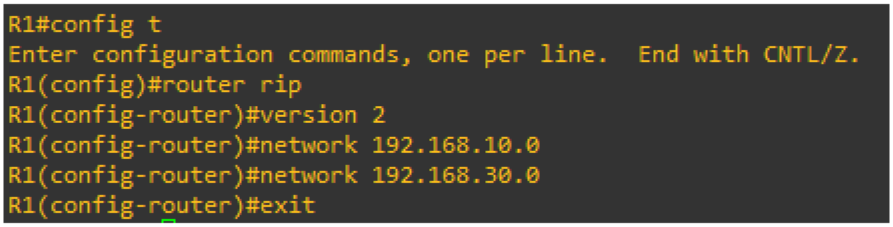
```
R2(config-if)#int f0/0
R2(config-if)#ip add 192.168.20.1 255.255.255.0
R2(config-if)#no shutdown
R2(config-if)#
*Dec  6 10:51:18.855: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
R2(config-if)#
*Dec  6 10:51:18.855: %ENTITY_ALARM-6-INFO: CLEAR INFO Fa0/0 Physical Port Administrative State Down
*Dec  6 10:51:19.855: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R2(config-if)#
```

Step 5: Enabling RIP on the routers

- In router R1, we use the following commands to enable RIP:

```
R1(config)#router rip
R1(config-router)#version 2
R1(config-router)#network 192.168.10.0
R1(config-router)#network 192.168.30.0
```

The screenshot of the commands executed is given below:

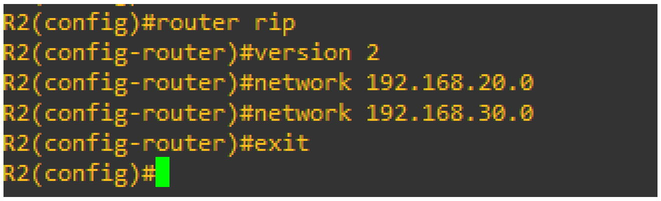


```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#router rip
R1(config-router)#version 2
R1(config-router)#network 192.168.10.0
R1(config-router)#network 192.168.30.0
R1(config-router)#exit
```

- In router R2, we use the following commands to enable RIP:

```
R2(config)#router rip
R2(config-router)#version 2
R2(config-router)#network 192.168.20.0
R2(config-router)#network 192.168.30.0
```

The screenshot of the commands executed is given below:



```
R2(config)#router rip
R2(config-router)#version 2
R2(config-router)#network 192.168.20.0
R2(config-router)#network 192.168.30.0
R2(config-router)#exit
R2(config)#
```

Step 6: Configuring IP of End Devices

- We set up the IP address of the end devices by clicking on the PC and going to the console.

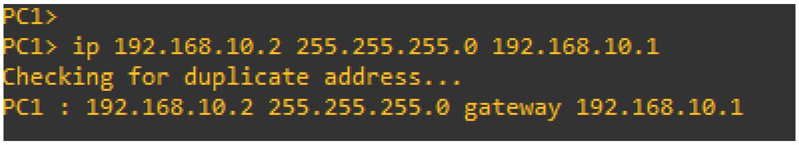
For **PC-1**, the command is:

```
ip 192.168.10.2 255.255.255.0 192.168.10.1
```

For **PC-2**, the command is:

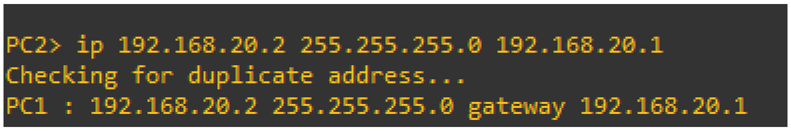
```
ip 192.168.20.2 255.255.255.0 192.168.20.1
```

Screenshot of the ip configuration in **PC-1** is given below:



```
PC1>
PC1> ip 192.168.10.2 255.255.255.0 192.168.10.1
Checking for duplicate address...
PC1 : 192.168.10.2 255.255.255.0 gateway 192.168.10.1
```

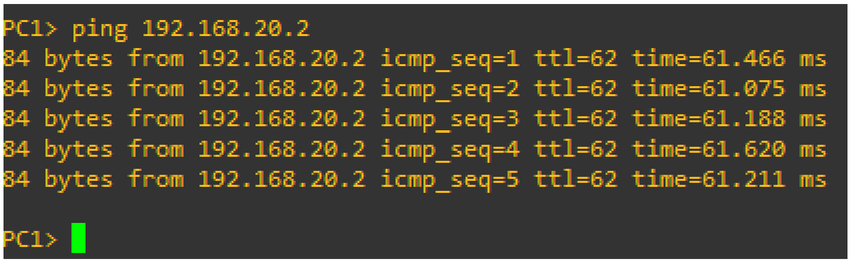
Screenshot of the ip configuration in **PC-2** is given below:



```
PC2> ip 192.168.20.2 255.255.255.0 192.168.20.1
Checking for duplicate address...
PC1 : 192.168.20.2 255.255.255.0 gateway 192.168.20.1
```

Step 7: Verifying the Configuration

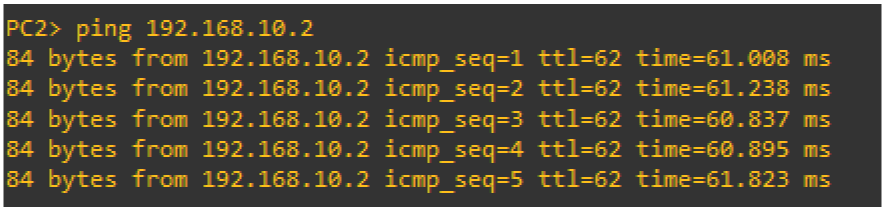
- We verify the configuration by pinging from **PC-1 to PC-2**



```
PC1> ping 192.168.20.2
84 bytes from 192.168.20.2 icmp_seq=1 ttl=62 time=61.466 ms
84 bytes from 192.168.20.2 icmp_seq=2 ttl=62 time=61.075 ms
84 bytes from 192.168.20.2 icmp_seq=3 ttl=62 time=61.188 ms
84 bytes from 192.168.20.2 icmp_seq=4 ttl=62 time=61.620 ms
84 bytes from 192.168.20.2 icmp_seq=5 ttl=62 time=61.211 ms

PC1>
```

- Then we ping from **PC-2 to PC-1**

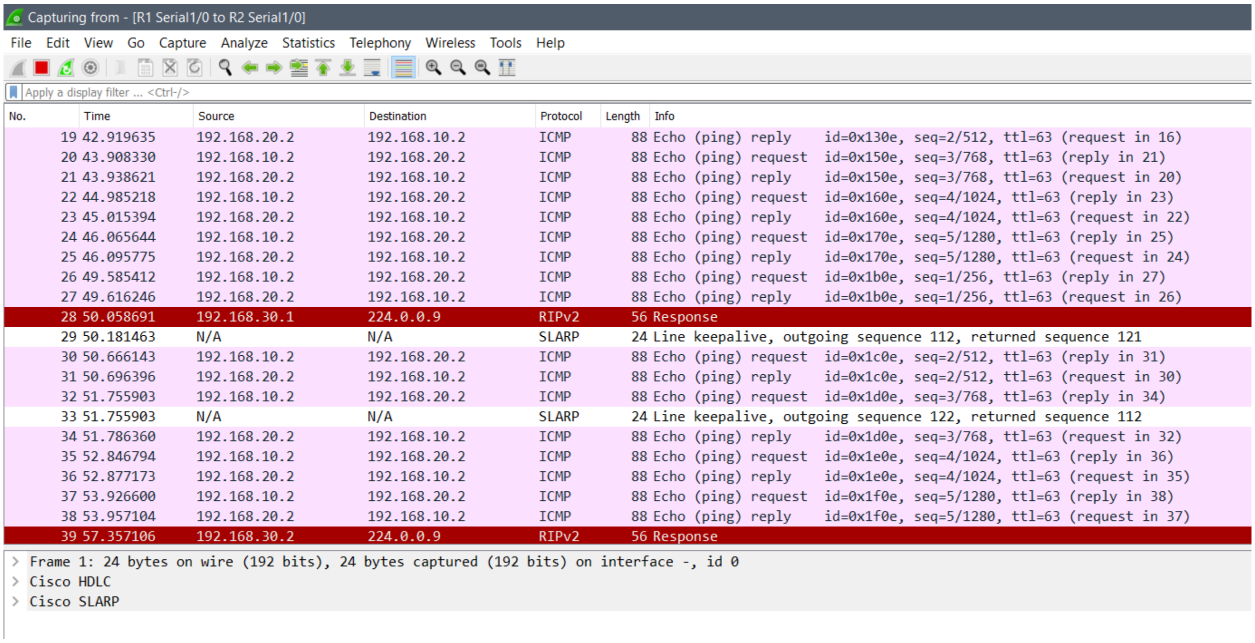


```
PC2> ping 192.168.10.2
84 bytes from 192.168.10.2 icmp_seq=1 ttl=62 time=61.008 ms
84 bytes from 192.168.10.2 icmp_seq=2 ttl=62 time=61.238 ms
84 bytes from 192.168.10.2 icmp_seq=3 ttl=62 time=60.837 ms
84 bytes from 192.168.10.2 icmp_seq=4 ttl=62 time=60.895 ms
84 bytes from 192.168.10.2 icmp_seq=5 ttl=62 time=61.823 ms
```

Step 8: Capturing the transmitted packets using Wireshark

- We right-click on a connection then click on start capture. This will open Wireshark. Then if we ping PC-1 from PC-2 the details of the sent and received packets will be shown.

  A screenshot of the captured packets in Wireshark is given below:



Capturing from - [R1 Serial1/0 to R2 Serial1/0]
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.     Time        Source          Destination     Protocol  Length Info
     19 42.919635   192.168.20.2    192.168.10.2    ICMP        88 Echo (ping) reply    id=0x130e, seq=2/512, ttl=63 (request in 16)
     20 43.908330   192.168.10.2    192.168.20.2    ICMP        88 Echo (ping) request  id=0x150e, seq=3/768, ttl=63 (reply in 21)
     21 43.938621   192.168.20.2    192.168.10.2    ICMP        88 Echo (ping) reply    id=0x150e, seq=3/768, ttl=63 (request in 20)
     22 44.985218   192.168.10.2    192.168.20.2    ICMP        88 Echo (ping) request  id=0x160e, seq=4/1024, ttl=63 (reply in 23)
     23 45.015394   192.168.20.2    192.168.10.2    ICMP        88 Echo (ping) reply    id=0x160e, seq=4/1024, ttl=63 (request in 22)
     24 46.065644   192.168.10.2    192.168.20.2    ICMP        88 Echo (ping) request  id=0x170e, seq=5/1280, ttl=63 (reply in 25)
     25 46.095775   192.168.20.2    192.168.10.2    ICMP        88 Echo (ping) reply    id=0x170e, seq=5/1280, ttl=63 (request in 24)
     26 49.585412   192.168.10.2    192.168.20.2    ICMP        88 Echo (ping) request  id=0x1b0e, seq=1/256, ttl=63 (reply in 27)
     27 49.616246   192.168.20.2    192.168.10.2    ICMP        88 Echo (ping) reply    id=0x1b0e, seq=1/256, ttl=63 (request in 26)
     28 50.058691   192.168.30.1    224.0.0.9       RIPv2       56 Response
     29 50.181463   N/A             N/A             SLARP       24 Line keepalive, outgoing sequence 112, returned sequence 121
     30 50.666143   192.168.10.2    192.168.20.2    ICMP        88 Echo (ping) request  id=0x1c0e, seq=2/512, ttl=63 (reply in 31)
     31 50.696396   192.168.20.2    192.168.10.2    ICMP        88 Echo (ping) reply    id=0x1c0e, seq=2/512, ttl=63 (request in 30)
     32 51.755903   192.168.10.2    192.168.20.2    ICMP        88 Echo (ping) request  id=0x1d0e, seq=3/768, ttl=63 (reply in 34)
     33 51.755903   N/A             N/A             SLARP       24 Line keepalive, outgoing sequence 122, returned sequence 112
     34 51.786360   192.168.20.2    192.168.10.2    ICMP        88 Echo (ping) reply    id=0x1d0e, seq=3/768, ttl=63 (request in 32)
     35 52.846794   192.168.10.2    192.168.20.2    ICMP        88 Echo (ping) request  id=0x1e0e, seq=4/1024, ttl=63 (reply in 36)
     36 52.877173   192.168.20.2    192.168.10.2    ICMP        88 Echo (ping) reply    id=0x1e0e, seq=4/1024, ttl=63 (request in 35)
     37 53.926600   192.168.10.2    192.168.20.2    ICMP        88 Echo (ping) request  id=0x1f0e, seq=5/1280, ttl=63 (reply in 38)
     38 53.957104   192.168.20.2    192.168.10.2    ICMP        88 Echo (ping) reply    id=0x1f0e, seq=5/1280, ttl=63 (request in 37)
     39 57.357106   192.168.30.2    224.0.0.9       RIPv2       56 Response

> Frame 1: 24 bytes on wire (192 bits), 24 bytes captured (192 bits) on interface -, id 0
> Cisco HDLC
> Cisco SLARP
```

Challenges (if any):

- I faced no major challenges