Lab 6

Computer Networks Laboratory

Static Routing: Single Segment & Multiple Segments

The lab's schedule

Published date:	03/05/22
Quiz date:	10,12/05/22
Deadline for the final report:	26/05/22

BEN-GURION UNIVERSITY OF THE NEGEV - COMMUNICATION SYSTEMS ENGINEERING

General instructions

- **The final report** will be based on your answers to the "*Attach to your report*" sections at this document.

- **Remember** to pay attention to the "Final report submission" in the Introduction Lab.

- Most of the laboratory experiments based on the book: "*Mastering Networks: An Internet Lab Manual*"

- It is recommended to **read the all exercise before you do it, to understand the main idea**.

- IP address is composed of four octets ("octet1.octet2.octet3.octet4").
  In our Labs all IP addresses will be according to the Network Figure, except *octet2* in network addresses between routers.
  ***Octet2* will be according to the pair's number ("10.*X*.0.1", *X = pair's number*).**

- **Write the number *X*** clearly **on the title page** of your final report.

Reading material

- Be familiar with these Linux commands by reading the manual:

rmdir
chmod
kill
ping
nano
ftp
arp
mv
cp
rm
mkdir
ifconfig
ssh
route
man
pwd
ls
more
tcpdump
scp

(For the lab entry quiz only study the highlighted commands)

- You can read the "*man pages*" by execute the command:

```
man "Name_of_instruction"
```

  on a Linux system or at the website: http://linux.die.net/man

- Read about filtering packets in Wireshark (both links):
  https://www.Wireshark .org/docs/wsug_html_chunked/ChWorkDisplayFilterSection.html
  https://www.Wireshark .org/docs/wsug_html_chunked/ChWorkBuildDisplayFilterSection.html

- Be familiar with the term of "*Static Routing*".
  http://en.wikipedia.org/wiki/Static_routing

- Be familiar with the routing table structure and its fields:
  http://en.wikipedia.org/wiki/Routing_table#Contents_of_routing_tables

- The route decision on a Router, based on the routing table and the principle of "*Longest prefix match*". Read about this topic:
  http://en.wikipedia.org/wiki/Longest_prefix_match

- Be familiar with the term of "*default gateway*".
  http://en.wikipedia.org/wiki/Default_gateway

- Be familiar with the Internet Control Message Protocol (*ICMP*)
  https://supportforums.cisco.com/document/7416/icmp-internet-control-message-protocol

- Be familiar with the working principles of "*traceroute*" and its uses:
  http://en.wikipedia.org/wiki/Traceroute#Implementation

Preliminary questions

- Write the syntax for a *Wireshark display filter* that shows IP datagrams with a destination IP address equal to 10.0.1.50 and Ethernet frame sizes greater than 400 bytes.

- Write the syntax for a *Wireshark display filter* that shows packets containing ICMP messages with a source or destination IP address equal to 10.0.1.12 and frame numbers between 15 and 30.

- In computer networks which is based on *static routing*, how a new entry is inserted?

- (In the routing table)What is the "*Gateway*" field and what its purpose?

- Given the following Routing table, through which gateway packets with source address 10.0.155.136, would be routed?

Network Destination	Gateway
0.0.0.0/0	10.0.2.1
10.0.0.0/16	10.0.2.2
10.0.144.0/20	10.0.2.3

10.0.155.0/25	10.0.2.4
10.0.155.128/32	10.0.2.5

- Given the following *routing table* at PC1, assuming its *default gateway* is 10.0.2.4, what is the range of IP addresses which will be routed to its *default gateway*?

Network Destination	**Gateway**
10.1.1.0/24	10.0.2.1
10.1.2.0/24	10.0.2.2
10.1.3.0/24	10.0.2.3

- What is the command to set a *static route* on a Linux PC to network 10.0.12.0/12 through gateway 10.1.35.4?

- What is the command to delete the entry from question 5?

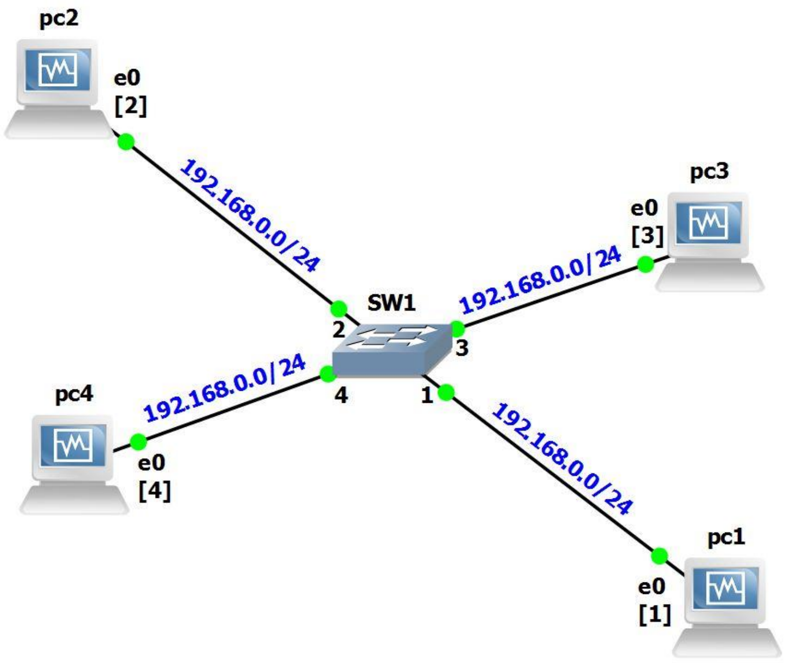
- Explain which packets are sent when issuing the command *traceroute* and what is the different between them?

- How could we use *tracroute* for discover a network's topology?

The practical section

1.Topology 1 Configuration

Topology 1



Do

1.1. Open GNS3.

1.2. Start a new project and save as "*Topology 1*".

1.3. Make sure that all four computers are imported.

1.4. Build this topology.

1.5. Configure all IP addresses of the network interfaces according to the figure.
For example:

```
pc1% ifconfig eth0 192.168.0.1 netmask 255.255.255.0
```

1.6. Check your network by performing *ping* between each pair of computers.

```
pc1% ping -c 5 192.168.0.2
```

Note to do the following two parts after you finish with experiments: "*Working with display filter in Wireshark*" and "*More complex display filters in Wireshark* ".

1.7. Save the project and stop all components.

1.8. Find the projects folder and compressed this project using ZIP. This will save the topology, but not the definitions that were set using the temporary command "ifconfig".

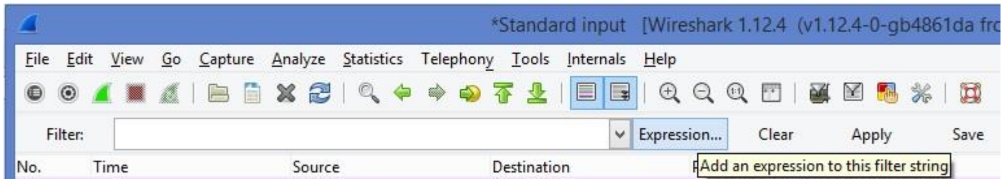
2. Working with display filter in Wireshark

You can set display filters, which allow you to select a subset of the captured data for display in the main window of Wireshark .

Pre

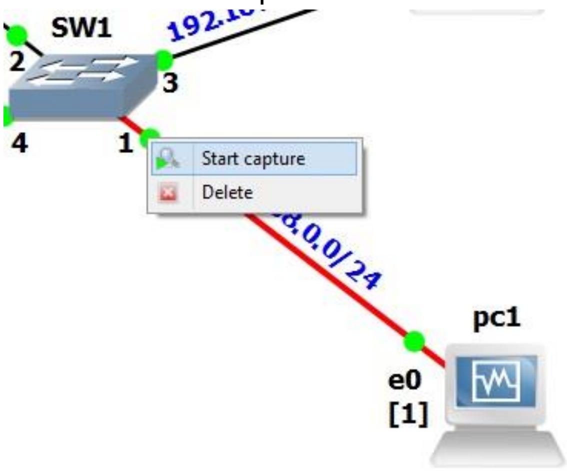
2.1. The exercise is based on *Topology 1*.

2.2. The “Filter Expression” dialog box is an excellent way to learn how to write Wireshark display filter strings.



Do

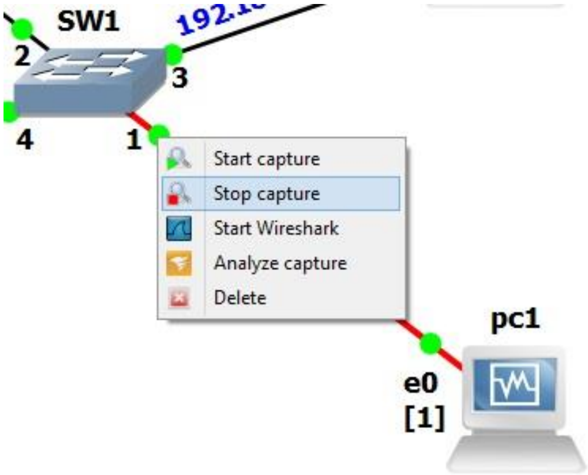
2.3. Start *Wireshark* on port 1 of the switch.



2.4. On PC2, issue a *ping* command to PC1:

```
pc2% ping -c 5 192.168.0.1
```

2.5. Stop capture traffic.



2.6. Save the *Wireshark pcap* to file.

2.7. Set a display filter so that only IP datagrams with destination IP address 192.168.0.2 are shown.

2.8. Set a display filter so that only IP datagrams with 192.168.0.2 as source or destination are shown.

Attach to your report

2.9. Attach the first filter you've used.

2.10. What are the packet types captured with the first filter?

2.11. Attach the second filter you've used.

2.12. How many packets were captured with the second filter?

3. More complex display filters in Wireshark

In this exercise, you learn how to use more sophisticated filters, in order to display specific packets.

Pre

3.1. The exercise is based on *Topology 1*.

Do

3.2. Start *Wireshark* on port 1 of the switch.

3.3. On PC2, issue a *ping* command to PC1:

```
pc2% ping -c 20 192.168.0.1
```

3.4. At the same time, start a ssh session from PC3 to PC1 as “ubu” user by typing:

```
pc3% ssh ubu@192.168.0.1
```

You're asked to connect as "ubu". Enter the password: `ubu`.
After you log in successfully to PC1, log out with the command `exit`.

3.5. Stop the capture of Wireshark .

3.6. Save the *Wireshark pcap* to file.

3.7. Apply a set of display filters to the captured traffic:

3.7.1. Display packets that contain ICMP messages with the IP address of PC2 as the destination IP address.

3.7.2. Display packets that contain TCP traffic with frame length that less than 100.

3.7.3. Display ICMP packets with type equal to 0 or TCP packets with a destination port equal to 22.

Attach to your report

3.8. Attach the first filter you've used.

3.9. Attach the second filter you've used.

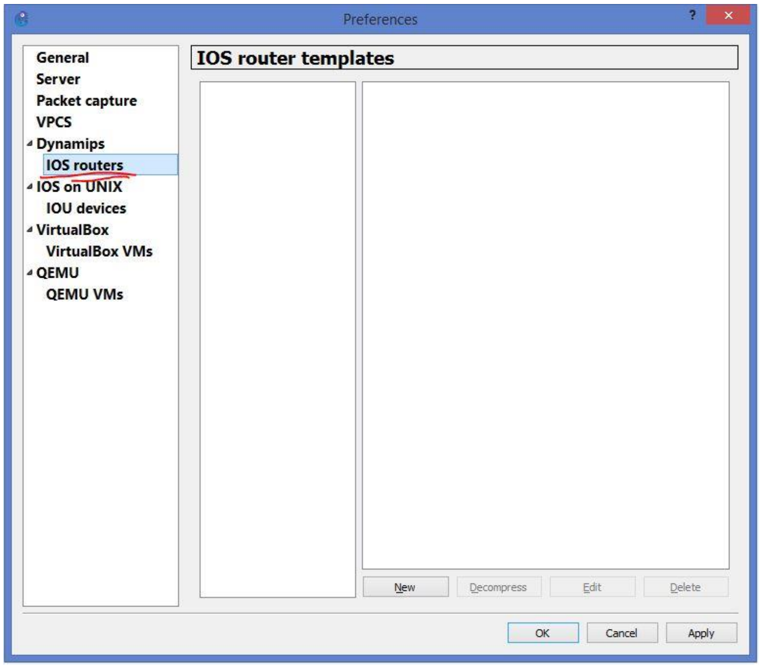
3.10. Attach the third filter you've used.

4. Adding a Router to GNS3

Do

4.1. At GNS3, go to: "`Edit->Preferences…`"

4.2. Choose IOS routers.

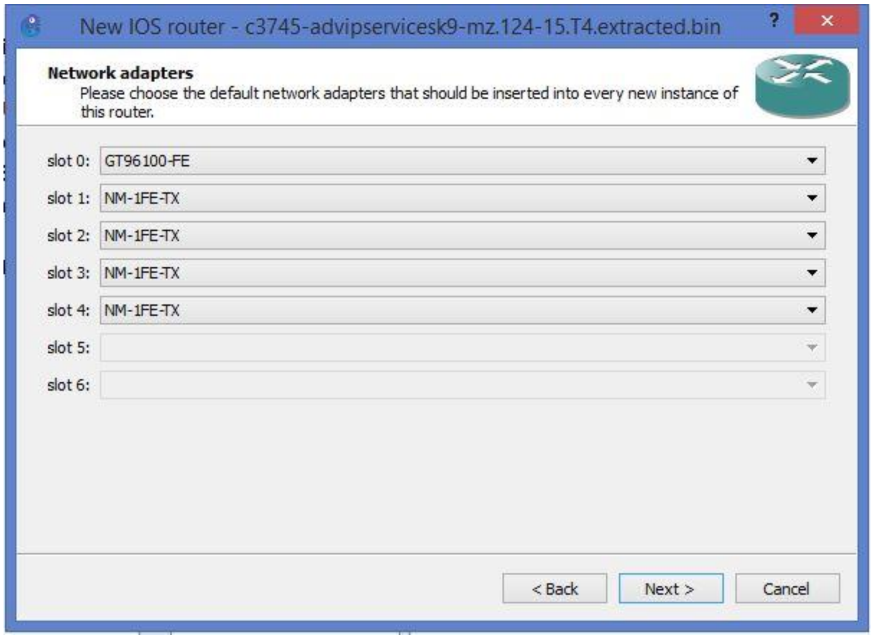


4.3. Click "*New*".

4.4. In the opened window, click "*Browse…*", and select the file: "*NC_2_Lab\GNS3\images\IOS\c3745-advipservicesk9-mz.124-15.T4.extracted.bin*".
And Click "*Next >*".

4.5. In the following windows, click: "*Next >*", "*Next >*".

4.6. In the "Network adapters" window, choose slots 1-4 to be "*NM-1FE-TX*" (slot 0 stays the same).



New IOS router - c3745-advipservicesk9-mz.124-15.T4.extracted.bin

Network adapters

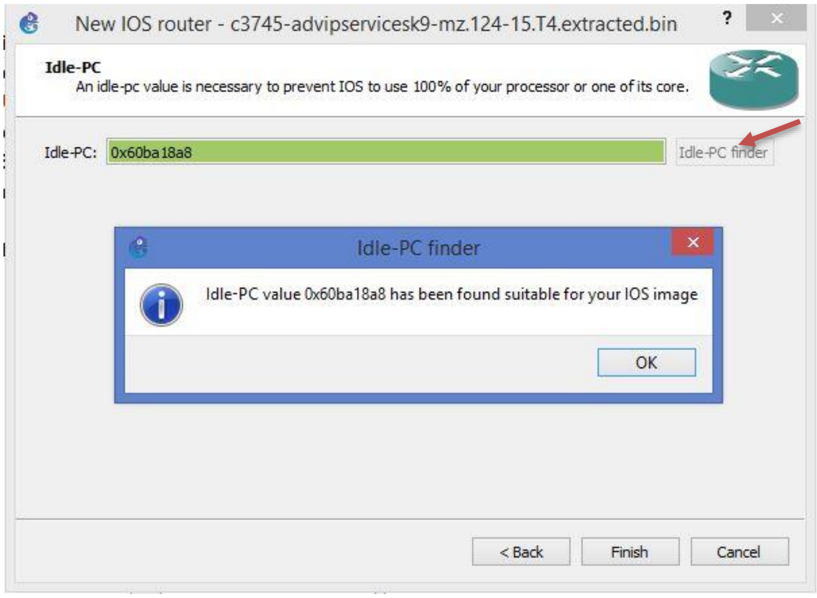
Please choose the default network adapters that should be inserted into every new instance of this router.

slot 0: GT96100-FE
slot 1: NM-1FE-TX
slot 2: NM-1FE-TX
slot 3: NM-1FE-TX
slot 4: NM-1FE-TX
slot 5:
slot 6:

And Click "*Next >*".

4.7. In the following window, click "*Next >*".

4.8. In the “Idle-PC” window, click “*Idle-PC finder*”, and wait for “*OK*”.



New IOS router - c3745-advipservicesk9-mz.124-15.T4.extracted.bin

Idle-PC

An idle-pc value is necessary to prevent IOS to use 100% of your processor or one of its core.

Idle-PC: 0x60ba18a8

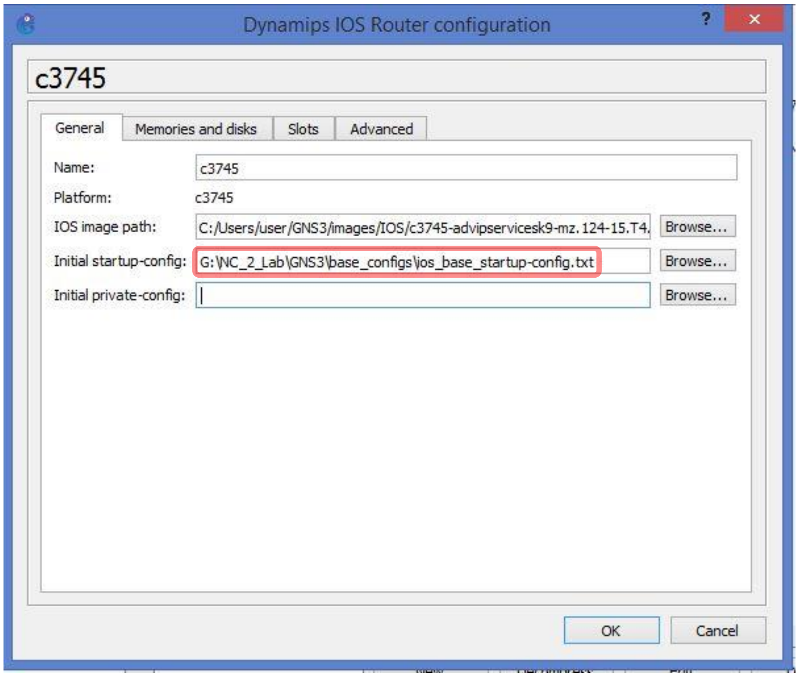
Idle-PC finder

Idle-PC value 0x60ba18a8 has been found suitable for your IOS image

4.9. Click “*Finish*”.

4.10. Choose the new Router and click “*edit*”.

4.11. In the tab "General", *browse…* for “*Initial startup-config*” file. Select the file: “*NC_2_Lab\GNS3\base_configs\ios_base_startup-config.txt*”.



Dynamips IOS Router configuration

c3745

General | Memories and disks | Slots | Advanced

Name: c3745

Platform: c3745

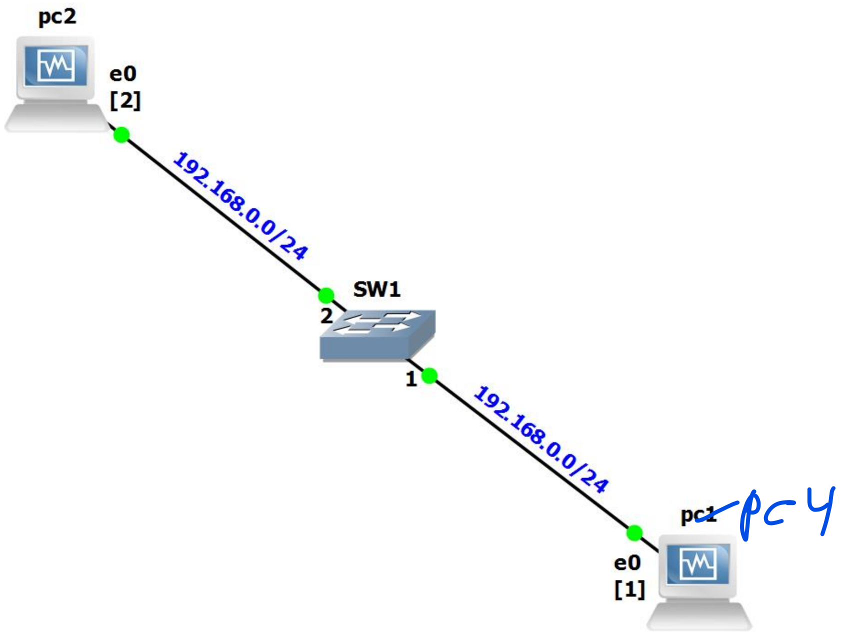
IOS image path: C:/Users/user/GNS3/images/IOS/c3745-advipservicesk9-mz.124-15.T4.

Initial startup-config: G:\NC_2_Lab\GNS3\base_configs\ios_base_startup-config.txt

Initial private-config:

And Click: “*OK*”, “*OK*”.

5. Moving from Switching to Routing



Pre

5.1. As a starting point, you can use *Topology 1*.

Do

5.2. Build the topology according to the figure.

5.3. Configure the network interfaces of the two PCs according to the figure.

5.4. Test your network by *ping* from PC1 to PC2.

5.5. Start Wireshark on switch port #1 and switch port #2.

5.6. *Ping* from PC1 to PC2 by:

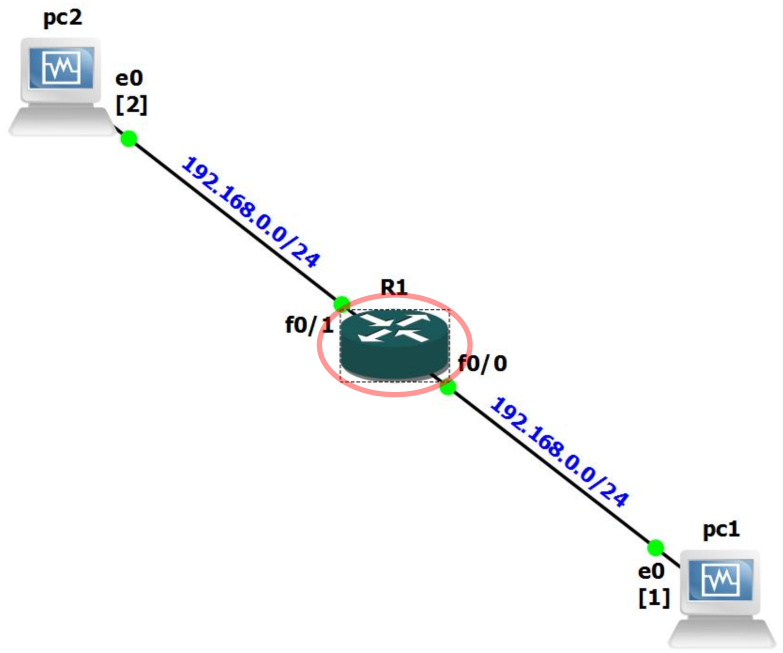
```
pc1% ping -c 2 192.168.0.2
```

5.7. Notice what are the types of the packets, and what the source and destination addresses (for IP and MAC).

5.8. Take a screenshot of the *ping* result.

5.9. Stop capture and save the Wireshark pcap to file.

5.10. Delete the switch and connect a router in place.



5.11. Start the router and repeat steps 5-9 (but start Wireshark on the router's ports instead).

5.12. Did the ping work?

Attach to your report

5.13. For the first scenario (Switch):

5.13.1. Describe the packets flow in the network (In your answer, referred to the packet's type, and which PC the IP and MAC address belongs to).

5.13.2. What is the result of the ping command?

5.14. For the second scenario (Router):

5.14.1. Describe the packets flow in the network (In your answer, referred to the packet's type and its source/destination).

5.14.2. What is the result of the ping command?

5.14.3. The *ping* fails, but explain how the output of the *ping* command related to packets flow on the links and what is the reason for the failure.

6. Configuring Cisco Router Interfaces

Pre

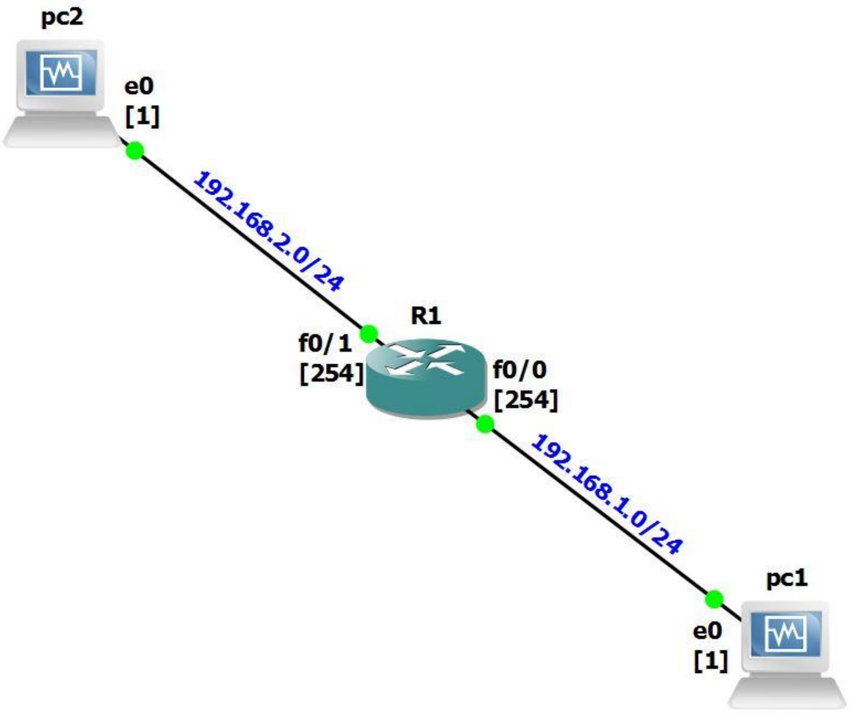
6.1. The exercise is based on the previous one.

Do

6.2. First, save as a new project as *Topology 2*.

6.3. Now, configure the network as two subnets, according to the figure below:

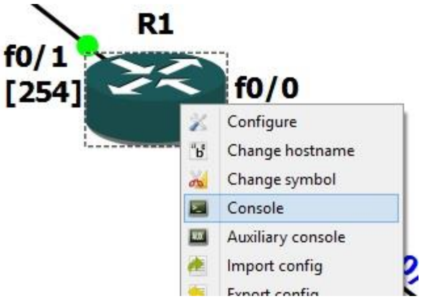
Topology 2



Notice the difference between the network addresses.

6.4. Configure the network interfaces of the two PCs.
It is recommended, to search the "*Useful_Commands.pdf*" for how to configure computer's network settings as permanent.

6.5. Open the router terminal and configure its two network interfaces.



Here is an example of how to configure one the network interfaces, "f0/0" of R1:

```
R1#configure terminal
R1(config)#interface FastEthernet0/0
R1(config-if)#ip address 192.168.1.254 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#end
```

6.6. Verify the network configuration by typing:

```
R1#show running-config
```

(Use the space bar to scroll down).

6.7. Also, try to *ping* from the Router to each PC, to verify the links.
For example:

```
R1#ping 192.168.1.1
```

6.8. Type `write` and press Enter to save the running configuration:

```
R1#write
Building configuration...
[OK]
```

This action ensures that the next time you start the Router, the current network configuration will be applied.

6.9. Save the project and stop all network components.

6.10. Close GNS3, find the topology folder and ZIP it.

6.11. Start GNS3 and open "*Topology 2*" again.

6.12. Start all network components and verify (using ping) that the network is still working properly.

6.13. **The correct working order outlines:**
- Open/New Project
- Save project as..
- Start
- Configure the network
- Write on all routers
- Save project
- Stop
- ZIP the project

6.14. Note that router command may be copy-pasted into the router terminal. If you wish, ask a lab instructor how to do it, or use your favorite Internet search engine.

7.Configuring Static Routing Table on a Linux PC

Configuring static routes in Linux is done with the command “route”.

```
route add -net “netaddress” netmask “mask” gw “gw_address”
route add -net “netaddress” netmask “mask” dev “iface”
```

Adds a routing table entry for the network prefix identified by IP address “*netaddress*” and netmask “*mask*”. The next-hop is identified by IP address “*gw_address*” or by interface “*iface*”.

```
route add -host “hostaddress” qw “gw_address”
route add -host “hostaddress” dev “iface”
```

Adds a host route entry for IP address “*hostaddres*” with the next-hop identified by IP address “*gw_address*” or by interface “*iface*”.

```
route add default gw “gw_address”
```

Sets the default route to IP address “*gw_address*”.

```
route del -net “netaddress” netmask “mask” gw “gw_address”
route del -host “hostaddress” gw “gw_address”
route del default gw “gw_address”
```

Deletes an existing route from the routing table. It is not necessary to type all arguments. If enough arguments are provided that it can be matched with an existing routing entry, the first entry that matches the given arguments is deleted.

```
route -e
```

Displays the current routing table with extended fields. The command is identical to the “`netstat -r`” command.

```
route -c
```

Displays the routing table cache.

Pre

7.1. The exercise is based on previous exercise "Moving from switching to routing" and on “*Topology 2*” (from “Configuring Cisco Router Interfaces”).

Do

7.2. In order to clear all the ARP tables of the PCs and the router, reboot all network components.

7.3. Make sure that the ARP tables are empty.
Use the following command to verify on the router:

```
R1#sh ip arp
```

(Note that the router interfaces entries always remain in the table)
Use the following command to verify on the PCs:

```
pc1% arp
```

7.4. Add static routes to the PCs.
Each PC must have a separate routing entry for each remote network.
PC1 needs to know how to reach subnet 192.168.2.0/24.
PC2 needs to know how to reach subnet 192.168.1.0/24.

7.5. Here is an example how to configure a static route entry on a computer:

```
pc1% route add -net 192.168.2.0 netmask 255.255.255.0 gw
192.168.1.254
```

7.6. Verify the configuration using the command:

```
pc1% route
```

7.7. Take a screenshot of the *route* result for the two PCs.

7.8. Now, try again steps 5-9 from the previous exercise "*Moving from switching to routing*" (but start Wireshark on the router's ports instead, and *Ping* to the new PC2 IP address). Pay attention also to the TTL field.

Another way to configure PC how to reach remote networks is by configuring a default gateway.

7.9. Delete the static routes you have set before and configure a default gateway for each **computer**.

7.10. For example:

```
pc1% route add default gw 192.168.1.254
```

7.11. Check the connection between PC1 to PC2 using ping.

Attach to your report

7.12. Explain the command in step 5. What does it do? What is its meaning?

7.13. Explain each field in the PC routing table, what does each field signify?

7.14. Explain the rule of each *entry* (row) in PC1 routing table (from step 7), that is, what is each row used for?

7.15. What is the result of the ping command? Did it work? Why?

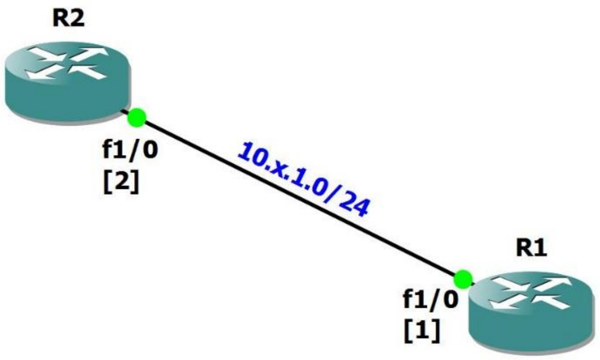
Notice, only 1 out of the 2 ICMP packets managed to get through the router and arrived to PC2. It seems that whenever this Cisco router sees an ICMP packet, it will check to see if the destination MAC address is present in the router's ARP cache. If the MAC address is not present, the router will discard this packet, but it will also perform an ARP resolution for the destination IP address. Then when the second ICMP echo request packet arrives it will be able to pull out the MAC from the cache and the second ICMP packet will be forwarded to that address. This explained why we observe only the second ICMP echo request being forwarded to the corresponding destination.

7.16. Explain how you can see the above-mentioned phenomena in the Wireshark pcap you captured. Add a screenshot if needed.

7.17. Locate one IP packet that goes through the router and describe what fields change and what fields remain unchanged in this packet, (pay attention to the identification field of the packet).

8. Topology 3 Configuration

Topology 3



Do

✓ 8.1. Start a new project and save as "*Topology 3*".

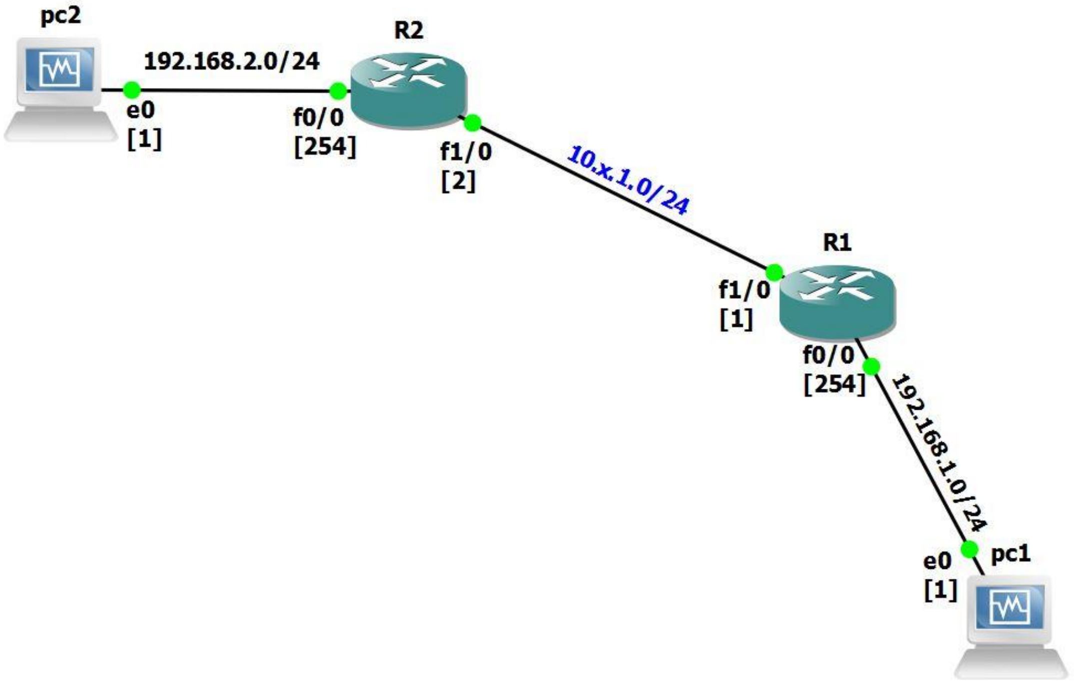
✓ 8.2. Build the topology according to the figure.

✓ 8.3. Configure the network interface on each router according to the figure.

✓ 8.4. Verify the connection between each pair of neighboring routers using ping.

✓ 8.5. Perform "`write`" on each router, save as the topology and ZIP it.

9. Configuring Static Routing Table on a Cisco Router



Pre

✓ 9.1. The exercise is based on *Topology 3*.
But first, save as a new project, before making any changes.

9.2. Search the "*Useful_Commands.pdf*", find how to configure permanent network settings for the PCs. Once this is done you will be able to reboot the system without the need to re-configure all PCs.

9.3. You may use the Linux command "*reboot*" to easily restart the virtual PC's. This will also save the Linux command line history.

Do

✓ 9.4. Add and configure the PCs according to the figure.

✓ 9.5. On the routers, configure the network interfaces, which connected to the PCs.

9.6. Set appropriate default gateway for each PC, according to the figure.

9.7. Display the content of the router's *routing table*:

```
R1#show ip route
```

Take a screenshot of the command output.

9.8. Start Wireshark on *R1* interface *f0/0* and on *R2* interface *f1/0*.

9.9. Issue ping from PC1 to PC2.

```
pc1% ping -c 3 192.168.2.1
```

9.10. Stop capture and save the Wireshark pcaps to files.

9.11. Take a screenshot of the *ping* result.

9.12. Now, configure on **each** router a static routing entry for the remote network.

9.13. For example:

```
R1#configure terminal
R1(config)#ip route 192.168.2.0 255.255.255.0 10.x.1.2
R1(config)#end
```

9.14. Verify the configurations by:

```
R1#show ip route
```

Take a screenshot of the command output.

9.15. Repeat steps 6-9 again.
Now, the ping should work properly.

Attach to your report

9.16. Before the additional configurations, the PC sent the packet to the network (to the default gateway). But still, the packet never reached its destination.
Using the Wireshark pcaps, explain how "Ping" knows to display the cause behind the packet loss in the network. Add a screenshot of the ping result to your report.

9.17. Compare R1's routing table before and after the additional configurations. What changes occurred?

9.18. Look at the routing table of R1. For each routing entry, explain what is its purpose, where does it route to, is there any other information the rother can gather from it? (in your answer refer to all the entry fields).

9.19. How many ICMP packets do not reach their destination? What is the reason?

10. Observing Traceroute

Pre

10.1. The exercise is based on the previous one.

Do

10.2. Start Wireshark on *R1* interface *f0/0*, on *R2* interface *f1/0* and on *R2* interface *f0/0*.
Apply a filter of "`udp or icmp`".

10.3. Execute a *traceroute* command from PC1 to PC2:

```
pc1% traceroute 192.168.2.1
```

10.4. Observe how *traceroute* gathers information on the route.
Pay attention to the TTL field and to the destination port of the UDP packets.

10.5. Stop capture and save the Wireshark pcaps to files.

Extra note: If you wish, you may also try to use the windows traceroute version "tracert" to find your path to a Google server on 8.8.8.8. It might get stuck en-route, can you guess why?

Attach to your report

10.6. What information gives us the *traceroute* command output?

10.7. As you can see, some of the first UDP packets did not reach the destination.
Can you find some principle or rule for way the packet are lost in the network? Try and locate the field in the IP header, which is the reason, and explain why the packets dropped.

10.8. Use the Wireshark output to explain the operation of *traceroute* and how it uses the ICMP indication packets that are sent back in the network (pay attention to the information that arrives in the ICMP data).

10.9. Explain how you could use *traceroute* to find connectivity problems in the network.
Think what happens when you have a long path and the trace fails at middle.

11. Multiple Matches in the Routing Table

A router or host uses a routing table to determine the next hop of the path of an IP datagram. In this exercise, you determine how an IP router resolves multiple matching entries in a routing table.

Pre

11.1. The exercise is based on previous exercise "Configuring static routing table on a Cisco router".
But first, save as a new project, before making any changes.

Do

11.2. Add the following routes to the routing table of *R1*:

```
R1(config)#ip route 10.x.0.0 255.255.0.0 10.x.1.61
R1(config)#ip route 10.x.3.0 255.255.255.0 10.x.1.71
R1(config)#ip route 10.x.3.9 255.255.255.255 10.x.1.81
```

11.3. Verify that the routing table contains the new routes:

```
R1#show ip route
```

11.4. Start Wireshark on *R1* interface *f1/0*.

11.5. Issue the following ping commands from PC1:

```
pc1% ping -c 1 10.x.3.9
pc1% ping -c 1 10.x.3.14
pc1% ping -c 1 10.x.4.1
```

Note that gateways with IP addresses 10.X.1.61, 10.X.1.71, and 10.X.1.81 do not exist. However, *R1* still sends ARP Request packets for these IP addresses.

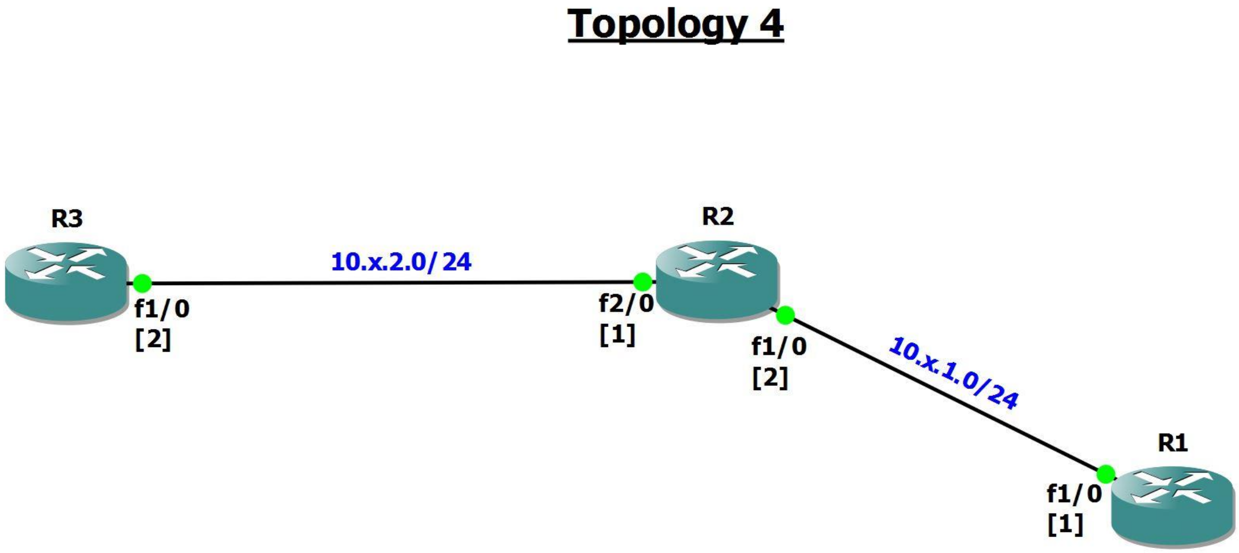
11.6. Stop capture and save the Wireshark pcap to file.

Attach to your report

11.7. Use the above outputs to indicate the number of matches for each of the preceding IP addresses.

11.8. Explain how R1 resolve multiple matches in the routing table, How is this mechanism known?

12. Topology 4 Configuration



Pre

12.1. The topology is based on *Topology 3*.
But first, before making any changes, save as a new project as *Topology 4*.

Do

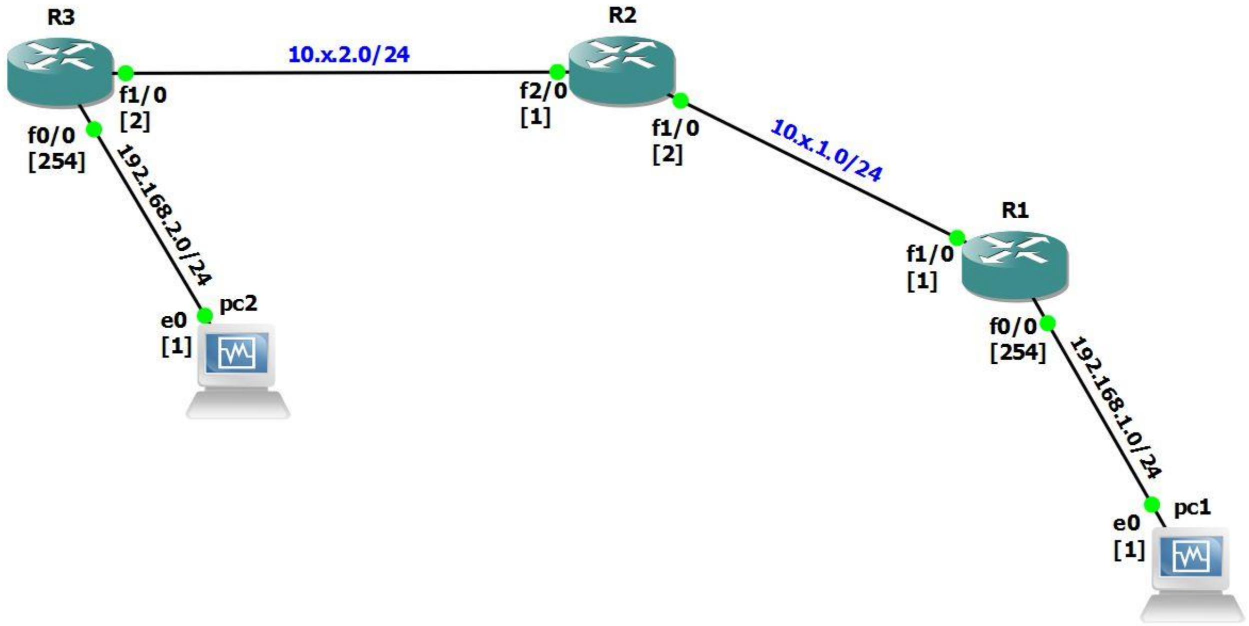
12.2. Build the topology according to the figure.

12.3. Configure the network interfaces on each router according to the figure.

12.4. Verify the connection between each pair of neighboring routers using ping.

12.5. Perform "`write`" on each router, save the topology and ZIP it.

13. Meaning of Default Routes



If a route matches, the packet is forwarded accordingly, otherwise the packet is forwarded to the default route of that router. In a default route, either the next-hop IP address or exit interface can be specified.
In this experiment, we will observe how a packet sent to an unknown network is handled.

Pre

13.1. The exercise is based on *Topology 4*.
But first, save as a new project, before making any changes.

Do

13.2. Add and configure the PCs according to the figure.

13.3. On the routers, configure the network interfaces, which are connected to the PCs.

13.4. Set appropriate default gateway for each PC, according to the figure.

13.5. Configure on *R1* a default routing entry to a next-hop, as follows:
On *R1*, default routing entry to *R2* interface *f1/0*.

```
R1#configure terminal
R1(config)#ip route 0.0.0.0 0.0.0.0 10.x.1.2
R1(config)#end
```

13.6. Configure on *R2* and *R3* default routing entries to an interfaces, as follows:
On *R2*, default routing entry through the interface *f2/0*.
On *R3*, default routing entry through the interface *f0/0*.

13.7. For example:

```
R2#configure terminal
R2(config)#ip route 0.0.0.0 0.0.0.0 FastEthernet2/0
R2(config)#end
```

13.8. Verify that the Routers' routing tables contains the new routes:

```
R1#show ip route
```

13.9. Start Wireshark on *R1* interface *f1/0*, on *R2* interface *f2/0* and on *R3* interface *f0/0*.

13.10. Issue a ping command from PC1 to a host on a network that does not exist.

```
pc1% ping –c 5 192.168.10.1
```

13.11. Observe Wireshark to see how the packets spread through the network (due to the configuration of default routes) although the destination network address does not exist.

13.12. Stop capture and save the Wireshark pcaps to files.

Note* in this experiment you will see that some of the routers replay to ARP requests not address to them. This mechanism is called "proxy ARP" and you are invited to ask the lab guides about this mechanism (or learn about it yourselves).

Attach to your report

13.13. What is the output of the ping command and why?

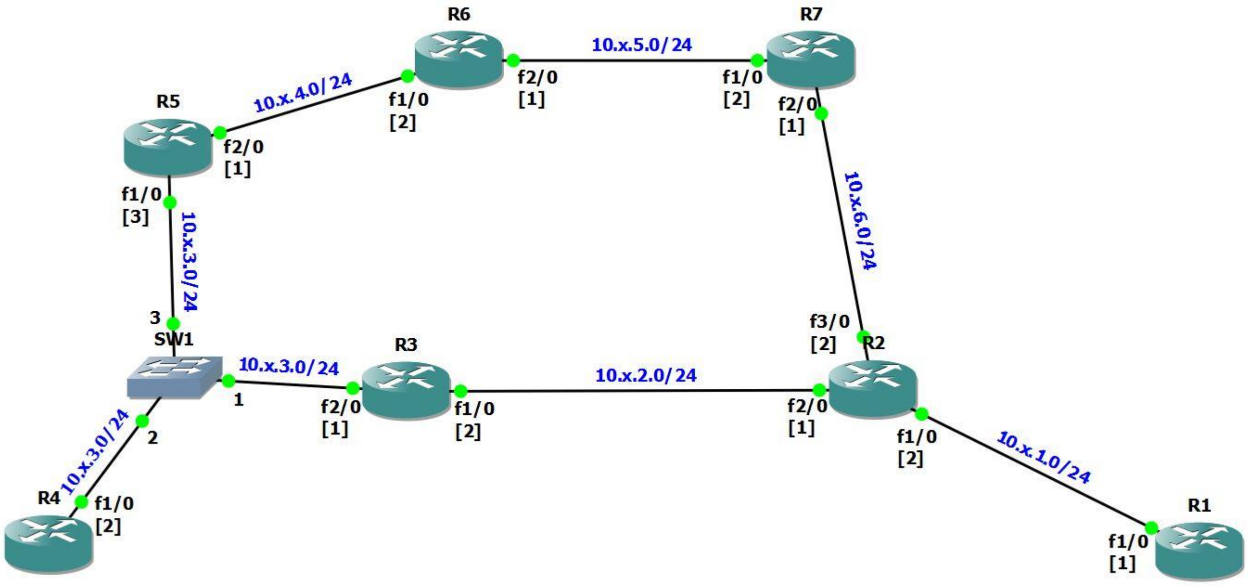
13.14. Observe the ARP query on each link and describe, which PC/Router looking for whom and whom answers back.
Pay attention to the MAC address in order to indicate the specific component.

13.15. Explain the difference between the ARP requests on the different links and how it is related to the default routes configuration.

13.16. Explain why the ICMP packet did not reach the last network (192.168.2.0/24), but reached the other previous networks.

14. Topology 6 Configuration

Topology 6



Pre

14.1. The topology is based on *Topology 4*.
But first, before making any changes, save as a new project as *Topology 6*.

14.2. Note: with minor changes this is going to be main topology you will use in the lab, it is very important you save it correctly.

Do

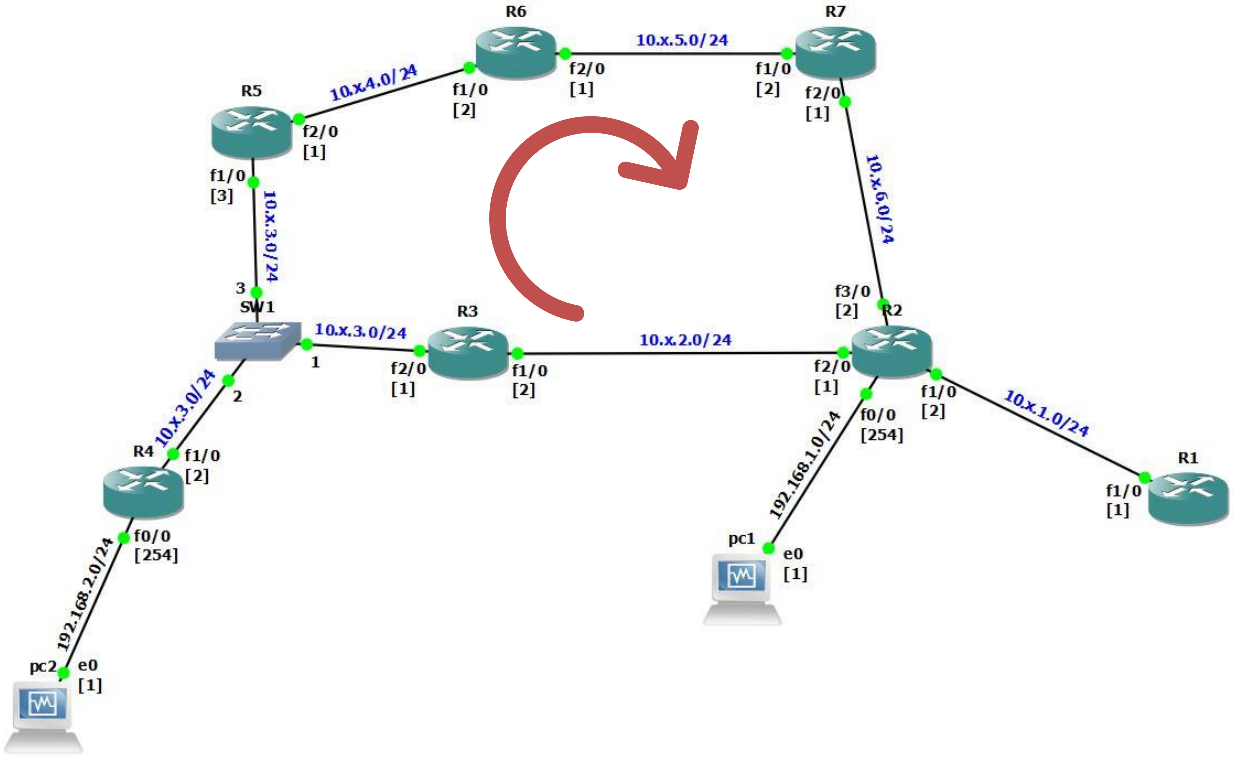
14.3. Build the topology according to the figure.

14.4. Configure the network interfaces on each router according to the figure.

14.5. Verify the connection between each pair of neighboring routers using ping.

14.6. Perform "`write`" on each router, save the topology and ZIP it.

15. Observing Routing Loops



A potential problem when setting routing tables manually is that *routing loops* may occur. In this part of the Lab you intentionally configure a *routing loop* in the configuration of the routing table and observe what happens to network traffic in such a situation.

Pre

15.1. The exercise is based on *Topology 6*.
But first, save as a new project, before making any changes.

Do

15.2. Add and configure the PCs according to the figure.

15.3. On the routers, configure the network interfaces, which connected to the PCs.

15.4. Set appropriate default gateway for each PC, according to the figure.

15.5. For each Router, configure a default route according to the table:

Router	Default Next-Hop
R2	10.x.2.2
R3	10.x.3.3
R5	10.x.4.2
R6	10.x.5.2
R7	10.x.6.2

Note that we have created *routing loop*: R2 -> R3 -> R5 -> R6 -> R7 -> R2

15.6. For example:

```
R2(config)#ip route 0.0.0.0 0.0.0.0 10.x.2.2
```

15.7. Issue a *traceroute* from PC1 to PC2 to verify that loop exists:

```
pc1% traceroute 192.168.2.1
```

Due to the need to fill first the ARP cache, probably, the first time will not be executed properly. If so, execute the command again.

15.8. Start Wireshark on *R2* interface *f0/0*, on *R2* interface *f2/0* and on *R7* interface *f2/0*.

15.9. Issue a ping command (with a single Echo request packet) from PC1 to PC2:

```
pc1% ping -c 1 192.168.2.1
```

Observe Wireshark to see that the single ICMP Echo Request packet is looping.

15.10. Stop capture and save the Wireshark pcaps to files.

Attach to your report

15.11. Explain how you can know that all packets observed in Wireshark are (most likely) the same instance of the single packet we sent using *ping*. (note: there is no "ID" field!)

15.12. Observe the IP header of the Echo request instances on the link between R2 and R3.
Which field changes between the different instances of the packet.
Explain the meaning of the difference between values of this field in a pair of sequential Wireshark entries for the packet.
What is the difference between each two sequential values you got? Explain why you would expect to get this value.

15.13. Explain why the ICMP Echo Request packet does not loop forever in the network.

15.14. Alice, a lab student suggested to set the initial value of the field you found in question 12 in this section, to 8. She claims that this will decrease the time needed to find routing loops in networks. What is the downside to this suggestion?

Good Luck!