Advanced Topics in Networking

Project 1 - Static routing

Patryk Figiel

December 20, 2024

Contents

1. Introduction

1.1. Goal of the Lab

The objective of this lab is to apply static routing and subnetting to configure a functional network. Building on Project 1, it involves creating subnets, assigning IP addresses, and configuring routes to ensure connectivity, enhancing understanding of subnetting, routing, and network management.

1.2. Preparation

Before starting the lab, an example from the presentation was reviewed to understand the task requirements and visualize the network setup. The network topology was then designed in GNS3, configuring routers, switches, and virtual machines (VMs) for the hosts. The VMs were assigned IP addresses according to the subnetting rules, and the routers were configured with appropriate interfaces and static routes. Finally, network connectivity was tested by performing ping tests between the hosts to ensure that all devices in different subnets could communicate and that the static routing was correctly set up.

2. Task I

2.1. IP Address Calculations

First name: Patryk - 6 characters
Last name: Figiel - 6 characters
Album Mati: 235417146252 - the sum is: 42

The first name and the last name yield the same sum. To differentiate, I added 1 to the last name.

Based on the above calculations, the IP addresses assigned in this task are as follows:
- **TinycoreVm:** 192.168.1.6
- **PC1:** 192.168.1.7
- **PC2:** 192.168.2.42
- **PC3:** 192.168.2.43

2.2. Network topology

The network topology is presented as follows:

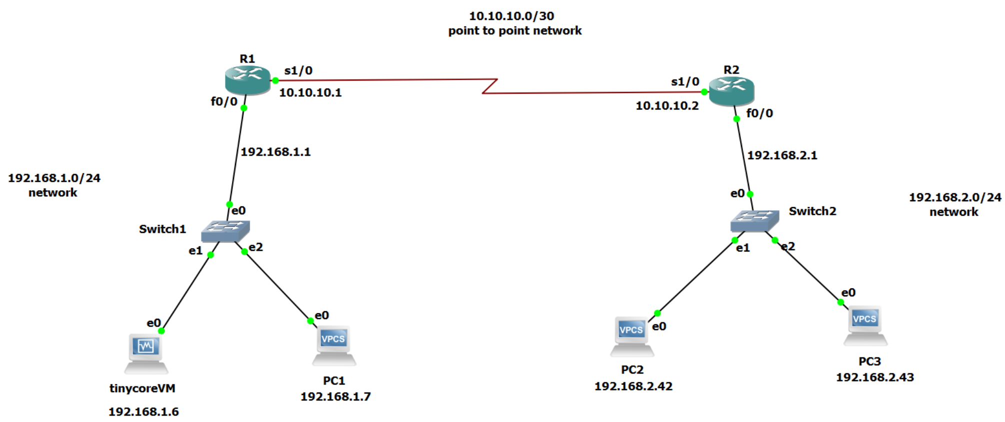
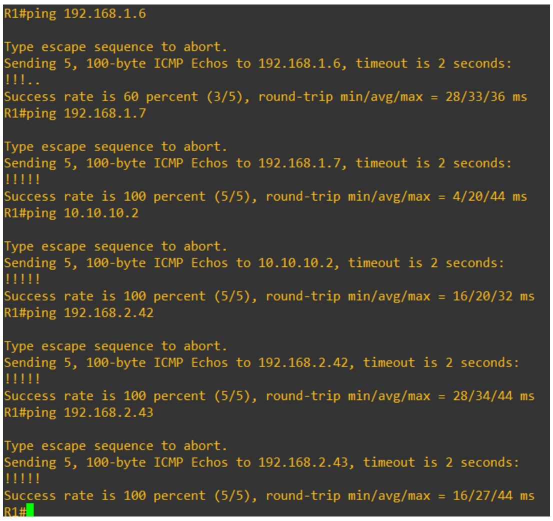


Figure 1. Network topology

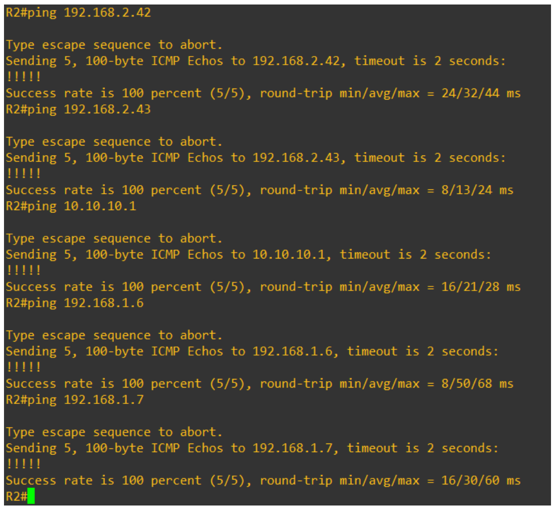
2.3. Connectivity Testing

Firstly the connection from R1 to all hosts and router R2 was checked. Ping command was used to check if there is connection to them. Then the same was done from router R2.



```
R1#ping 192.168.1.6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.6, timeout is 2 seconds:
!!!..
Success rate is 60 percent (3/5), round-trip min/avg/max = 28/33/36 ms
R1#ping 192.168.1.7

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.7, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/20/44 ms
R1#ping 10.10.10.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/20/32 ms
R1#ping 192.168.2.42

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.42, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/34/44 ms
R1#ping 192.168.2.43

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.43, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/27/44 ms
R1#
```

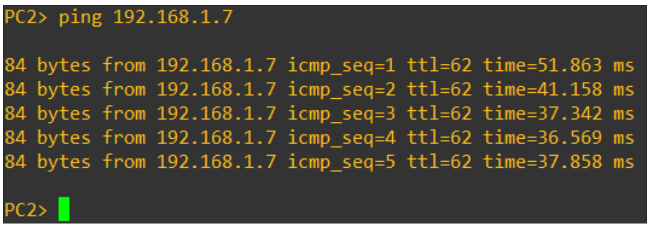
Figure 2. Connection test from router 1



```
R2#ping 192.168.2.42

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.42, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/32/44 ms
R2#ping 192.168.2.43

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.43, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/13/24 ms
R2#ping 10.10.10.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/21/28 ms
R2#ping 192.168.1.6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/50/68 ms
R2#ping 192.168.1.7

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.7, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/30/60 ms
R2#
```

Figure 3. Connection test from router 2

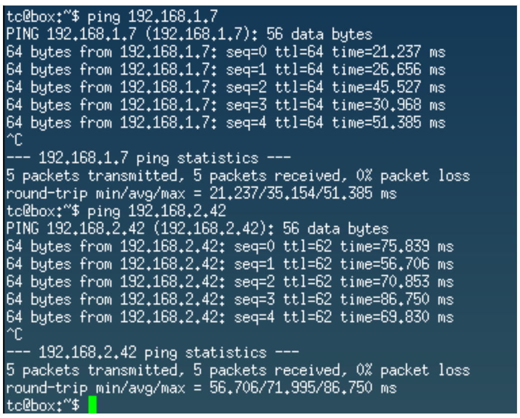
The next test was about testing the connection between PC2 and PC1 to verify the connection between the networks of one of the hosts:



```
PC2> ping 192.168.1.7

84 bytes from 192.168.1.7 icmp_seq=1 ttl=62 time=51.863 ms
84 bytes from 192.168.1.7 icmp_seq=2 ttl=62 time=41.158 ms
84 bytes from 192.168.1.7 icmp_seq=3 ttl=62 time=37.342 ms
84 bytes from 192.168.1.7 icmp_seq=4 ttl=62 time=36.569 ms
84 bytes from 192.168.1.7 icmp_seq=5 ttl=62 time=37.858 ms

PC2>
```

Figure 4. Connection test from PC2

The last test was executed from the tinycore virtual machine:



```
tc@box:~$ ping 192.168.1.7
PING 192.168.1.7 (192.168.1.7): 56 data bytes
64 bytes from 192.168.1.7: seq=0 ttl=64 time=21.237 ms
64 bytes from 192.168.1.7: seq=1 ttl=64 time=26.656 ms
64 bytes from 192.168.1.7: seq=2 ttl=64 time=45.527 ms
64 bytes from 192.168.1.7: seq=3 ttl=64 time=30.968 ms
64 bytes from 192.168.1.7: seq=4 ttl=64 time=51.385 ms
^C
--- 192.168.1.7 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 21.237/35.154/51.385 ms
tc@box:~$ ping 192.168.2.42
PING 192.168.2.42 (192.168.2.42): 56 data bytes
64 bytes from 192.168.2.42: seq=0 ttl=62 time=75.839 ms
64 bytes from 192.168.2.42: seq=1 ttl=62 time=56.706 ms
64 bytes from 192.168.2.42: seq=2 ttl=62 time=70.853 ms
64 bytes from 192.168.2.42: seq=3 ttl=62 time=86.750 ms
64 bytes from 192.168.2.42: seq=4 ttl=62 time=69.830 ms
^C
--- 192.168.2.42 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 56.706/71.995/86.750 ms
tc@box:~$
```

Figure 5. Connection test from tinycore virtual machine

As we can see all the routing paths and IP addresses were set correctly and it is possible to reach every node. Therefore, tests were executed successfully.

3. Task II

3.1. Objective

The goal of Task 2 was to reconfigure the existing network by dividing the 192.168.1.0/24 and 192.168.2.0/24 networks into smaller subnets, ensuring each subnet could accommodate at least two host devices.

3.2. Updated Network Topology

The updated topology is illustrated in the attached diagram. Key changes include:

- subdivision of the 192.168.1.0/24 and 192.168.2.0/24 networks into smaller subnets using subnetting
- adding two routers (R3 and R4) connected to R1 via separate serial interfaces for subnets under 192.168.1.0/24
- using two distinct interfaces on R2 to connect subnets under 192.168.2.0/24
- integration of switches to interconnect VPCs and TinyCore VM hosts with their respective gateways

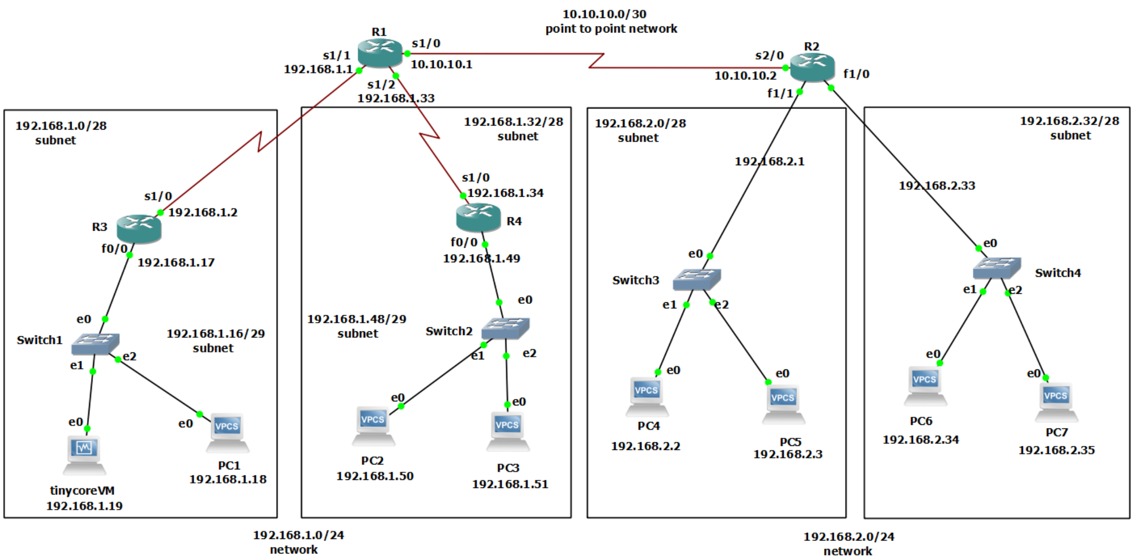


Figure 6. Topology for task 2

3.3. Subnetting Details

The original networks were divided as follows:

192.168.1.0/24 Network

- subnet 1 (192.168.1.0/28): used for hosts connected via R3
- subnet 2 (192.168.1.16/29): connected to R3 for TinyCore VM and PC1
- subnet 3 (192.168.1.32/28): used for hosts connected via R4
- subnet 4 (192.168.1.48/29): connected to R4 for PC2 and PC3

192.168.2.0/24 Network

- subnet 1 (192.168.2.0/28): hosts PC4 and PC5, connected to R2 via interface f1/0
- subnet 2 (192.168.2.32/28): hosts PC6 and PC7, connected to R2 via interface f1/1

Point-to-Point Network

- 10.10.10.0/30: serial connection between R1 and R2

3.4. Connectivity Testing

Ping tests were conducted between devices in different subnets to verify end-to-end communication. All tests were successful, indicating that routing and subnetting were properly implemented. Below I present example tests performed from tinycoreVM and PC4 to all subnets in topolgy.

TinycoreVM and PC4 tests

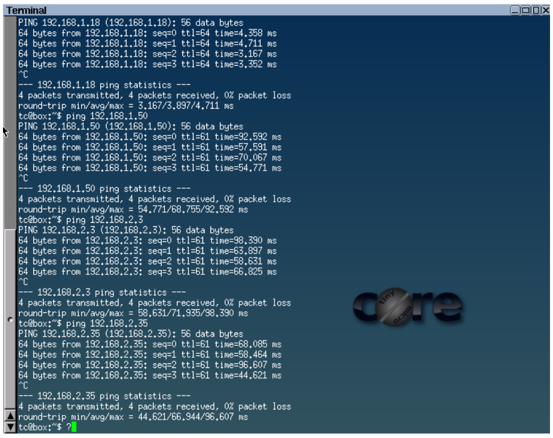


Figure 7. Connection tests from TinyCoreVM

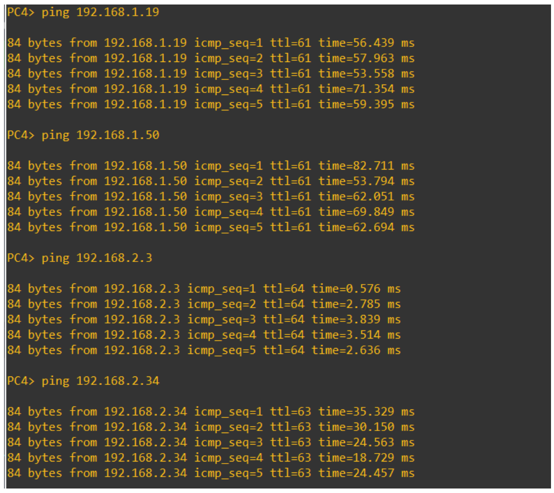


Figure 8. Connection tests from PC4

3.5. Routing tables

Router R1:

Network	Subnet Mask	Next Hop/Interface
192.168.1.0	255.255.255.240	Connected via s1/1
192.168.1.32	255.255.255.240	Connected via s1/2
192.168.2.0	255.255.255.0	10.10.10.2 (via s1/0)
10.10.10.0	255.255.255.252	Connected via s1/0

Router R2:

Network	Subnet Mask	Next Hop/Interface
192.168.2.0	255.255.255.240	Connected via f1/1
192.168.2.32	255.255.255.240	Connected via f1/0
192.168.1.0	255.255.255.0	10.10.10.1 (via s2/0)
10.10.10.0	255.255.255.252	Connected via s2/0

Router R3:

Network	Subnet Mask	Next Hop/Interface
192.168.1.0	255.255.255.240	Connected via s1/0
192.168.1.16	255.255.255.248	Connected via f0/0
192.168.2.0	255.255.255.0	192.168.1.33 (via s1/0)

Router R4:

Network	Subnet Mask	Next Hop/Interface
192.168.1.32	255.255.255.240	Connected via s1/0
192.168.1.48	255.255.255.248	Connected via f0/0
192.168.2.0	255.255.255.0	192.168.1.34 (via s1/0)
192.168.1.0	255.255.255.0	192.168.1.34 (via s1/0)

3.6. Encapsulation/Decapsulation Processes in a Router

Decapsulation: The router removes the Layer 2 (Data Link Layer) header and trailer from the incoming frame to expose the Layer 3 (Network Layer) packet. This allows the router to process the IP header.
Encapsulation: After determining the next hop, the router encapsulates the packet in a new Layer 2 frame. The source MAC address is set to the router's outgoing interface, and the destination MAC address is set to the next-hop device.

4. Summary

In this report, I successfully implemented static routing and subnetting. I configured a network, verified connectivity, and subdivided networks into smaller subnets. All tests confirmed proper routing and connectivity, demonstrating key networking concepts.