

Лабораторная работа №5

Отчет

Лисовская Арина Валерьевна

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
3.1	Моделирование сети на базе коммутатора	6
3.2	Моделирование сети на базе маршрутизатора FRR	11
3.3	Моделирование сети на базе маршрутизатора VyOS	14
4	Выводы	19

Список иллюстраций

3.1	Топология сети с коммутатором	6
3.2	Настройка IP-адресации на PC2	7
3.3	Настройка IP-адресации на PC1	8
3.4	Проверка связи с PC1 на PC2	9
3.5	Проверка связи с PC2 на PC1	9
3.6	Окно захвата трафика в Wireshark	10
3.7	Детальный анализ пакета	10
3.8	Топология сети с маршрутизатором FRR	11
3.9	Настройка интерфейсов маршрутизатора FRR	11
3.10	Просмотр текущей конфигурации FRR	12
3.11	Проверка статуса интерфейсов FRR	12
3.12	Пинг со шлюзом FRR	13
3.13	Анализ ARP трафика при работе с FRR	14
3.14	Топология сети с маршрутизатором VyOS	14
3.15	Проверка настроек PC1	15
3.16	Настройка интерфейса eth0 в VyOS	16
3.17	Просмотр конфигурации VyOS	17
3.18	Проверка связи с маршрутизатором VyOS	18
3.19	Анализ ICMP и ARP трафика с VyOS	18

1 Цель работы

Построение простейших моделей сети на базе коммутатора и маршрутизаторов FRR и VyOS в GNS3, анализ трафика посредством Wireshark.

2 Задание

1. Моделирование простейшей сети на базе коммутатора в GNS3.
2. Анализ трафика в GNS3 посредством Wireshark (ARP, ICMP).
3. Моделирование простейшей сети на базе маршрутизатора FRR в GNS3.
4. Моделирование простейшей сети на базе маршрутизатора VyOS в GNS3.

3 Выполнение лабораторной работы

3.1 Моделирование сети на базе коммутатора

В начале работы я создаю топологию сети в GNS3, состоящую из двух виртуальных ПК (VPCS) и одного коммутатора Ethernet. Устройства соединяются интерфейсами, и я запускаю эмуляцию (рис. 3.1).

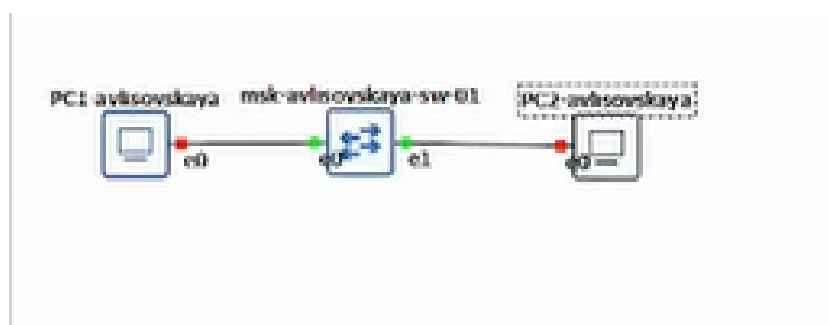
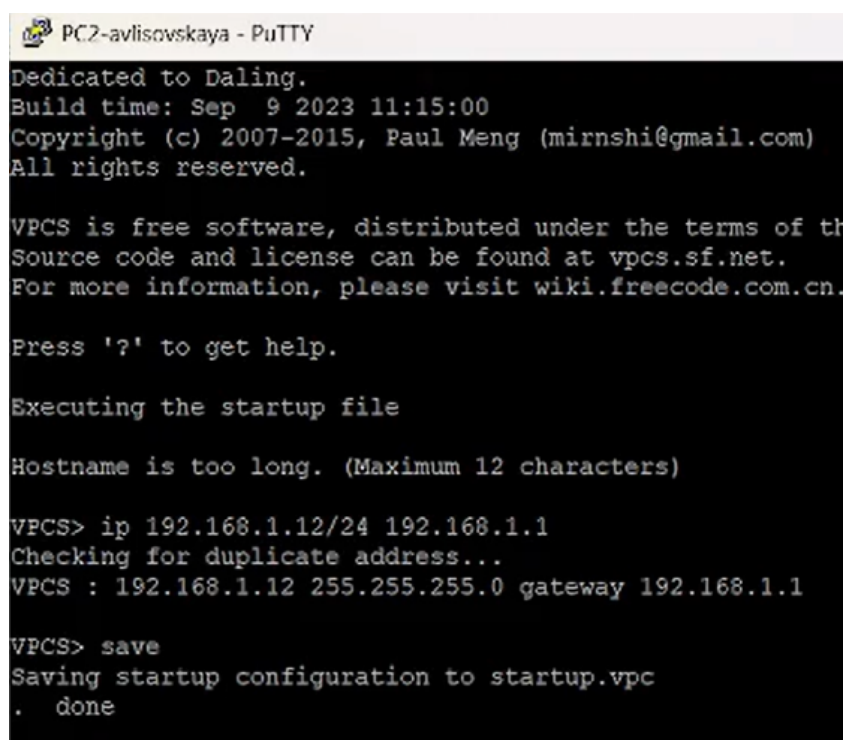


Рис. 3.1: Топология сети с коммутатором

Далее перехожу к настройке второго оконечного устройства (PC2). В терминале PuTTY выполняется процесс загрузки эмулятора VPCS. Командой `ip 192.168.1.12/24 192.168.1.1` присваиваю устройству IP-адрес 192.168.1.12 в сети 192.168.1.0/24 с указанием шлюза. Система проверяет адрес на уникальность, после чего я сохраняю конфигурацию командой `save` (рис. 3.2).



```
PC2-avliskovskaya - PuTTY
Dedicated to Daling.
Build time: Sep 9 2023 11:15:00
Copyright (c) 2007-2015, Paul Meng (mirnshi@gmail.com)
All rights reserved.

VPCS is free software, distributed under the terms of the GPL.
Source code and license can be found at vpcs.sf.net.
For more information, please visit wiki.freecode.com.cn.

Press '?' to get help.

Executing the startup file

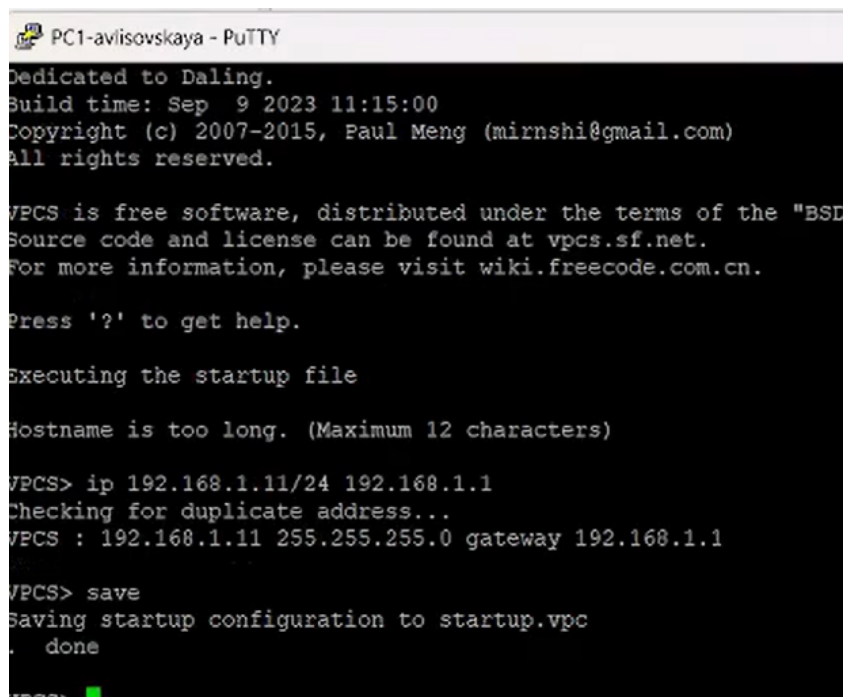
Hostname is too long. (Maximum 12 characters)

VPCS> ip 192.168.1.12/24 192.168.1.1
Checking for duplicate address...
VPCS : 192.168.1.12 255.255.255.0 gateway 192.168.1.1

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

Рис. 3.2: Настройка IP-адресации на PC2

Аналогичную настройку выполняю для первого виртуального ПК (PC1). Ввожу команду `ip 192.168.1.11/24 192.168.1.1`, присваивая адрес 192.168.1.11. Это завершает базовую конфигурацию сети из двух ПК и коммутатора, конфигурация также сохраняется (рис. 3.3).



```
PC1-avisoyskaya - PuTTY
Dedicated to Daling.
Build time: Sep 9 2023 11:15:00
Copyright (c) 2007-2015, Paul Meng (mirnshi@gmail.com)
All rights reserved.

VPCS is free software, distributed under the terms of the "BSD
Source code and license can be found at vpcs.sf.net.
For more information, please visit wiki.freecode.com.cn.

Press '?' to get help.

Executing the startup file

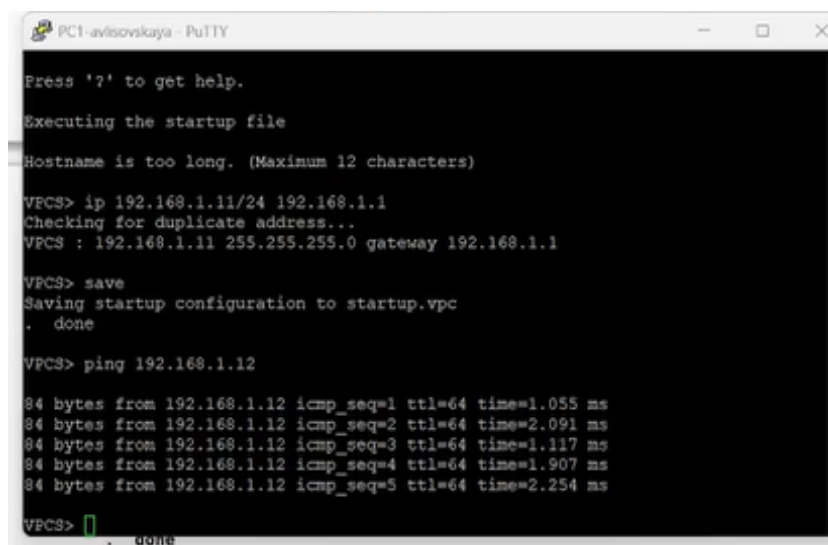
Hostname is too long. (Maximum 12 characters)

VPCS> ip 192.168.1.11/24 192.168.1.1
Checking for duplicate address...
VPCS : 192.168.1.11 255.255.255.0 gateway 192.168.1.1

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

Рис. 3.3: Настройка IP-адресации на PC1

После настройки обоих ПК выполняю проверку работоспособности сети с первого узла. Команда `ping 192.168.1.12`, отправленная с PC1 (192.168.1.11) на PC2, получает успешные ответы. В выводе видны метрики `ttl=64` и время задержки около 1 мс, что свидетельствует о прямой связи в одной подсети через коммутатор (рис. 3.4).



```
PC1-avilovskaya - PuTTY

Press '?' to get help.

Executing the startup file

Hostname is too long. (Maximum 12 characters)

VPCS> ip 192.168.1.11/24 192.168.1.1
Checking for duplicate address...
VPCS : 192.168.1.11 255.255.255.0 gateway 192.168.1.1

VPCS> save
Saving startup configuration to startup.vpc
. done

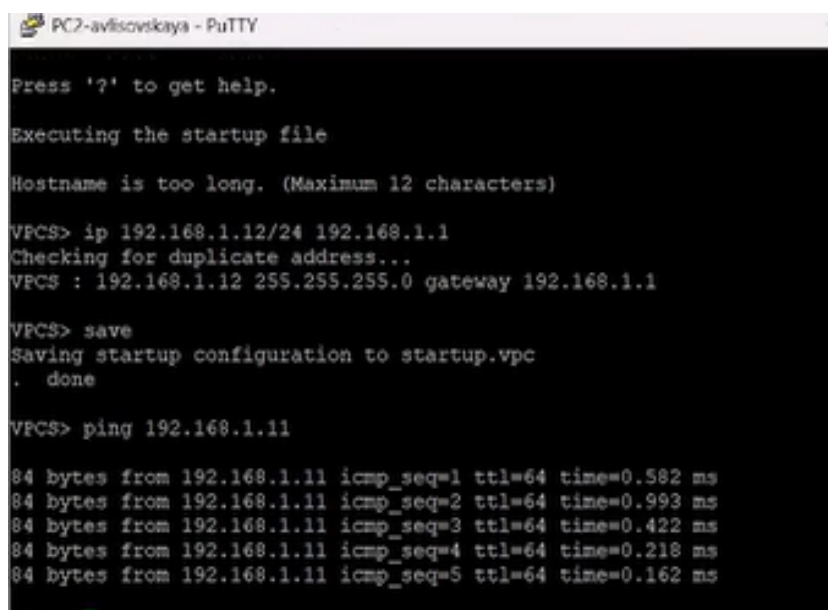
VPCS> ping 192.168.1.12

84 bytes from 192.168.1.12 icmp_seq=1 ttl=64 time=1.055 ms
84 bytes from 192.168.1.12 icmp_seq=2 ttl=64 time=2.091 ms
84 bytes from 192.168.1.12 icmp_seq=3 ttl=64 time=1.117 ms
84 bytes from 192.168.1.12 icmp_seq=4 ttl=64 time=1.907 ms
84 bytes from 192.168.1.12 icmp_seq=5 ttl=64 time=2.254 ms

VPCS> . done
```

Рис. 3.4: Проверка связи с PC1 на PC2

Затем проверяю связь в обратном направлении. В терминале второго ПК (PC2) ввожу команду `ping 192.168.1.11`. Пинг проходит успешно, что подтверждает корректную двустороннюю связь и правильную работу коммутации кадров (рис. 3.5).



```
PC2-avilovskaya - PuTTY

Press '?' to get help.

Executing the startup file

Hostname is too long. (Maximum 12 characters)

VPCS> ip 192.168.1.12/24 192.168.1.1
Checking for duplicate address...
VPCS : 192.168.1.12 255.255.255.0 gateway 192.168.1.1

VPCS> save
Saving startup configuration to startup.vpc
. done

VPCS> ping 192.168.1.11

84 bytes from 192.168.1.11 icmp_seq=1 ttl=64 time=0.582 ms
84 bytes from 192.168.1.11 icmp_seq=2 ttl=64 time=0.993 ms
84 bytes from 192.168.1.11 icmp_seq=3 ttl=64 time=0.422 ms
84 bytes from 192.168.1.11 icmp_seq=4 ttl=64 time=0.218 ms
84 bytes from 192.168.1.11 icmp_seq=5 ttl=64 time=0.162 ms
```

Рис. 3.5: Проверка связи с PC2 на PC1

Для анализа трафика я запускаю захват пакетов в Wireshark на линке между

PC1 и коммутатором. В окне анализатора отображаются перехваченные кадры, включая протоколы ARP и ICMPv6, что позволяет детально изучить процесс сетевого взаимодействия (рис. 3.6).

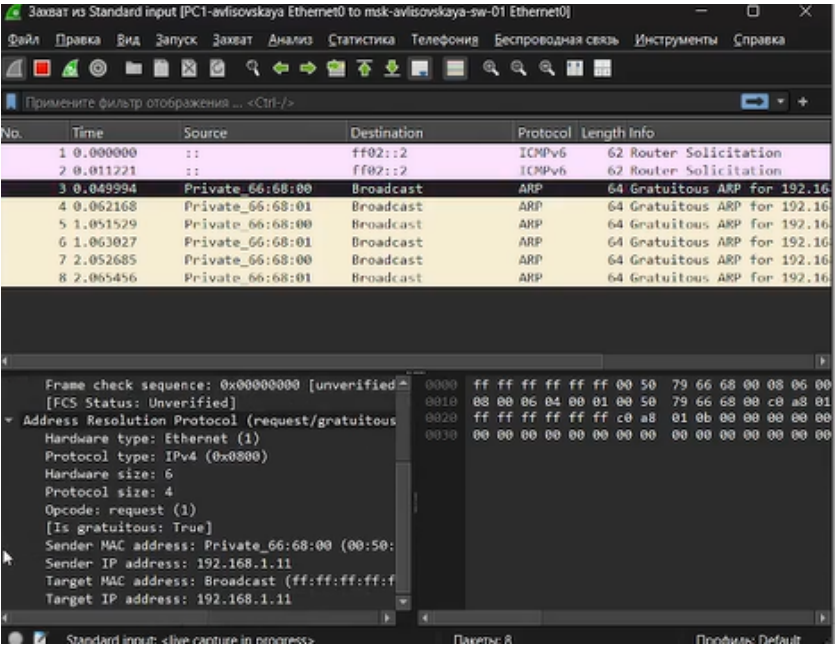


Рис. 3.6: Окно захвата трафика в Wireshark

Также просматриваю детальную информацию о захваченном пакете в нижней панели Wireshark. Здесь отображается шестнадцатеричный дамп кадра и расшифровка заголовков уровней модели OSI (рис. 3.7).

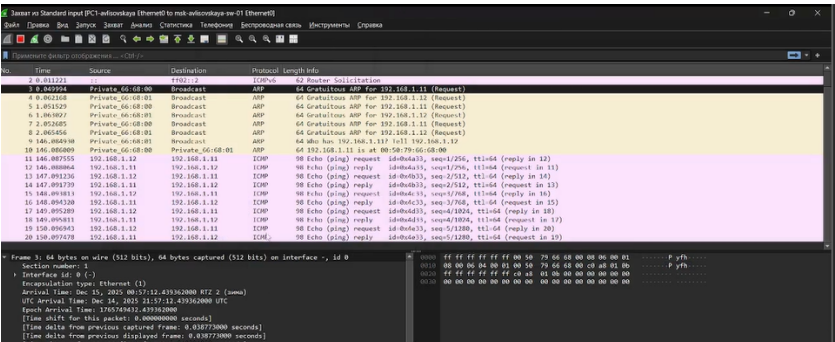


Рис. 3.7: Детальный анализ пакета

3.2 Моделирование сети на базе маршрутизатора FRR

Перехожу к выполнению задания с маршрутизатором FRR. Строю в GNS3 новую топологию, включающую PC1, коммутатор и маршрутизатор FRR (msk-user-gw-01), соединенные соответствующими интерфейсами (рис. 3.8).



Рис. 3.8: Топология сети с маршрутизатором FRR

Выполняю базовую конфигурацию маршрутизатора FRR через консоль. Вхожу в режим конфигурации (`configure terminal`), меняю имя хоста на `msk-avlisovskaya-gw-01`. Затем настраиваю интерфейс `eth0`: назначаю IP-адрес `192.168.1.1/24` и включаю интерфейс командой `no shutdown`. Конфигурация сохраняется командой `write memory` (рис. 3.9).

```
msk-avlisovskaya-gw-01 - PuTTY
Started watchfrr
* Starting sshd ... [ ok ]

Hello, this is FRRouting (version 8.2.2).
Copyright 1996-2005 Kunihiko Ishiguro, et al.

frr# configure terminal
frr(config)# hostname msk-avlisovskaya-gw-01
msk-avlisovskaya-gw-01(config)# exit
msk-avlisovskaya-gw-01# write memory
Note: this version of vtysh never writes vtysh.conf
Building Configuration...
Integrated configuration saved to /etc/frr/frr.conf
[OK]
msk-avlisovskaya-gw-01# configure terminal
msk-avlisovskaya-gw-01(config)# interface eth0
msk-avlisovskaya-gw-01(config-if)# ip address 192.168.1.1/24
msk-avlisovskaya-gw-01(config-if)# no shutdown
msk-avlisovskaya-gw-01(config-if)# exit
msk-avlisovskaya-gw-01(config)# exit
msk-avlisovskaya-gw-01# write memory
Note: this version of vtysh never writes vtysh.conf
```

Рис. 3.9: Настройка интерфейсов маршрутизатора FRR

Проверяю текущую конфигурацию устройства командой `show running-`

config. В выводе видно, что имя хоста успешно изменено, а интерфейсу eth0 присвоен корректный IP-адрес (рис. 3.10).

```
msk-avlisovskaya-gw-01# show running-config
Building configuration...

Current configuration:
!
frr version 8.2.2
frr defaults traditional
hostname frr
hostname msk-avlisovskaya-gw-01
service integrated-vtysh-config
!
interface eth0
 ip address 192.168.1.1/24
exit
!
end
msk-avlisovskaya-gw-01#
```

Рис. 3.10: Просмотр текущей конфигурации FRR

Дополнительно проверяю состояние интерфейсов командой `show interface brief`. Интерфейс eth0 находится в статусе up, ему присвоен адрес 192.168.1.1/24, что подтверждает его готовность к работе (рис. 3.11).

```
msk-avlisovskaya-gw-01# show interface brief
Interface      Status VRF      Addresses
-----
eth0           up    default  192.168.1.1/24
eth1           down  default
eth2           down  default
eth3           down  default
eth4           down  default
eth5           down  default
eth6           down  default
eth7           down  default
lo             up    default
pimreg        up    default
msk-avlisovskaya-gw-01#
```

Рис. 3.11: Проверка статуса интерфейсов FRR

После настройки маршрутизатора проверяю связь между PC1 и шлюзом. С узла PC1 (192.168.1.10) выполняю `ping 192.168.1.1`. Успешные ответы подтверждают, что маршрутизатор отвечает на интерфейсе eth0, и стек TCP/IP настроен верно (рис. 3.12).

```
PC1-avtsovskaya - PuTTY
Copyright (c) 2007-2015, Paul Meng (mirnshi@gmail.com)
All rights reserved.

VPCS is free software, distributed under the terms of the "BSD"
Source code and license can be found at vpcs.sf.net.
For more information, please visit wiki.freecode.com.cn.

Press '?' to get help.

Executing the startup file

Checking for duplicate address...
VPCS : 192.168.1.10 255.255.255.0 gateway 192.168.1.1

VPCS> ping 192.168.1.1

84 bytes from 192.168.1.1 icmp_seq=1 ttl=64 time=15.380 ms
84 bytes from 192.168.1.1 icmp_seq=2 ttl=64 time=8.598 ms
84 bytes from 192.168.1.1 icmp_seq=3 ttl=64 time=8.797 ms
84 bytes from 192.168.1.1 icmp_seq=4 ttl=64 time=6.608 ms
84 bytes from 192.168.1.1 icmp_seq=5 ttl=64 time=3.683 ms

VPCS>
```

Рис. 3.12: Пинг со шлюзом FRR

Анализирую трафик на соединении с маршрутизатором. В Wireshark видны пакеты ARP (Gratuitous ARP от PC1) и запросы маршрутизатора (Router Solicitation). Это демонстрирует процессы разрешения адресов на канальном уровне перед началом обмена IP-пакетами (рис. 3.13).

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	::	ff02::2	ICMPv6	62	Router Solicitation
2	0.050956	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192..
3	1.052063	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192..
4	2.054846	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192..
5	23.906539	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192..
6	39.751245	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192..
7	40.747938	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192..
8	150.817197	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192..
9	182.198107	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192..
10	204.072709	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192..

Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface	
Ethernet II, Src: Private_66:68:00 (00:50:79:66:00:00), Dst: ff02::2 (01:00:5e:00:00:02)	0000 33 33 00 00 00 02 00 50 79 66 68 00 86 dd
Internet Protocol Version 6, Src: ::, Dst: ff02::2	0010 00 00 00 08 3a ff 00 00 00 00 00 00 00
Internet Control Message Protocol v6	0020 00 00 00 00 00 00 ff 02 00 00 00 00 00
	0030 00 00 00 00 00 02 85 00 7b b8 00 00 00 00

Рис. 3.13: Анализ ARP трафика при работе с FRR

3.3 Моделирование сети на базе маршрутизатора VyOS

Перехожу к последней части работы — настройке сети с маршрутизатором VyOS. Схема сети аналогична предыдущей, но в качестве шлюза используется образ VyOS (рис. 3.14).

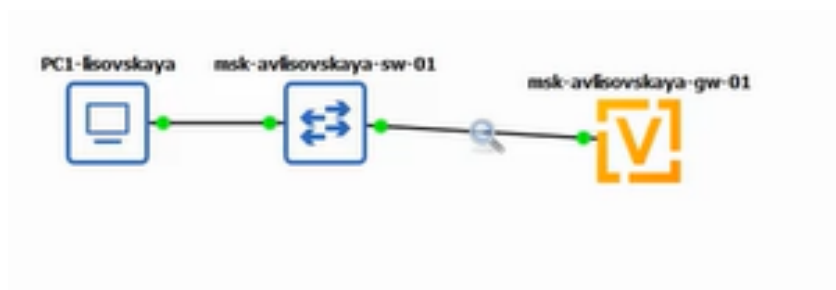
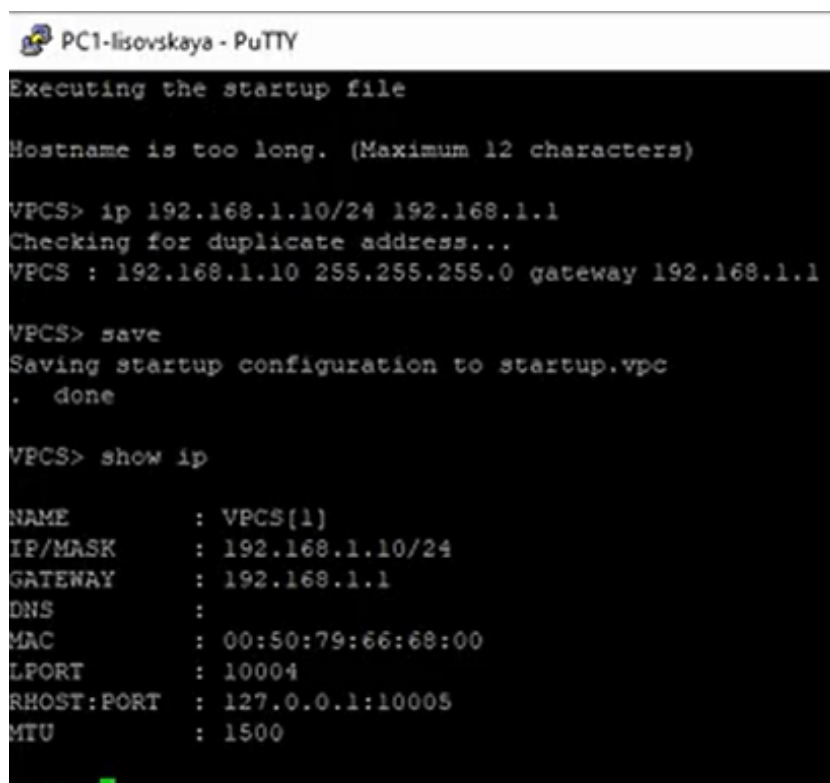


Рис. 3.14: Топология сети с маршрутизатором VyOS

Подготавливаю оконечное устройство PC1. Назначаю ему IP-адрес 192.168.1.10 и проверяю настройки командой `show ip`. Вывод подтверждает корректность IP-адреса, маски и шлюза (рис. 3.15).



```
PC1-lisovskaya - PuTTY
Executing the startup file

Hostname is too long. (Maximum 12 characters)

VPCS> ip 192.168.1.10/24 192.168.1.1
Checking for duplicate address...
VPCS : 192.168.1.10 255.255.255.0 gateway 192.168.1.1

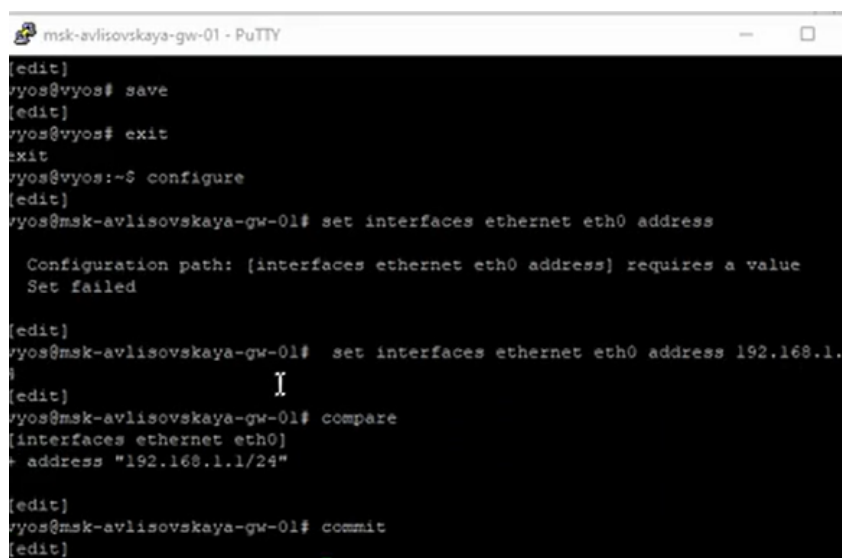
VPCS> save
Saving startup configuration to startup.vpc
. done

VPCS> show ip

NAME       : VPCS[1]
IP/MASK    : 192.168.1.10/24
GATEWAY    : 192.168.1.1
DNS        :
MAC        : 00:50:79:66:68:00
LPORT     : 10004
RHOST:PORT : 127.0.0.1:10005
MTU        : 1500
```

Рис. 3.15: Проверка настроек PC1

Выполняю настройку интерфейса маршрутизатора VyOS. В режиме конфигурации использую команду `set interfaces ethernet eth0 address 192.168.1.1/24`. После этого проверяю изменения через `compare` и применяю их ключевой командой `commit`, так как в VyOS изменения не вступают в силу без подтверждения (рис. 3.16).



```
msh-avlisovskaya-gw-01 - PuTTY
[edit]
vyos@vyos# save
[edit]
vyos@vyos# exit
exit
vyos@vyos:~$ configure
[edit]
vyos@msh-avlisovskaya-gw-01# set interfaces ethernet eth0 address

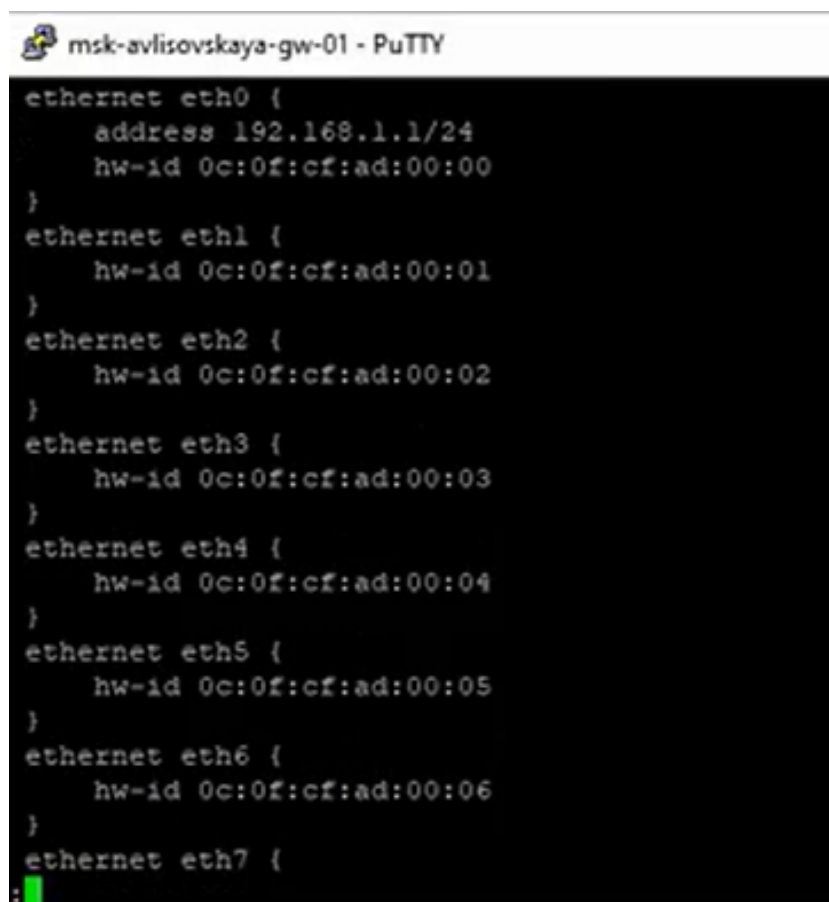
Configuration path: [interfaces ethernet eth0 address] requires a value
Set failed

[edit]
vyos@msh-avlisovskaya-gw-01# set interfaces ethernet eth0 address 192.168.1.
[edit]
vyos@msh-avlisovskaya-gw-01# compare
[interfaces ethernet eth0]
+ address "192.168.1.1/24"

[edit]
vyos@msh-avlisovskaya-gw-01# commit
[edit]
```

Рис. 3.16: Настройка интерфейса eth0 в VyOS

Просматриваю итоговую конфигурацию VyOS. В секции ethernet eth0 прописан адрес 192.168.1.1/24, также видны физические адреса интерфейсов. Это подтверждает успешное применение настроек (рис. 3.17).



```
msh-avlisovskaya-gw-01 - PuTTY
ethernet eth0 {
    address 192.168.1.1/24
    hw-id 0c:0f:cf:ad:00:00
}
ethernet eth1 {
    hw-id 0c:0f:cf:ad:00:01
}
ethernet eth2 {
    hw-id 0c:0f:cf:ad:00:02
}
ethernet eth3 {
    hw-id 0c:0f:cf:ad:00:03
}
ethernet eth4 {
    hw-id 0c:0f:cf:ad:00:04
}
ethernet eth5 {
    hw-id 0c:0f:cf:ad:00:05
}
ethernet eth6 {
    hw-id 0c:0f:cf:ad:00:06
}
ethernet eth7 {
```

Рис. 3.17: Просмотр конфигурации VyOS

Выполняю финальную проверку связи. С PC1 запускаю пинг до шлюза VyOS (192.168.1.1). Ответы приходят с минимальной задержкой (time=0.509 ms), что свидетельствует о правильной настройке всей сетевой схемы (рис. 3.18).

```
PC1-lisovskaya - PuTTY
VPCS> save
Saving startup configuration to startup.vpc
. done

VPCS> show ip

NAME       : VPCS[1]
IP/MASK     : 192.168.1.10/24
GATEWAY     : 192.168.1.1
DNS         :
MAC         : 00:50:79:66:68:00
LPORT      : 10004
RHOST:PORT  : 127.0.0.1:10005
MTU         : 1500

VPCS> ping 192.168.1.1

84 bytes from 192.168.1.1 icmp_seq=1 ttl=64 time=0.509 ms
84 bytes from 192.168.1.1 icmp_seq=2 ttl=64 time=0.969 ms
84 bytes from 192.168.1.1 icmp_seq=3 ttl=64 time=0.864 ms
84 bytes from 192.168.1.1 icmp_seq=4 ttl=64 time=0.937 ms
84 bytes from 192.168.1.1 icmp_seq=5 ttl=64 time=0.817 ms

VPCS>
```

Рис. 3.18: Проверка связи с маршрутизатором VyOS

В завершение анализирую “живой” трафик ICMP в Wireshark. Видны запросы Echo (ping) request от PC1 и ответы reply от маршрутизатора. Также зафиксирован процесс ARP (запрос MAC-адреса PC1 маршрутизатором), необходимый для инкапсуляции IP-пакета в Ethernet-кадр (рис. 3.19).

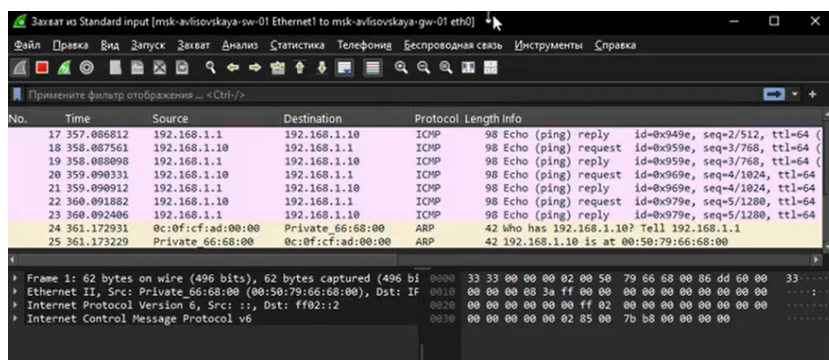


Рис. 3.19: Анализ ICMP и ARP трафика с VyOS

4 Выводы

В ходе выполнения лабораторной работы я построила и изучила простейшие модели сетей в среде GNS3. Были приобретены навыки настройки IP-адресации на виртуальных ПК (VPCS), а также базовой конфигурации программных маршрутизаторов FRR и VyOS (назначение адресов, управление интерфейсами). С помощью анализатора Wireshark я изучила структуру сетевого трафика, наблюдая работу протоколов ARP и ICMP на практике.