

# **Отчёт по лабораторной работе 5**

**Простые сети в GNS3. Анализ трафика**

Метвалли Ахмед Фарг Набеех

# **Содержание**

<b>1 Цель работы</b>	<b>5</b>
<b>2 Выполнение</b>	<b>6</b>
2.1 Моделирование простейшей сети на базе коммутатора в GNS3 . . . . .	6
2.1.1 Построение топологии и настройка IP-адресов . . . . .	6
2.1.2 Анализ трафика в Wireshark . . . . .	8
2.2 Моделирование простейшей сети на базе маршрутизатора FRR в GNS3	12
2.2.1 Построение топологии и настройка устройств . . . . .	12
2.2.2 Проверка связи и анализ трафика . . . . .	15
2.3 Моделирование простейшей сети на базе маршрутизатора VyOS в GNS3 . . . . .	17
2.3.1 Построение топологии и настройка устройств . . . . .	17
2.3.2 Проверка связи и анализ трафика . . . . .	20
<b>3 Заключение</b>	<b>22</b>

# Список иллюстраций

2.1 Топология сети в GNS3 . . . . .	6
2.2 Просмотр списка команд VPCS . . . . .	7
2.3 Настройка IP и проверка связи между ПК . . . . .	7
2.4 Захват ARP-пакетов в Wireshark . . . . .	8
2.5 ICMP-эхо-запрос и ответ . . . . .	9
2.6 UDP-эхо-запрос в Wireshark . . . . .	10
2.7 TCP-эхо-запрос и установка соединения . . . . .	11
2.8 Проверка различных режимов ping в VPCS . . . . .	12
2.9 Топология сети с маршрутизатором FRR . . . . .	13
2.10 Настройка IP-адреса для PC1 . . . . .	13
2.11 Настройка маршрутизатора FRR . . . . .	14
2.12 Проверка конфигурации FRR . . . . .	15
2.13 Проверка связи между PC1 и маршрутизатором . . . . .	16
2.14 Захват ICMP и ARP пакетов в Wireshark . . . . .	17
2.15 Топология сети с маршрутизатором VyOS . . . . .	18
2.16 Настройка маршрутизатора VyOS . . . . .	19
2.17 Просмотр интерфейсов VyOS . . . . .	20
2.18 Проверка связи между ПК и маршрутизатором . . . . .	20
2.19 Анализ ICMP и ARP пакетов в Wireshark . . . . .	21

# **Список таблиц**

# **1 Цель работы**

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

## 2 Выполнение

### 2.1 Моделирование простейшей сети на базе коммутатора в GNS3

#### 2.1.1 Построение топологии и настройка IP-адресов

1. В приложении **GNS3** создан новый проект и размещены устройства: коммутатор **msk-ahmedfarg-sw-01** и два виртуальных ПК – **PC1-ahmedfarg** и **PC2-ahmedfarg**.

Устройства соединены кабелями Ethernet, формируя простую локальную сеть.

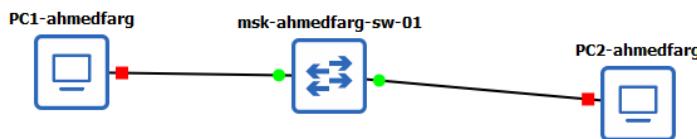


Рис. 2.1: Топология сети в GNS3

2. На каждом ПК была открыта консоль **VPCS**. Для просмотра доступных команд введена команда `?`, отобразившая список поддерживаемых операций.

```

VPCS> ?
?
arp Print help
clear ARG Shortcut for: show arp. Show arp table
dhcp [OPTION] Clear IPv4/IPv6, arp/neighbor cache, command history
disconnect Shortcut for: ip dhcp. Get IPv4 address via DHCP
echo TEXT Exit the telnet session (daemon mode)
help Display TEXT in output. See also set echo ?
history Shortcut for: show history. List the command history
ip ARG ... [OPTION] Configure the current VPC's IP settings. See ip ?
load [FILENAME] Load the configuration/script from the file FILENAME
ping HOST [OPTION ...] Ping HOST with ICMP (default) or TCP/UDP. See ping ?
quit Quit program
relay ARG ... Configure packet relay between UDP ports. See relay ?
rlogin [ip] port Telnet to port on host at ip (relative to host PC)
save [FILENAME] Save the configuration to the file FILENAME
set ARG ... Set VPC name and other options. Try set ?
show [ARG ...] Print the information of VPCs (default). See show ?
sleep [seconds] [TEXT] Print TEXT and pause running script for seconds
trace HOST [OPTION ...] Print the path packets take to network HOST
version Shortcut for: show version

To get command syntax help, please enter '?' as an argument of the command.

VPCS>

```

Рис. 2.2: Просмотр списка команд VPCS

### 3. Выполнена настройка IP-адресов:

- для **PC1-ahmedfarg**: ip 192.168.1.11/24 192.168.1.1
- для **PC2-ahmedfarg**: ip 192.168.1.12/24 192.168.1.1

После настройки параметры были сохранены командой `save`. Проверка связи с помощью `ping` показала успешный обмен ICMP-пакетами между устройствами.

```

history Shortcut for: show history. List the command history
ip ARG ... [OPTION] Configure the current VPC's IP settings. See ip ?
load [FILENAME] Load the configuration/script from the file FILENAME
ping HOST [OPTION ...] Ping HOST with ICMP (default) or TCP/UDP. See ping ?
quit Quit program
relay ARG ... Configure packet relay between UDP ports. See relay ?
rlogin [ip] port Telnet to port on host at ip (relative to host PC)
save [FILENAME] Save the configuration to the file FILENAME
set ARG ... Set VPC name and other options. Try set ?
show [ARG ...] Print the information of VPCs (default). See show ?
sleep [seconds] [TEXT] Print TEXT and pause running script for seconds
trace HOST [OPTION ...] Print the path packets take to network HOST
version Shortcut for: show version

To get command syntax help, please enter '?' as an argument of the command.

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 192.168.1.11
84 bytes from 192.168.1.11 icmp_seq=1 ttl=64 time=1.461 ms
84 bytes from 192.168.1.11 icmp_seq=2 ttl=64 time=1.472 ms
84 bytes from 192.168.1.11 icmp_seq=3 ttl=64 time=1.583 ms
84 bytes from 192.168.1.11 icmp_seq=4 ttl=64 time=2.009 ms
84 bytes from 192.168.1.11 icmp_seq=5 ttl=64 time=0.909 ms

VPCS>

```

Рис. 2.3: Настройка IP и проверка связи между ПК

## 2.1.2 Анализ трафика в Wireshark

4. Для анализа сетевого обмена запущен захват трафика на линке между **PC1** и коммутатором. В окне **Wireshark** зафиксированы ARP-пакеты, определяющие соответствие IP- и MAC-адресов.

На скриншоте видно, что оба устройства отправляют *Gratuitous ARP*-запросы с целью обновления ARP-таблиц в сети.

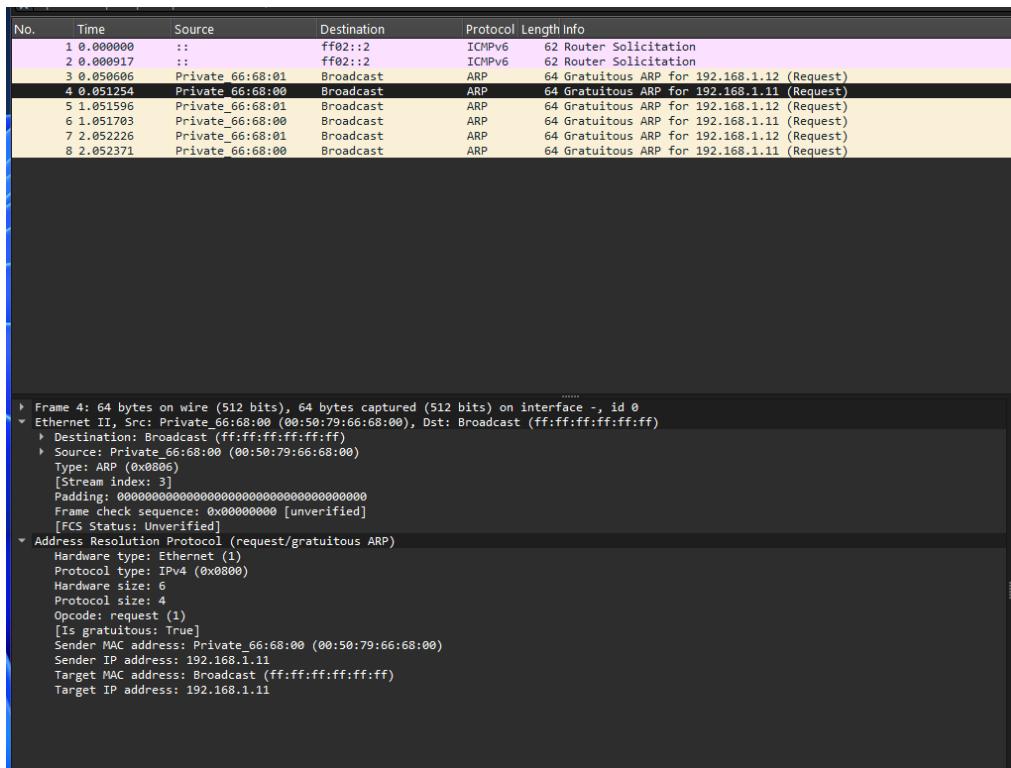


Рис. 2.4: Захват ARP-пакетов в Wireshark

5. Далее был выполнен обмен ICMP-пакетами (эхо-запрос и эхо-ответ) при выполнении команды **ping** с **PC2** на **PC1**.

На захвate видно тип пакета *Echo (ping) request/reply* и соответствующие IP-адреса источника и назначения.

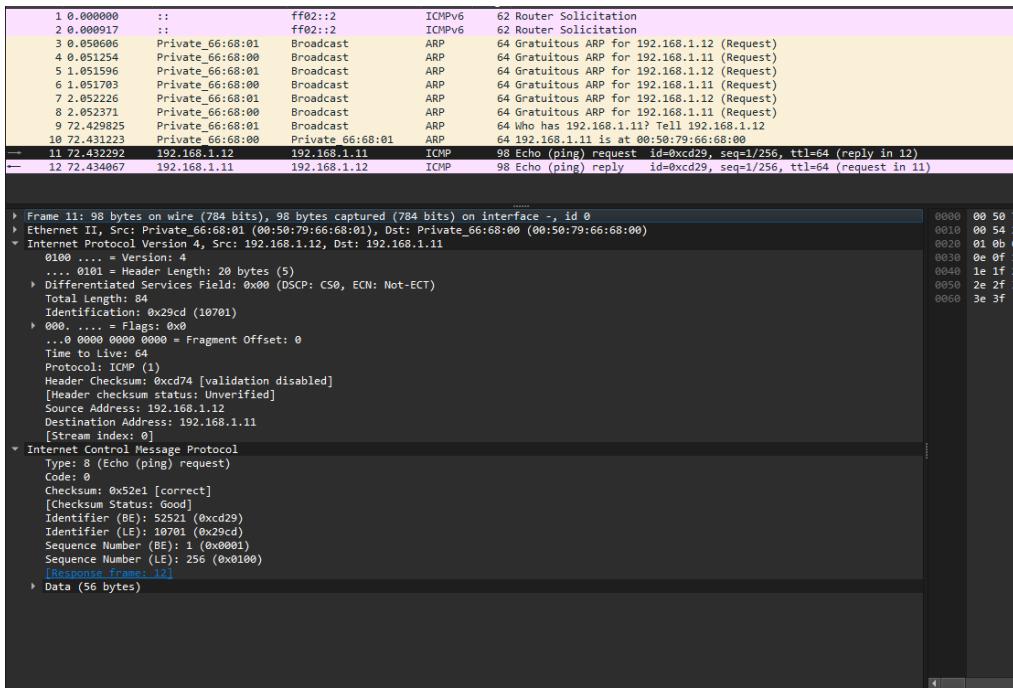


Рис. 2.5: ICMP-эхо-запрос и ответ

6. При выполнении `ping` с флагом `-2` был отправлен UDP-эхо-запрос.

Анализ показал использование протокола **UDP (port 7)** для передачи данных, при этом в поле *Payload* отображается содержимое передаваемого сообщения.

No.	Time	Source	Destination	Protocol	Length Info
1	0.000000	::	ff02::2	ICMPv6	62 Router Solicitation
2	0.000917	::	ff02::2	ICMPv6	62 Router Solicitation
3	0.050606	Private_66:68:01	Broadcast	ARP	64 Gratuitous ARP for 192.168.1.12 (Request)
4	0.051254	Private_66:68:00	Broadcast	ARP	64 Gratuitous ARP for 192.168.1.11 (Request)
5	1.051596	Private_66:68:01	Broadcast	ARP	64 Gratuitous ARP for 192.168.1.12 (Request)
6	1.051703	Private_66:68:00	Broadcast	ARP	64 Gratuitous ARP for 192.168.1.11 (Request)
7	2.052226	Private_66:68:01	Broadcast	ARP	64 Gratuitous ARP for 192.168.1.12 (Request)
8	2.052371	Private_66:68:00	Broadcast	ARP	64 Gratuitous ARP for 192.168.1.11 (Request)
9	72.429825	Private_66:68:01	Broadcast	ARP	64 Who has 192.168.1.11? Tell 192.168.1.12
10	72.431223	Private_66:68:00	Private_66:68:01	ARP	64 192.168.1.11 is at 00:50:79:66:68:00
11	72.432292	192.168.1.12	192.168.1.11	ICMP	98 Echo (ping) request id=0xcd29, seq=1/256, ttl=64 (reply in 11)
12	72.434867	192.168.1.11	192.168.1.12	ICMP	98 Echo (ping) reply id=0xcd29, seq=1/256, ttl=64 (request in 11)
13	152.549098	192.168.1.12	192.168.1.11	ECHO	98 Request
14	152.549765	192.168.1.11	192.168.1.12	ECHO	98 Response

▶ Frame 13: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface -, id 0  
 ▶ Ethernet II, Src: Private\_66:68:01 (00:50:79:66:68:01), Dst: Private\_66:68:00 (00:50:79:66:68:00)  
 ▶ Internet Protocol Version 4, Src: 192.168.1.12, Dst: 192.168.1.11  
 0100 .... = Version: 4  
 .... 0101 = Header Length: 20 bytes (5)  
 ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)  
 Total Length: 84  
 Identification: 0x2a1e (10782)  
 ▶ 000. .... = Flags: 0x0  
 ...0 0000 0000 0000 = Fragment Offset: 0  
 Time to Live: 64  
 Protocol: UDP (17)  
 Header Checksum: 0xcd13 [validation disabled]  
 [Header checksum status: Unverified]  
 Source Address: 192.168.1.12  
 Destination Address: 192.168.1.11  
 [Stream index: 0]  
 ▶ User Datagram Protocol, Src Port: 37007, Dst Port: 7  
 Source Port: 37007  
 Destination Port: 7  
 Length: 64  
 Checksum: 0x4fe5 [unverified]  
 [Checksum Status: Unverified]  
 [Stream index: 0]  
 [Stream Packet Number: 1]  
 ▶ [Timestamps]  
 UDP payload (56 bytes)  
 ▶ Echo  
 Echo data: 0050796668010e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f

Рис. 2.6: UDP-эхо-запрос в Wireshark

## 7. При использовании опции -3 команда ping выполнялась в режиме TCP.

В захваченных пакетах Wireshark зафиксированы типичные фазы установления TCP-соединения: *SYN*, *SYN-ACK* и *ACK*, подтверждающие корректную работу транспортного уровня.

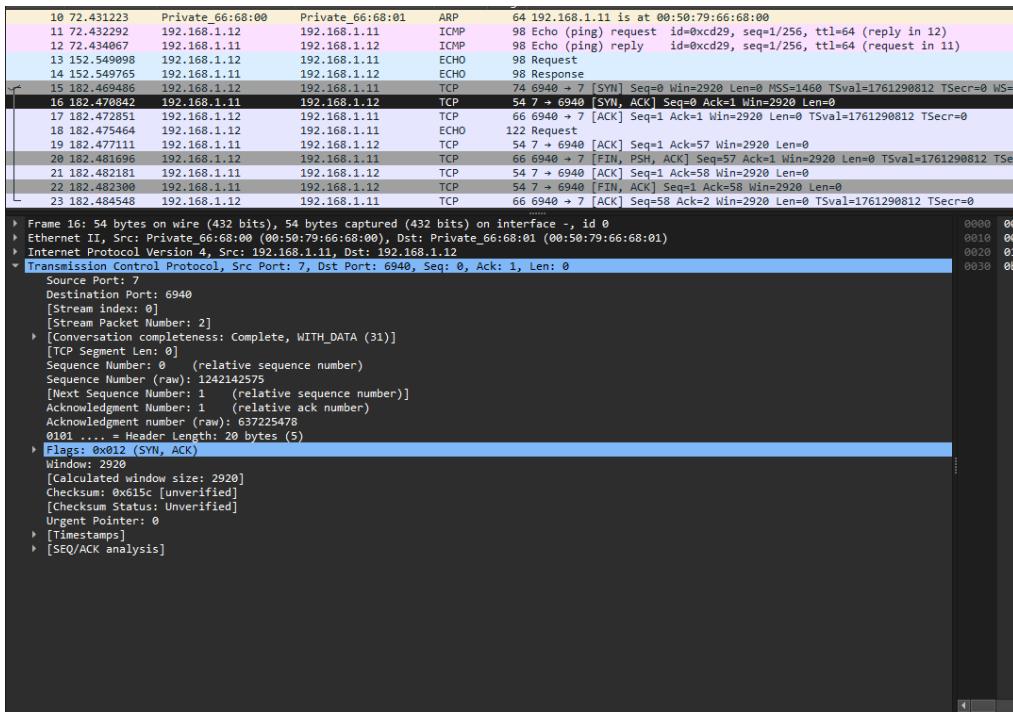


Рис. 2.7: TCP-эхо-запрос и установка соединения

8. В терминале VPCS была изучена справка по команде `ping`, где представлены возможные режимы работы: ICMP (-1), UDP (-2), TCP (-3).  
Все три режима были протестированы, что подтвердило корректное взаимодействие на уровнях **ICMP**, **UDP** и **TCP**.

```

VPCS> ping ?
ping HOST [OPTION ...]
Ping the network HOST. HOST can be an ip address or name
Options:
  -1           ICMP mode, default
  -2           UDP mode
  -3           TCP mode
  -c count   Packet count, default 5
  -D           Set the Don't Fragment bit
  -f FLAG    Tcp header FLAG |C|E|U|A|P|R|S|F|
                bits |7 6 5 4 3 2 1 0|
  -i ms      Wait ms milliseconds between sending each packet
  -l size    Data size
  -P protocol Use IP protocol in ping packets
                1 - ICMP (default), 17 - UDP, 6 - TCP
  -p port    Destination port
  -s port    Source port
  -T ttl     Set ttl, default 64
  -t           Send packets until interrupted by Ctrl+C
  -w ms      Wait ms milliseconds to receive the response

Notes: 1. Using names requires DNS to be set.
       2. Use Ctrl+C to stop the command.

VPCS> ping 192.168.1.11 -c 1
84 bytes from 192.168.1.11 icmp_seq=1 ttl=64 time=3.048 ms
VPCS> ping 192.168.1.11 -c 1 -2
84 bytes from 192.168.1.11 udp_seq=1 ttl=64 time=1.612 ms
VPCS> ping 192.168.1.11 -c 1 -3
Connect 7@192.168.1.11 seq=1 ttl=64 time=3.769 ms
SendData 7@192.168.1.11 seq=1 ttl=64 time=3.933 ms
Close    7@192.168.1.11 seq=1 ttl=64 time=2.934 ms
VPCS>

```

Рис. 2.8: Проверка различных режимов ping в VPCS

## 2.2 Моделирование простейшей сети на базе маршрутизатора FRR в GNS3

### 2.2.1 Построение топологии и настройка устройств

1. В приложении **GNS3** создан новый проект. В рабочей области размещены устройства: коммутатор **msk-ahmedfarg-sw-01**, маршрутизатор **msk-ahmedfarg-gw-01** и конечное устройство **PC1-ahmedfarg**. Все элементы соединены между собой кабелями Ethernet, формируя простейшую сеть.

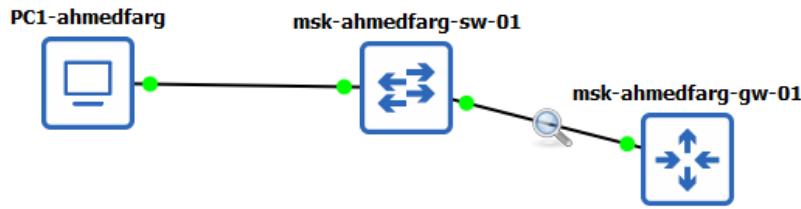


Рис. 2.9: Топология сети с маршрутизатором FRR

2. В консоли **VPCS** для узла **PC1-ahmedfarg** выполнена настройка IP-адреса:

```

ip 192.168.1.10/24 192.168.1.1
save
show ip

```

В результате устройство получило адрес 192.168.1.10/24 с шлюзом 192.168.1.1.

```

Checking for duplicate address...
VPCS : 192.168.1.11 255.255.255.0 gateway 192.168.1.1

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

VPCS>

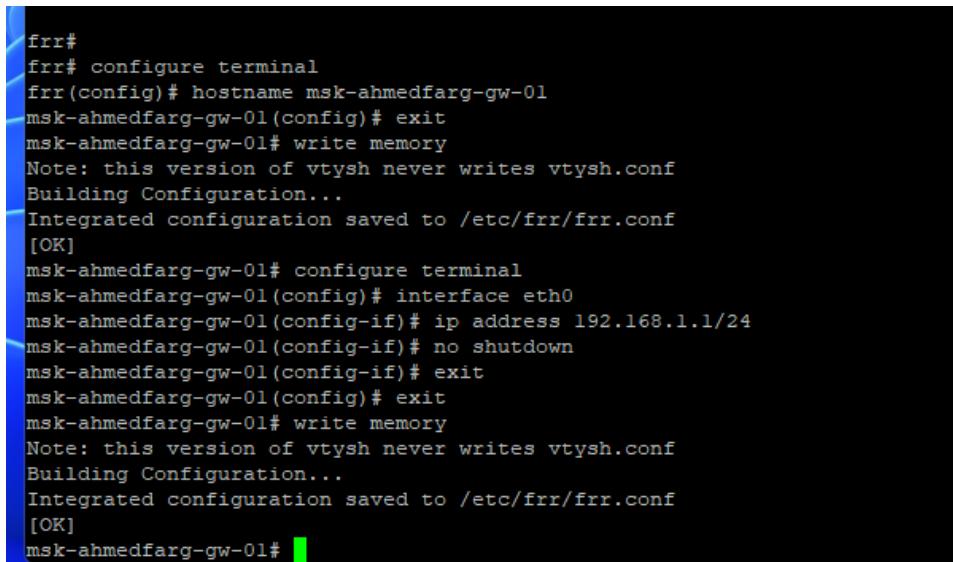
```

Рис. 2.10: Настройка IP-адреса для PC1

3. На маршрутизаторе **FRR** выполнена базовая настройка. В режиме конфигурации был изменён hostname и задан IP-адрес интерфейса eth0:

```
configure terminal  
hostname msk-ahmedfarg-gw-01  
interface eth0  
ip address 192.168.1.1/24  
no shutdown  
exit  
write memory
```

Конфигурация была сохранена и успешно применена.



```
frr#  
frr# configure terminal  
frr(config)# hostname msk-ahmedfarg-gw-01  
msk-ahmedfarg-gw-01(config)# exit  
msk-ahmedfarg-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-ahmedfarg-gw-01# configure terminal  
msk-ahmedfarg-gw-01(config)# interface eth0  
msk-ahmedfarg-gw-01(config-if)# ip address 192.168.1.1/24  
msk-ahmedfarg-gw-01(config-if)# no shutdown  
msk-ahmedfarg-gw-01(config-if)# exit  
msk-ahmedfarg-gw-01(config)# exit  
msk-ahmedfarg-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-ahmedfarg-gw-01#
```

Рис. 2.11: Настройка маршрутизатора FRR

4. Для проверки настроек маршрутизатора использованы команды:

```
show running-config  
show interface brief
```

В выводе показано, что интерфейс **eth0** активен и имеет адрес 192.168.1.1/24.

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

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

Рис. 2.12: Проверка конфигурации FRR

## 2.2.2 Проверка связи и анализ трафика

5. На **PC1-ahmedfarg** выполнена проверка связи с маршрутизатором. Команда `ping 192.168.1.1` показала успешные ответы, что подтверждает корректность настройки IP-сети.

```
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>
VPCS> ping 192.168.1.1

84 bytes from 192.168.1.1 icmp_seq=1 ttl=64 time=5.054 ms
84 bytes from 192.168.1.1 icmp_seq=2 ttl=64 time=3.336 ms
84 bytes from 192.168.1.1 icmp_seq=3 ttl=64 time=3.501 ms
84 bytes from 192.168.1.1 icmp_seq=4 ttl=64 time=4.085 ms
84 bytes from 192.168.1.1 icmp_seq=5 ttl=64 time=3.708 ms

VPCS>
```

Рис. 2.13: Проверка связи между PC1 и маршрутизатором

6. В **Wireshark** был запущен захват пакетов на линии между коммутатором и маршрутизатором.

Анализ показал обмен ICMP-пакетами *Echo Request* и *Echo Reply*, а также предварительные ARP-запросы, использовавшиеся для определения MAC-адресов.

Это подтверждает успешное взаимодействие между конечным устройством и маршрутизатором.

No.	Time	Source	Destination	Protocol	Length Info
15	247.702524	Private_66:68:00	Broadcast	ARP	64 Who has 192.168.1.1? Tell 192.168.1.10
16	247.706995	0c:be:b1:30:00:00	Private_66:68:00	ARP	60 192.168.1.1 is at 0c:be:b1:30:00:00
17	247.708986	192.168.1.10	192.168.1.1	ICMP	98 Echo (ping) request id=0xa42b, seq=1/256, ttl=64 (request in 18)
18	247.712756	192.168.1.1	192.168.1.10	ICMP	98 Echo (ping) reply id=0xa42b, seq=1/256, ttl=64 (request in 17)
19	248.714566	192.168.1.10	192.168.1.1	ICMP	98 Echo (ping) request id=0xa52b, seq=2/512, ttl=64 (reply in 20)
20	248.717031	192.168.1.10	192.168.1.10	ICMP	98 Echo (ping) reply id=0xa52b, seq=2/512, ttl=64 (request in 19)
→ 21	249.719663	192.168.1.10	192.168.1.1	ICMP	98 Echo (ping) request id=0xa62b, seq=3/768, ttl=64 (reply in 22)
→ 22	249.721816	192.168.1.1	192.168.1.10	ICMP	98 Echo (ping) reply id=0xa62b, seq=3/768, ttl=64 (request in 21)
23	250.722441	192.168.1.10	192.168.1.1	ICMP	98 Echo (ping) request id=0xa72b, seq=4/1024, ttl=64 (reply in 24)
24	250.722781	192.168.1.1	192.168.1.10	ICMP	98 Echo (ping) reply id=0xa72b, seq=4/1024, ttl=64 (request in 23)
25	251.728606	192.168.1.10	192.168.1.1	ICMP	98 Echo (ping) request id=0xa82b, seq=5/1280, ttl=64 (reply in 26)
26	251.731391	192.168.1.1	192.168.1.10	ICMP	98 Echo (ping) reply id=0xa82b, seq=5/1280, ttl=64 (request in 25)
27	252.773800	0c:be:b1:30:00:00	Private_66:68:00	ARP	60 Who has 192.168.1.10? Tell 192.168.1.1
28	252.774428	Private_66:68:00	0c:be:b1:30:00:00	ARP	60 192.168.1.10 is at 00:50:79:66:68:00
					.....
					0000 00
					0010 00
					0020 01
					0030 00
					0040 1e
					0050 2e
					0060 3e

Frame 21: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface -, id 0  
Ethernet II, Src: Private\_66:68:00 (00:50:79:66:68:00), Dst: 0c:be:b1:30:00:00 (0c:be:b1:30:00:00)  
Internet Protocol Version 4, Src: 192.168.1.10, Dst: 192.168.1.1  
 0100 .... = Version: 4  
.... 0101 = Header Length: 20 bytes (5)  
 Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)  
 Total Length: 84  
Identification: 0x2ba6 (11174)  
 000. .... = Flags: 0x0  
...0 0000 0000 0000 = Fragment Offset: 0  
 Time to Live: 64  
Protocol: ICMP (1)  
Header Checksum: 0xcb7a [validation disabled]  
[Header checksum status: Unverified]  
 Source Address: 192.168.1.10  
 Destination Address: 192.168.1.1  
[Stream index: 0]  
 Internet Control Message Protocol  
 Type: 8 (Echo (ping) request)  
Code: 0  
Checksum: 0x79dd [correct]  
[Checksum Status: Good]  
Identifier (BE): 42539 (0xa62b)  
Identifier (LE): 11174 (0x2ba6)  
Sequence Number (BE): 3 (0x0003)  
Sequence Number (LE): 768 (0x300)  
[Response frame: 22]  
 Data (56 bytes)

Рис. 2.14: Захват ICMP и ARP пакетов в Wireshark

## 2.3 Моделирование простейшей сети на базе маршрутизатора VyOS в GNS3

### 2.3.1 Построение топологии и настройка устройств

- В приложении **GNS3** создан новый проект. В рабочей области размещены устройства: коммутатор **msk-ahmedfarg-sw-01**, маршрутизатор **msk-ahmedfarg-gw-01** на базе **VyOS** и конечное устройство **PC1-ahmedfarg**. Все элементы соединены между собой кабелями Ethernet, формируя простейшую сеть.

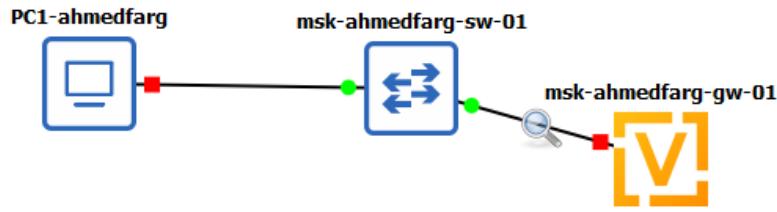


Рис. 2.15: Топология сети с маршрутизатором VyOS

2. На ПК **PC1-ahmedfarg** через консоль **VPCS** была выполнена настройка IP-адреса:

```

ip 192.168.1.10/24 192.168.1.1
save
show ip

```

В результате узлу был присвоен адрес 192.168.1.10/24, а шлюзом по умолчанию назначен 192.168.1.1.

3. На маршрутизаторе **VyOS** после входа в систему выполнен переход в режим конфигурации командой **configure**.

Затем заданы имя устройства и IP-адрес интерфейса **eth0**:

```

set system host-name msk-ahmedfarg-gw-01
set interfaces ethernet eth0 address 192.168.1.1/24
compare
commit
save

```

Изменения применены и сохранены.

```
vyos login: vyos
Password:
Welcome to VyOS!

Check out project news at https://blog.vyos.io
and feel free to report bugs at https://vyos.dev

You can change this banner using "set system login banner post-login" command.

VyOS is a free software distribution that includes multiple components,
you can check individual component licenses under /usr/share/doc/*copyright
vyos@vyos:~$ install image
You are trying to install from an already installed system. An ISO
image file to install or URL must be specified.
Exiting...
vyos@vyos:~$ configure
[edit]
vyos@vyos# system

    Invalid command: [system]

[edit]
vyos@vyos# set system host-name msk-ahmedfarg-gw-01
[edit]
vyos@vyos# set interfaces ethernet eth0 address 192.168.1.1/24
[edit]
vyos@vyos# compare
[edit interfaces ethernet eth0]
+address 192.168.1.1/24
[edit system]
>host-name msk-ahmedfarg-gw-01
[edit]
vyos@vyos#
```

Рис. 2.16: Настройка маршрутизатора VyOS

4. Для проверки корректности конфигурации маршрутизатора была выполнена команда `show interfaces`.

В выводе отображается активный интерфейс **eth0** с IP-адресом **192.168.1.1/24** и соответствующим MAC-адресом.

```
vyos@vyos#  
[edit]  
vyos@vyos# show interfaces  
  ethernet eth0 {  
    address 192.168.1.1/24  
    hw-id 0c:25:cc:60:00:00  
  }  
  ethernet eth1 {  
    hw-id 0c:25:cc:60:00:01  
  }  
  ethernet eth2 {  
    hw-id 0c:25:cc:60:00:02  
  }  
  loopback lo {  
  }  
yth[edit]  
vyos@vyos#
```

Рис. 2.17: Просмотр интерфейсов VyOS

### 2.3.2 Проверка связи и анализ трафика

- Проверка связи между устройствами выполнена с помощью команды **ping 192.168.1.1** на **PC1-ahmedfarg**.

Устройство успешно получает ответы от маршрутизатора, что подтверждает правильность конфигурации сети.

```
;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=1.505 ms  
; 84 bytes from 192.168.1.1 icmp_seq=2 ttl=64 time=1.008 ms  
; 84 bytes from 192.168.1.1 icmp_seq=3 ttl=64 time=1.074 ms  
; 84 bytes from 192.168.1.1 icmp_seq=4 ttl=64 time=1.315 ms  
; 84 bytes from 192.168.1.1 icmp_seq=5 ttl=64 time=0.871 ms  
;  
;VPCS>
```

Рис. 2.18: Проверка связи между ПК и маршрутизатором

- В **Wireshark** запущен захват пакетов на соединении между коммутатором и маршрутизатором.

Анализ показал корректный обмен ARP и ICMP Echo Request/Reply пакетами между устройствами.

Это демонстрирует, что сетевой стек функционирует правильно и маршрутизатор отвечает на запросы.

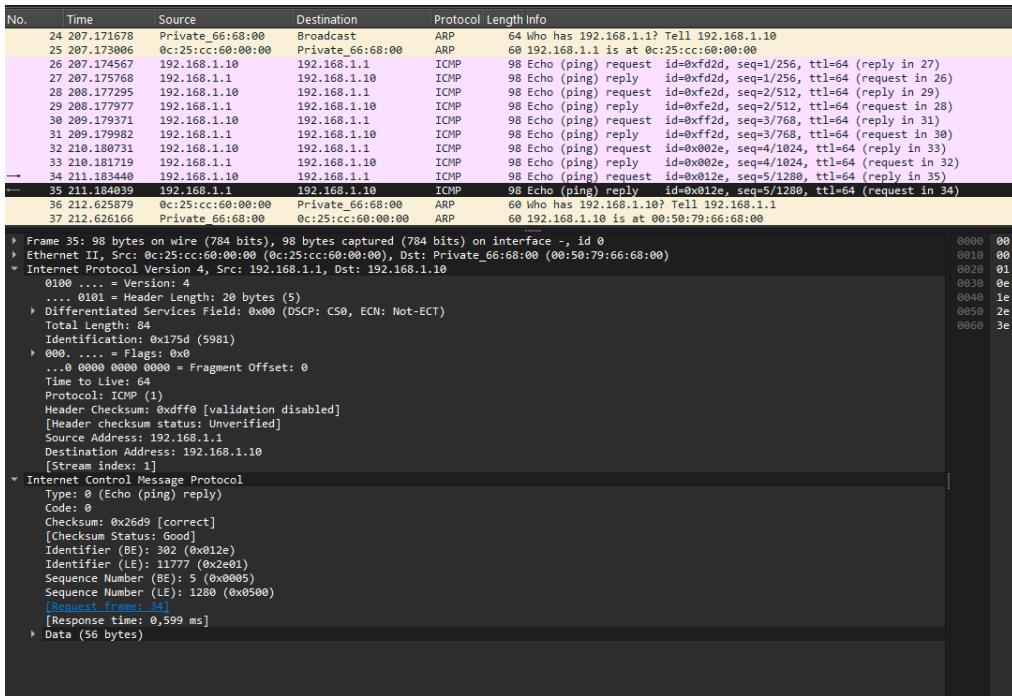


Рис. 2.19: Анализ ICMP и ARP пакетов в Wireshark

## 3 Заключение

В ходе работы была выполнена настройка и запуск виртуальной машины **GNS3**, а также добавлены и сконфигурированы образы маршрутизаторов **FRR** и **VyOS**. Построены и протестированы простейшие сетевые топологии с использованием коммутатора Ethernet и виртуальных ПК.

Настроена IP-адресация, выполнена проверка связности между узлами и проведён анализ сетевого трафика с помощью **Wireshark**.