

Отчёт по лабораторной работе 5

Простые сети в GNS3. Анализ трафика

Хамди Мохаммад

Содержание

1 Цель работы	5
2 Ход выполнения	6
2.1 Построение простой сети в GNS3	6
2.2 Настройка IP-адресов на узлах	7
2.3 Анализ трафика в Wireshark	8
2.4 Моделирование сети с маршрутизатором FRR	12
2.5 Моделирование сети с маршрутизатором VyOS	17
3 Заключение	20

Список иллюстраций

2.1 Топология сети в GNS3	6
2.2 Просмотр синтаксиса команд VPCS	7
2.3 Настройка IP-адреса на PC1	7
2.4 Проверка связности между PC1 и PC2	8
2.5 Захват ARP-пакетов в Wireshark	9
2.6 ICMP-запрос и ответ	10
2.7 Эхо-запросы в разных режимах	12
2.8 Топология сети с маршрутизатором FRR	13
2.9 Настройка IP-адреса на PC1	13
2.10 Настройка маршрутизатора FRR	14
2.11 Проверка конфигурации FRR	15
2.12 Проверка соединения между ПК и маршрутизатором	16
2.13 Анализ ICMP-трафика в Wireshark	16
2.14 Настройка маршрутизатора VyOS	17
2.15 Просмотр интерфейсов VyOS	18
2.16 Проверка связи между ПК и маршрутизатором VyOS	18
2.17 ICMP-анализ для VyOS	19

Список таблиц

1 Цель работы

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

2 Ход выполнения

2.1 Построение простой сети в GNS3

1. В рабочей области **GNS3** была создана топология сети, состоящая из двух виртуальных ПК и одного коммутатора Ethernet. Устройства соединены между собой прямыми линками.

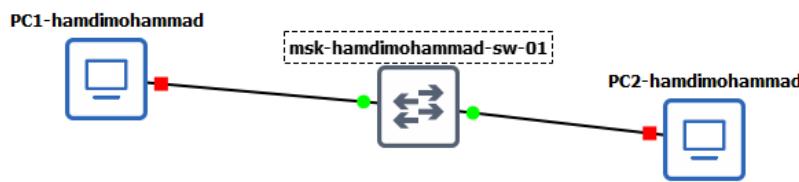


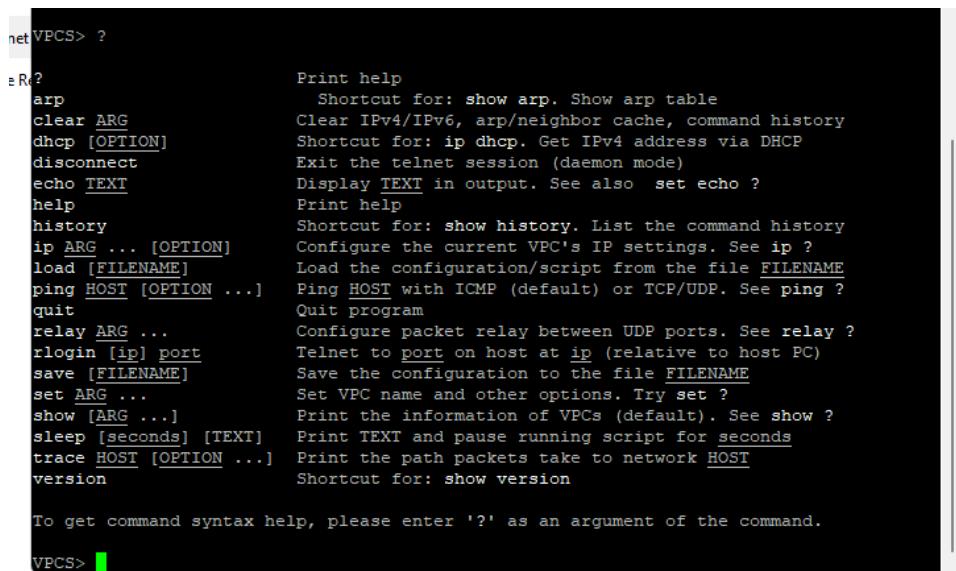
Рис. 2.1: Топология сети в GNS3

2. Каждому устройству было присвоено имя в соответствии с требованиями задания:
 - **PC1-hamdimohammad**
 - **PC2-hamdimohammad**
 - **msk-hamdimohammad-sw-01**

После соединения устройств проверено отображение интерфейсов и активное состояние линков.

2.2 Настройка IP-адресов на узлах

3. В терминале VPCS выполнена проверка доступных команд с помощью запроса ?, что позволило ознакомиться с синтаксисом конфигурации.



```
net VPCS> ?

?R?                                     Print help
arp                                         Shortcut for: show arp. Show arp table
clear ARG                                    Clear IPv4/IPv6, arp/neighbor cache, command history
dhcp [OPTION]                                Shortcut for: ip dhcp. Get IPv4 address via DHCP
disconnect                                   Exit the telnet session (daemon mode)
echo TEXT                                    Display TEXT in output. See also set echo ?
help                                         Print help
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 синтаксиса команд VPCS

4. На **PC1-hamdimohammad** был задан IP-адрес 192.168.1.11/24 с шлюзом 192.168.1.1, после чего конфигурация сохранена.



```
VPCS>
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. 2.3: Настройка IP-адреса на PC1

5. На **PC2-hamdimohammad** аналогично был настроен адрес 192.168.1.12/24 с тем же шлюзом. Проверка связности командой ping 192.168.1.11 показала успешный обмен ICMP-пакетами.

```
VPCS>
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>
VPCS> ping 192.168.1.11

84 bytes from 192.168.1.11 icmp_seq=1 ttl=64 time=2.892 ms
84 bytes from 192.168.1.11 icmp_seq=2 ttl=64 time=1.475 ms
84 bytes from 192.168.1.11 icmp_seq=3 ttl=64 time=0.747 ms
84 bytes from 192.168.1.11 icmp_seq=4 ttl=64 time=1.193 ms
84 bytes from 192.168.1.11 icmp_seq=5 ttl=64 time=0.937 ms

VPCS> [redacted]
```

Рис. 2.4: Проверка связности между PC1 и PC2

2.3 Анализ трафика в Wireshark

6. На соединении между **PC1** и **коммутатором** запущен захват пакетов в **Wireshark**. После старта узлов в окне анализатора появились **ARP-запросы** – объявление об IP-адресах устройств в сети.

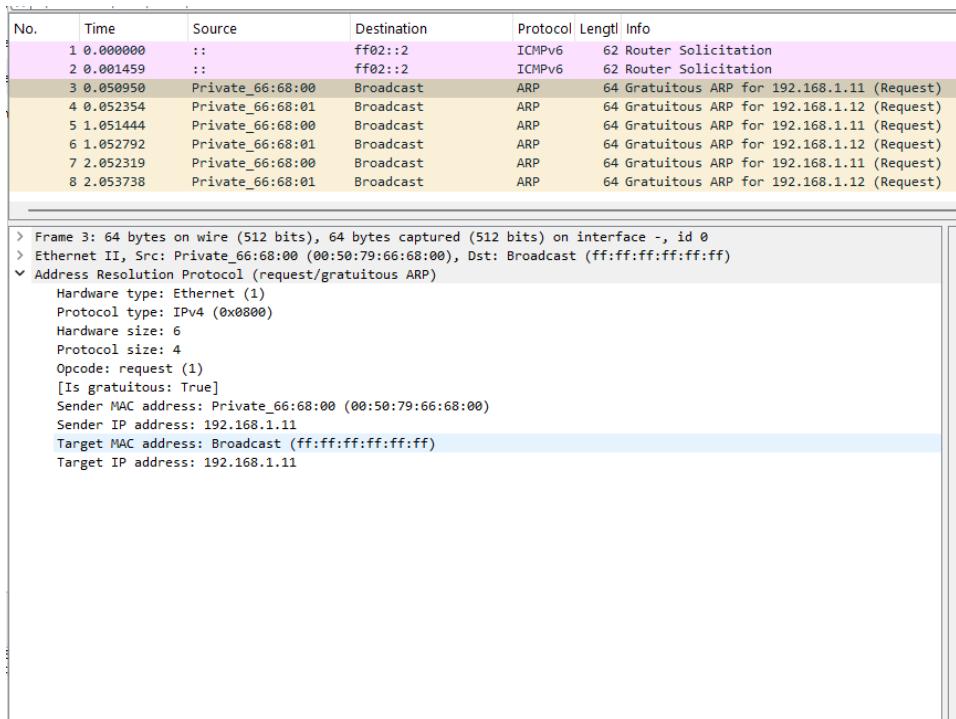


Рис. 2.5: Захват ARP-пакетов в Wireshark

В частности, зафиксированы **Gratuitous ARP**-пакеты от обоих ПК, а также ответы вида “*Who has 192.168.1.12? Tell 192.168.1.11*”, подтверждающие нормальную работу ARP-протокола.

7. Далее при выполнении команды ping был зафиксирован обмен **ICMP Echo Request** и **Echo Reply** между адресами 192.168.1.11 и 192.168.1.12.

No.	Time	Source	Destination	Protocol	Length	Info
5 1.051444		Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.11 (Request)
6 1.052792		Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.12 (Request)
7 2.052319		Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.11 (Request)
8 2.053738		Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.12 (Request)
9 136.488132		Private_66:68:00	Broadcast	ARP	64	Who has 192.168.1.12? Tell 192.168.1.11
10 136.488828		Private_66:68:01	Private_66:68:00	ARP	64	192.168.1.12 is at 00:50:79:66:68:01
→ 11 136.489563	192.168.1.11		192.168.1.12	ICMP	98	Echo (ping) request id=0xaaf, seq=1/256, ttl=64 (reply in 12)
← 12 136.490251	192.168.1.12		192.168.1.11	ICMP	98	Echo (ping) reply id=0xaaf, seq=1/256, ttl=64 (request in 11)

```
> Frame 11: 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: Private_66:68:01 (00:50:79:66:68:01)
> Internet Protocol Version 4, Src: 192.168.1.11, Dst: 192.168.1.12
└ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x758b [correct]
    [Checksum Status: Good]
    Identifier (BE): 43647 (0xaaa7f)
    Identifier (LE): 32682 (0x7faa)
    Sequence Number (BE): 1 (0x00001)
    Sequence Number (LE): 256 (0x0100)
    [Response frame: 12]
    > Data (56 bytes)

0000 00 50 79 66 68 01 00
0010 00 54 7f aa 00 00 40
0020 01 0c 08 00 75 8b aa
0030 0e 0f 10 11 12 13 14
0040 1e 1f 20 21 22 23 24
0050 2e 2f 30 31 32 33 34
0060 3e 3f
```

Рис. 2.6: ICMP-запрос и ответ

8. При выполнении эхо-запросов в различных режимах (ICMP, UDP, TCP) с помощью параметра -р зафиксированы пакеты соответствующих протоколов:

- **UDP-эхо-запросы** передавались по порту 7 с полезной нагрузкой в 56 байт.

No.	Time	Source	Destination	Protocol	Length	Info
7 2.052319		Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.11 (Request)
8 2.053738		Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.12 (Request)
9 136.488132		Private_66:68:00	Broadcast	ARP	64	Who has 192.168.1.12? Tell 192.168.1.11
10 136.488828		Private_66:68:01	Private_66:68:00	ARP	64	192.168.1.12 is at 00:50:79:66:68:01
11 136.489563	192.168.1.11		192.168.1.12	ICMP	98	Echo (ping) request id=0xaaf, seq=1/256, ttl=64 (reply in 12)
12 136.490251	192.168.1.12		192.168.1.11	ICMP	98	Echo (ping) reply id=0xaaf, seq=1/256, ttl=64 (request in 11)
13 167.008801	192.168.1.11		192.168.1.12	ECHO	98	Request
14 167.009130	192.168.1.12		192.168.1.11	ECHO	98	Response

```
> Frame 13: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface -, id 0
> Ethernet II, Src: Private_66:68:00 (00:50:79:66:68:00), Dst: Private_66:68:01 (00:50:79:66:68:01)
> Internet Protocol Version 4, Src: 192.168.1.11, Dst: 192.168.1.12
└ User Datagram Protocol, Src Port: 36323, Dst Port: 7
    Source Port: 36323
    Destination Port: 7
    Length: 64
    Checksum: 0x5292 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
    [Stream Packet Number: 1]
    > [Timestamps]
    UDP payload (56 bytes)
    < Echo
        Echo data: 0050796660000e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435363738393..
```

```
0000 00 50 79 66 68 01 00
0010 00 54 7f c9 00 00 40 11 77
0020 01 0c 08 e3 00 87 00 40 52
0030 0e 0f 10 11 12 13 14 15 16
0040 1e 1f 20 21 22 23 24 25 26
0050 2e 2f 30 31 32 33 34 35 36
0060 3e 3f
```

- **TCP-эхо-запросы** сопровождались установкой соединения по схеме **SYN-ACK-FIN**, что подтверждает корректное выполнение трёхстороннего рукопожатия TCP.

13	32.216550	192.168.1.11	192.168.1.12	ECHO	98 Request		
14	32.218051	192.168.1.12	192.168.1.11	ECHO	98 Response		
15	37.024191	192.168.1.11	192.168.1.12	TCP	74 12660 → 7 [SYN] Seq=0 Win=2920 Len=0 MSS=1460 TStamp=1761378568 TSectr=0 WS=2		
16	37.024841	192.168.1.12	192.168.1.11	TCP	54 7 → 12660 [SYN, ACK] Seq=0 Ack=1 Win=2920 Len=0		
17	37.026762	192.168.1.11	192.168.1.12	TCP	66 12660 → 7 [ACK] Seq=1 Ack=57 Win=2920 Len=0		
18	37.028674	192.168.1.11	192.168.1.12	ECHO	122 Request		
19	37.029600	192.168.1.12	192.168.1.11	TCP	54 7 → 12660 [ACK] Seq=1 Ack=57 Win=2920 Len=0		
20	37.032789	192.168.1.11	192.168.1.12	TCP	66 12660 → 7 [FIN, PSH, ACK] Seq=57 Ack=1 Win=0 TStamp=1761378568 TSectr=0		
21	37.033877	192.168.1.12	192.168.1.11	TCP	54 7 → 12660 [ACK] Seq=1 Ack=58 Win=2920 Len=0		
22	37.033955	192.168.1.12	192.168.1.11	TCP	54 7 → 12660 [FIN, ACK] Seq=1 Ack=58 Win=2920 Len=0		
23	37.037537	192.168.1.11	192.168.1.12	TCP	66 12660 → 7 [ACK] Seq=58 Ack=2 Win=0 TStamp=1761378568 TSectr=0		
> Frame 19: 54 bytes on wire (432 bits), 54 bytes captured (432 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							
Transmission Control Protocol, Src Port: 7, Dst Port: 12660, Seq: 1, Ack: 57, Len: 0							
Source Port: 7 Destination Port: 12660 [Stream index: 0] [Stream Packet Number: 5]							
> [Conversation completeness: Complete, WITH_DATA (31)] [TCP Segment Len: 0]							
Sequence Number: 1 (relative sequence number) Sequence Number (raw): 23403566							
Next Sequence Number: 1 (relative sequence number) Acknowledgment Number: 57 (relative ack number) Acknowledgment number (raw): 1330066428							
0101 ... = Header Length: 20 bytes (5) Flags: 0x010 (ACK) Window: 2920							
[Calculated window size: 2920] [Window size scaling factor: -2 (no window scaling used)] Checksum: 0x0d8c [unverified] [Checksum Status: Unverified] Urgent Pointer: 0							
> [Timestamps] > [SEQ/ACK analysis] [This is an ACK to the segment in frame: 18] [The RTT to ACK the segment was: 0.000926000 seconds] [iRTT: 0.002571000 seconds]							
0000 00 50 79 66 68 00 00 00 0010 00 28 81 00 00 40 06 0020 01 0b 00 07 31 74 01 65 0030 0b 68 0d 8c 00 00							

9. В терминале PC2 были выполнены по одному эхо-запросу в каждом из режимов (ICMP, UDP, TCP), что подтвердило доступность узла PC1 на всех уровнях транспортного протокола.

```

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 192.168.1.12 -c 1
84 bytes from 192.168.1.12 icmp_seq=1 ttl=64 time=2.718 ms
VPCS> ping 192.168.1.12 -c 1 -2
84 bytes from 192.168.1.12 udp_seq=1 ttl=64 time=2.983 ms
VPCS> ping 192.168.1.12 -c 1 -3
Connect 7@192.168.1.12 seq=1 ttl=64 time=2.478 ms
SendData 7@192.168.1.12 seq=1 ttl=64 time=3.002 ms
Close   7@192.168.1.12 seq=1 ttl=64 time=4.422 ms
VPCS>

```

Рис. 2.7: Эхо-запросы в разных режимах

2.4 Моделирование сети с маршрутизатором FRR

1. В рабочей области **GNS3** была создана топология, включающая один маршрутизатор **FRR**, один коммутатор Ethernet и одно оконечное устройство **VPCS**.

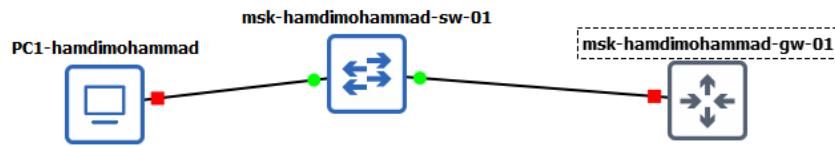


Рис. 2.8: Топология сети с маршрутизатором FRR

2. Окончному устройству **PC1-hamdimohammad** присвоен IP-адрес 192.168.1.10/24 и шлюз по умолчанию 192.168.1.1. Конфигурация сохранена, проверка выполнена с помощью команды `show ip`.

```
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. 2.9: Настройка IP-адреса на PC1

3. На маршрутизаторе **FRR** выполнена базовая настройка:

- изменено имя устройства на **msk-hamdimohammad-gw-01**;
- активирован интерфейс `eth0`;

- назначен IP-адрес 192.168.1.1/24.

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

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

Рис. 2.10: Настройка маршрутизатора FRR

4. Конфигурация маршрутизатора проверена командами `show running-config` и `show interface brief`. Видно, что интерфейс `eth0` активен и имеет корректный IP-адрес.

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

Рис. 2.11: Проверка конфигурации FRR

5. Проверка связности между ПК и маршрутизатором командой ping 192.168.1.1 показала успешный обмен ICMP-пакетами.

```
VPCS> ping 192.168.1.1

84 bytes from 192.168.1.1 icmp_seq=1 ttl=64 time=2.253 ms
84 bytes from 192.168.1.1 icmp_seq=2 ttl=64 time=1.706 ms
84 bytes from 192.168.1.1 icmp_seq=3 ttl=64 time=1.843 ms
84 bytes from 192.168.1.1 icmp_seq=4 ttl=64 time=2.299 ms
84 bytes from 192.168.1.1 icmp_seq=5 ttl=64 time=3.206 ms

VPCS>
```

Рис. 2.12: Проверка соединения между ПК и маршрутизатором

6. В **Wireshark** зафиксированы ICMP-запросы и ответы между устройствами.

Анализ показал корректную работу протоколов **ARP** и **ICMP**, обеспечивающих адресное разрешение и обмен эхо-запросами.

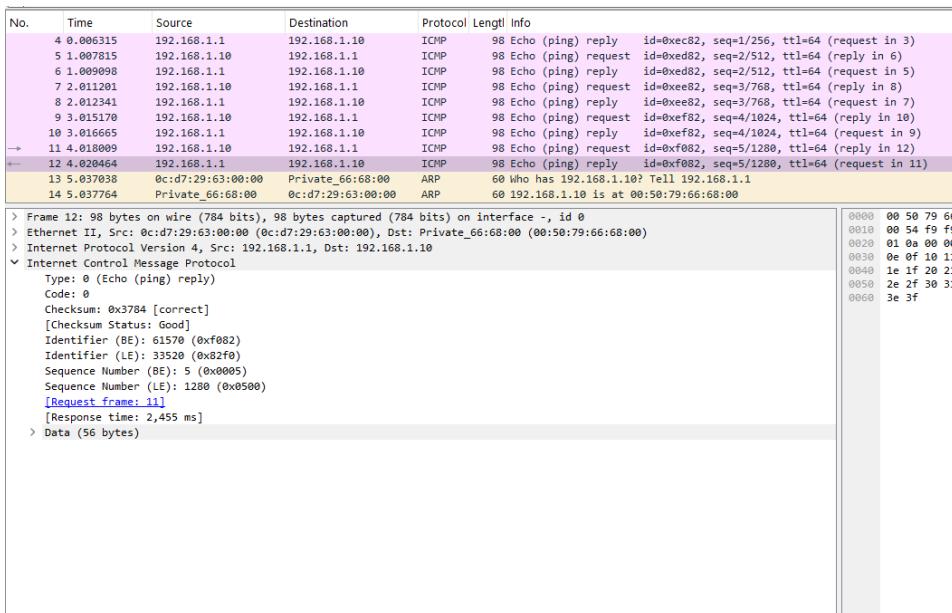


Рис. 2.13: Анализ ICMP-трафика в Wireshark

2.5 Моделирование сети с маршрутизатором VyOS

7. В новой топологии **GNS3** размещены маршрутизатор **VyOS**, коммутатор **Ethernet** и устройство **PC1-hamdimohammad**.
8. ПК получил IP-адрес 192.168.1.10/24 и шлюз 192.168.1.1. Настройка сохранена.
9. После входа на маршрутизатор VyOS под учетной записью **vyos/vyos** и установки системы выполнена базовая настройка:
 - установлено имя устройства **msk-hamdimohammad-gw-01**;
 - интерфейсу **eth0** присвоен адрес 192.168.1.1/24.

```
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# set system host-name msk-hamdimohammad-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-hamdimohammad-gw-01
[edit]
vyos@vyos# commit
```

Рис. 2.14: Настройка маршрутизатора VyOS

10. Проверка интерфейсов показала, что **eth0** активен и имеет правильный адрес.

```
[edit]
vyos@vyos# show interfaces
ethernet eth0 {
    address 192.168.1.1/24
    hw-id 0c:96:7e:5a:00:00
}
ethernet eth1 {
    hw-id 0c:96:7e:5a:00:01
}
ethernet eth2 {
    hw-id 0c:96:7e:5a:00:02
}
loopback lo {
}
[edit]
vyos@vyos#
```

Рис. 2.15: Просмотр интерфейсов VyOS

11. Проверка связи с ПК командой `ping 192.168.1.1` прошла успешно – получены ответы от маршрутизатора с временем отклика менее 5 мс.

```
VPCS>
VPCS> ping 192.168.1.1
.
84 bytes from 192.168.1.1 icmp_seq=1 ttl=64 time=2.775 ms
84 bytes from 192.168.1.1 icmp_seq=2 ttl=64 time=2.233 ms
84 bytes from 192.168.1.1 icmp_seq=3 ttl=64 time=4.229 ms
84 bytes from 192.168.1.1 icmp_seq=4 ttl=64 time=2.681 ms
84 bytes from 192.168.1.1 icmp_seq=5 ttl=64 time=2.798 ms
VPCS>
```

Рис. 2.16: Проверка связи между ПК и маршрутизатором VyOS

12. В **Wireshark** зафиксированы пакеты **ICMP Echo Request** и **Echo Reply** между адресами 192.168.1.10 и 192.168.1.1. Протокол отработал корректно, обмен пакетами происходил без потерь.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.10	192.168.1.1	ICMP	98	Echo (ping) request id=0xa584, seq=1/256, ttl=64 (reply in 2)
2	0.002004	192.168.1.1	192.168.1.10	ICMP	98	Echo (ping) reply id=0xa584, seq=1/256, ttl=64 (request in 1)
3	1.003784	192.168.1.10	192.168.1.1	ICMP	98	Echo (ping) request id=0xa684, seq=2/512, ttl=64 (reply in 4)
4	1.005253	192.168.1.1	192.168.1.10	ICMP	98	Echo (ping) reply id=0xa684, seq=2/512, ttl=64 (request in 3)
5	2.008156	192.168.1.10	192.168.1.1	ICMP	98	Echo (ping) request id=0xa784, seq=3/768, ttl=64 (reply in 6)
6	2.011016	192.168.1.1	192.168.1.10	ICMP	98	Echo (ping) reply id=0xa784, seq=3/768, ttl=64 (request in 5)
7	3.012660	192.168.1.10	192.168.1.1	ICMP	98	Echo (ping) request id=0xa884, seq=4/1024, ttl=64 (reply in 8)
8	3.014549	192.168.1.1	192.168.1.10	ICMP	98	Echo (ping) reply id=0xa884, seq=4/1024, ttl=64 (request in 7)
→	9.4.017875	192.168.1.10	192.168.1.1	ICMP	98	Echo (ping) request id=0xa984, seq=5/1280, ttl=64 (reply in 10)
→	10.4.019807	192.168.1.1	192.168.1.10	ICMP	98	Echo (ping) reply id=0xa984, seq=5/1280, ttl=64 (request in 9)

> Frame 10: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface -, id 0
 > Ethernet II, Src: 0c:96:7e:5a:00:00 (0c:96:7e:5a:00:00), Dst: Private_66:68:00 (00:50:79:66:68:00)
 ✓ Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.10
 0100 = Version: 4
 0101 = Header Length: 20 bytes (5)
 > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 Total Length: 84
 Identification: 0xeb51 (60241)
 > 000. = Flags: 0x0
 ...0 0000 0000 0000 = Fragment Offset: 0
 Time to Live: 64
 Protocol: ICMP (1)
 Header Checksum: 0x0bfc [validation disabled]
 [Header checksum status: Unverified]
 Source Address: 192.168.1.1
 Destination Address: 192.168.1.10
 [Stream index: 0]
 ✓ Internet Control Message Protocol
 Type: 0 (Echo (ping) reply)
 Code: 0
 Checksum: 0x7e82 [correct]
 [Checksum Status: Good]
 Identifier (BE): 43396 (0xa984)
 Identifier (LE): 33961 (0x84a9)
 Sequence Number (BE): 5 (0x0005)
 Sequence Number (LE): 1280 (0x0500)
 [Request frame: 9]
 [Response time: 1,932 ms]
 > Data (56 bytes)

0000	00 50 79 :
0010	00 54 eb :
0020	01 0a 00 :
0030	0e 0f 10 :
0040	1e 1f 20 :
0050	2e 2f 30 :
0060	3e 3f

Рис. 2.17: ICMP-анализ для VyOS

3 Заключение

В ходе лабораторной работы были смоделированы две простейшие сети на базе маршрутизаторов **FRR** и **VyOS**. Для обеих конфигураций выполнена настройка IP-адресации, проверена связность между устройствами и проанализирован сетевой трафик.

Захват в **Wireshark** подтвердил корректную работу протоколов **ARP** и **ICMP**, обеспечивающих базовый обмен сообщениями между маршрутизатором и оконечным устройством.