

Отчёт по лабораторной работе №5

Простые сети в GNS3. Анализ трафика

Авдадаев Джамал Геланиевич

Содержание

1 Цель работы	5
2 Выполнение задания	6
2.1 Построение простейшей сети и проверка связности	6
2.1.1 Просмотр синтаксиса команд VPCS	6
2.1.2 Назначение IP-адресов оконечным устройствам	7
2.1.3 Проверка связности между узлами	9
2.2 Анализ трафика в GNS3 с использованием Wireshark	10
2.2.1 Анализ ARP-сообщений	10
2.2.2 Анализ ICMP-сообщений	11
2.2.3 Анализ UDP-сообщений	11
2.2.4 Анализ TCP-сообщений	12
2.3 Моделирование простейшей сети на базе маршрутизатора FRR в GNS3	13
2.3.1 Построение топологии сети	13
2.3.2 Настройка IP-адресации на оконечном устройстве	14
2.3.3 Настройка маршрутизатора FRR	15
2.3.4 Проверка конфигурации маршрутизатора	16
2.3.5 Проверка сетевой связности	17
2.3.6 Анализ трафика в Wireshark	18
2.4 Моделирование простейшей сети на базе маршрутизатора VyOS в GNS3	19
2.4.1 Построение топологии сети	20
2.4.2 Настройка IP-адресации на оконечном устройстве	20
2.4.3 Настройка маршрутизатора VyOS	21
2.4.4 Проверка сетевой связности	22
2.4.5 Анализ трафика в Wireshark	23
3 Заключение	25

Список иллюстраций

2.1 Топология сети в GNS3	6
2.2 Справка команд VPCS	7
2.3 Назначение IP-адреса PC1	8
2.4 Назначение IP-адреса PC2	9
2.5 ARP-трафик в Wireshark	10
2.6 ICMP Echo request и reply	11
2.7 UDP-пакет при ping в UDP-режиме	12
2.8 TCP-сессия при ping в TCP-режиме	13
2.9 Топология сети с маршрутизатором FRR	14
2.10 Настройка IP-адресации на PC1	15
2.11 Настройка интерфейса eth0 на маршрутизаторе FRR	16
2.12 Проверка конфигурации и интерфейсов FRR	17
2.13 Ping с PC1 на маршрутизатор	18
2.14 Анализ ARP и ICMP-трафика в Wireshark	19
2.15 Топология сети с маршрутизатором VyOS	20
2.16 Настройка и проверка интерфейсов маршрутизатора VyOS	22
2.17 Ping с PC1 на маршрутизатор VyOS	23
2.18 Ping с PC1 на маршрутизатор VyOS	24

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 задания

2.1 Построение простейшей сети и проверка связности

В ходе выполнения лабораторной работы в среде моделирования **GNS3** была построена простейшая локальная сеть, состоящая из одного Ethernet-коммутатора и двух оконечных устройств типа **VPCS**. Узлы были переименованы с включением имени учётной записи студента: **PC1-dgavdadaev**, **PC2-dgavdadaev**, коммутатор – **msk-dgavdadaev-sw-01**. После переименования оконечные устройства были соединены с коммутатором, в результате чего был сформирован единый широковещательный домен.

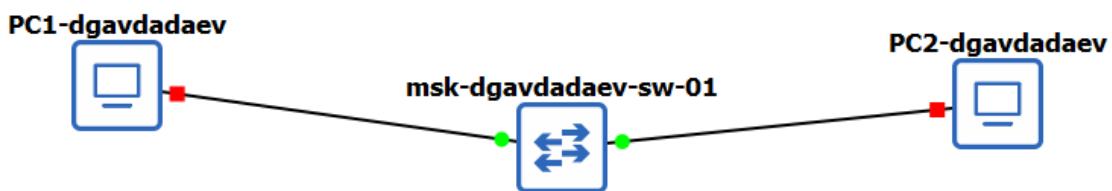


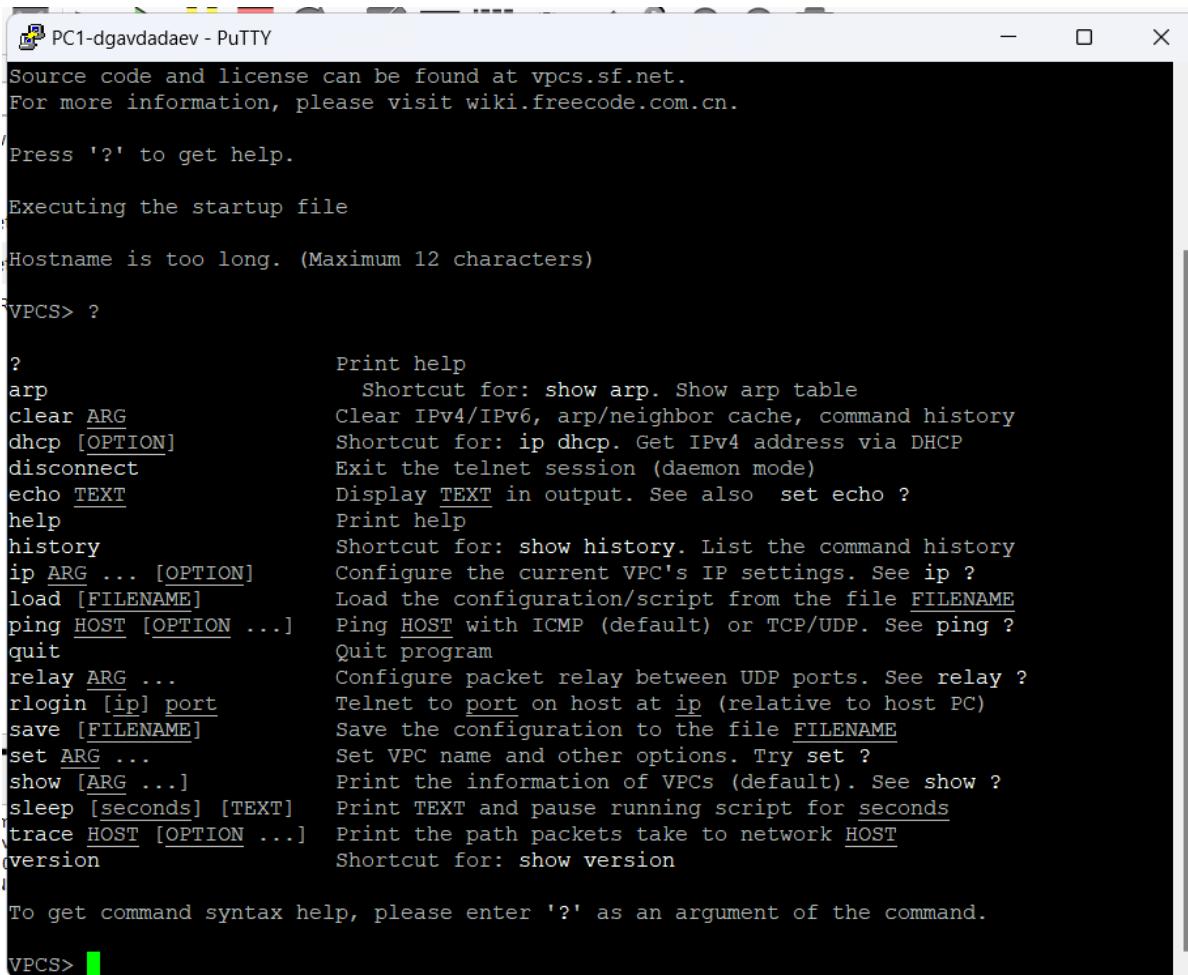
Рис. 2.1: Топология сети в GNS3

2.1.1 Просмотр синтаксиса команд VPCS

После запуска узлов был открыт терминал **PC1**. Для ознакомления с доступными командами VPCS была вызвана встроенная справка с помощью команды `?.` В

терминале отобразился перечень основных команд, предназначенных для настройки IP-параметров, диагностики соединений и управления конфигурацией.

При старте VPCS также выводится сообщение “*Hostname is too long (Maximum 12 characters)*”, связанное с ограничением длины имени хоста в VPCS. Данное сообщение носит информационный характер и не влияет на функционирование сети.



```
PC1-dgavdadaev - PuTTY
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> ?

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

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

VPCS> 
```

Рис. 2.2: Справка команд VPCS

2.1.2 Назначение IP-адресов оконечным устройствам

Для настройки сетевых параметров на **PC1** предварительно был просмотрен синтаксис команды настройки адресации с помощью `ip ?`. После этого устрой-

ству был назначен IP-адрес **192.168.1.11** с маской **/24**, а также указан шлюз по умолчанию **192.168.1.1**. После ввода параметров конфигурация была сохранена командой **save**.

Процесс задания IP-адреса и сохранения конфигурации на PC1 показан на рисунке ниже.

```
PC1-dgavdadaev - PuTTY

VPCS>
VPCS> ip ?

ip ARG ... [OPTION]
Configure the current VPC's IP settings
  ARG ....:
    address [mask] [gateway]
    address [gateway] [mask]
      Set the VPC's ip, default gateway ip and network mask
      Default IPv4 mask is /24, IPv6 is /64. Example:
      ip 10.1.1.70/26 10.1.1.65 set the VPC's ip to 10.1.1.70,
      the gateway to 10.1.1.65, the netmask to 255.255.255.192.
      In tap mode, the ip of the tapx is the maximum host ID
      of the subnet. In the example above the tapx ip would be
      10.1.1.126
      mask may be written as /26, 26 or 255.255.255.192
  auto      Attempt to obtain IPv6 address, mask and gateway using SLAAC
  dhcp [OPTION]  Attempt to obtain IPv4 address, mask, gateway, DNS via DHCP
    -d        Show DHCP packet decode
    -r        Renew DHCP lease
    -x        Release DHCP lease
  dns ip      Set DNS server ip, delete if ip is '0'
  dns6 ipv6   Set DNS server ipv6, delete if ipv6 is '0'
  domain NAME  Set local domain name to NAME

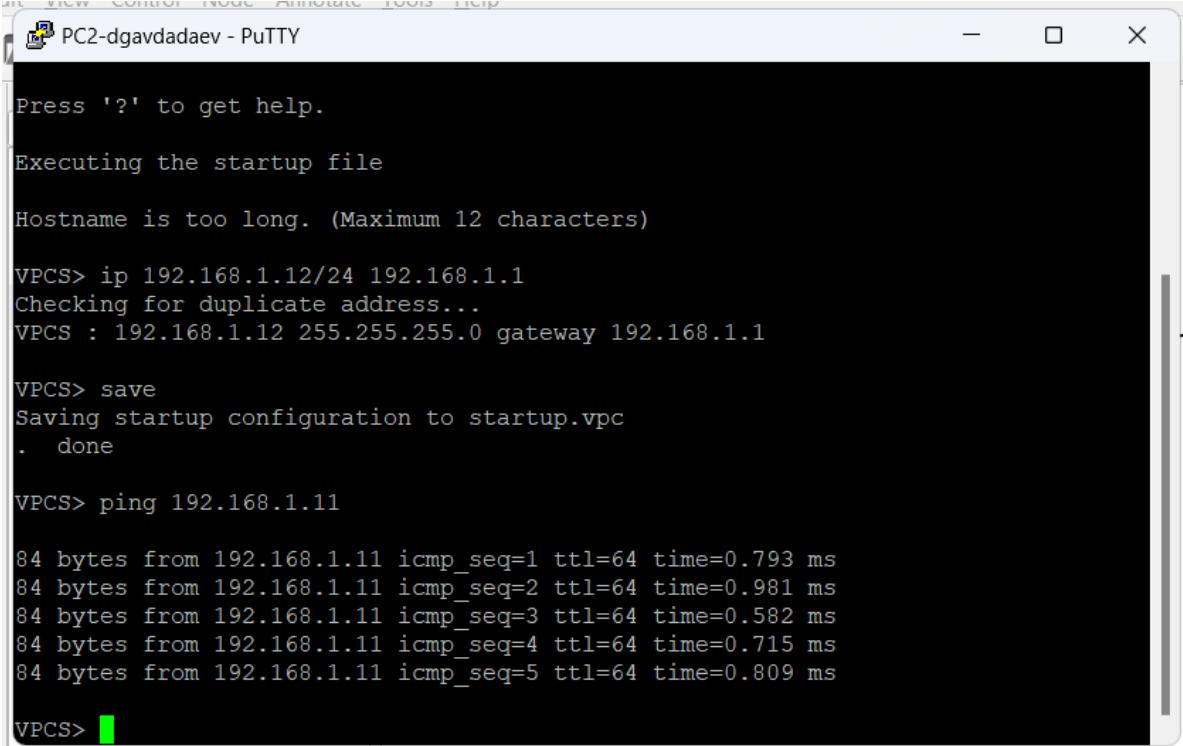
VPCS> ip 192.168.1.11/24 192.168.1.1
Checking for duplicate address...
VPCS : 192.168.1.11 255.255.255.0 gateway 192.168.1.1

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

VPCS>
```

Рис. 2.3: Назначение IP-адреса PC1

Аналогичным образом на **PC2** был настроен IP-адрес **192.168.1.12/24** со шлюзом **192.168.1.1**, после чего конфигурация также была сохранена.



PC2-dgavdadaev - 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=0.793 ms
84 bytes from 192.168.1.11 icmp_seq=2 ttl=64 time=0.981 ms
84 bytes from 192.168.1.11 icmp_seq=3 ttl=64 time=0.582 ms
84 bytes from 192.168.1.11 icmp_seq=4 ttl=64 time=0.715 ms
84 bytes from 192.168.1.11 icmp_seq=5 ttl=64 time=0.809 ms

VPCS>
```

Рис. 2.4: Назначение IP-адреса PC2

2.1.3 Проверка связности между узлами

Для проверки работоспособности созданной сети и корректности настроенной адресации с **PC2** был выполнен эхо-запрос к **PC1**. В ответ были получены ICMP Echo Reply-пакеты без потерь, что свидетельствует о корректной работе коммутатора, правильной IP-адресации и наличии сетевой связности между узлами в одной подсети.

Результат выполнения проверки связности представлен на рисунке выше (см. Screenshot_4.png).

2.2 Анализ трафика в GNS3 с использованием Wireshark

На следующем этапе работы был выполнен захват и анализ сетевого трафика с помощью **Wireshark**. Захват был запущен на соединении между **PC1** и коммутатором. После запуска всех узлов в проекте в Wireshark начали отображаться кадры канального и сетевого уровней.

2.2.1 Анализ ARP-сообщений

В процессе захвата были зафиксированы ARP-кадры, в том числе **gratuitous ARP**. Данные сообщения отправляются узлом широковещательно и используются для проверки уникальности IP-адреса, а также для оповещения других устройств сети о соответствии IP- и MAC-адресов.

В разборе ARP-кадра видно, что IP-адрес отправителя совпадает с целевым IP-адресом, а MAC-адрес назначения является широковещательным. Это подтверждает, что кадр является gratuitous ARP-запросом.

3 0.050386	Private_66:68:00	Broadcast	ARP	64 Gratuitous ARP for 192.168.1.11 (Request)
4 0.050426	Private_66:68:01	Broadcast	ARP	64 Gratuitous ARP for 192.168.1.12 (Request)
5 1.050376	Private_66:68:00	Broadcast	ARP	64 Gratuitous ARP for 192.168.1.11 (Request)
6 1.051310	Private_66:68:01	Broadcast	ARP	64 Gratuitous ARP for 192.168.1.12 (Request)

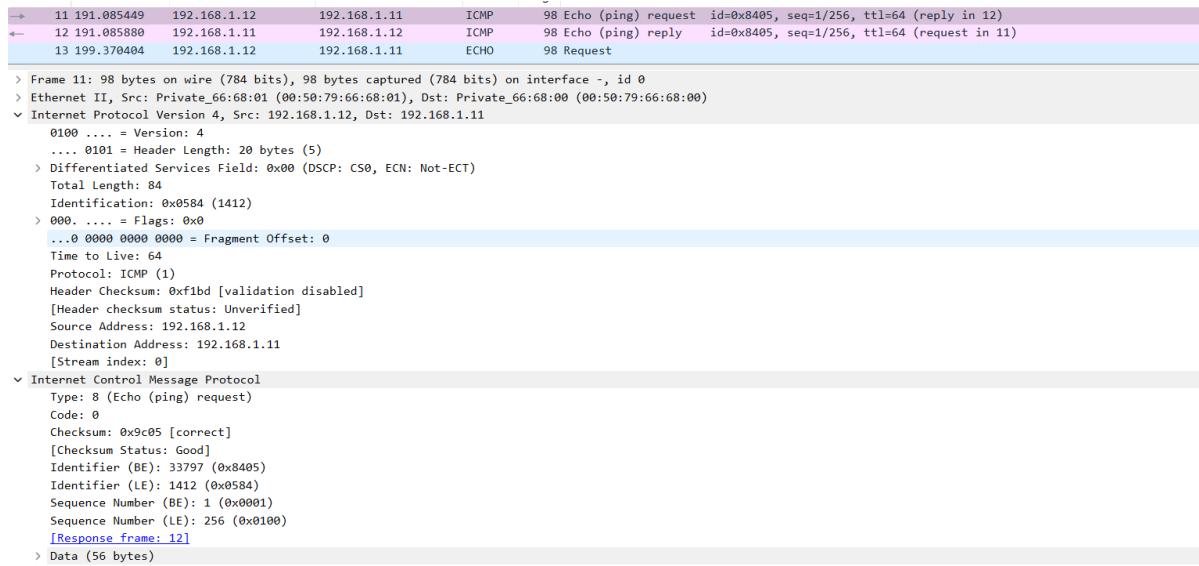

```
> Frame 3: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface -, id 0
> Ethernet II, Src: Private_66:68:00 (00:50:79:66:68:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
✓ Address Resolution Protocol (request/gratuitous ARP)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    [Is gratuitous: True]
    Sender MAC address: Private_66:68:00 (00:50:79:66:68:00)
    Sender IP address: 192.168.1.11
    Target MAC address: Broadcast (ff:ff:ff:ff:ff:ff)
    Target IP address: 192.168.1.11
```

Рис. 2.5: ARP-трафик в Wireshark

2.2.2 Анализ ICMP-сообщений

В терминале **PC2** была вызвана справка по параметрам команды `ping`, после чего выполнен одиночный эхо-запрос в **ICMP-режиме** к узлу **PC1**. В Wireshark был зафиксирован ICMP Echo Request от адреса **192.168.1.12** к **192.168.1.11**, а также соответствующий ICMP Echo Reply в обратном направлении.

В деталях пакета отображаются поля протоколов Ethernet, IPv4 и ICMP, включая тип сообщения, идентификатор, номер последовательности и значение TTL.



The screenshot shows a Wireshark capture window with three frames. Frame 11 is an ICMP Echo Request from 192.168.1.12 to 192.168.1.11. Frame 12 is an ICMP Echo Reply from 192.168.1.11 to 192.168.1.12. Frame 13 is an ECHO Request from 192.168.1.12 to 192.168.1.11. The details pane shows the ICMP header fields for each frame, including Type, Code, Identifier, Sequence Number, and TTL. The ICMP Type field is 8 (Echo (ping) request) for the first two frames and 98 (Request) for the third.

Рис. 2.6: ICMP Echo request и reply

2.2.3 Анализ UDP-сообщений

После этого был выполнен одиночный эхо-запрос в **UDP-режиме**. В Wireshark был зафиксирован UDP-пакет, инкапсулированный в IPv4, с целевым портом **7** (служба Echo). В пакете присутствует полезная нагрузка, возвращаемая получателем.

Анализ показывает, что при использовании UDP отсутствует установление соединения, а обмен данными осуществляется без подтверждения доставки на транспортном уровне.

11	191.085449	192.168.1.12	192.168.1.11	ICMP	98 Echo (ping) request	id=0x8405, seq=1/256, ttl=64 (reply in 12)
12	191.085880	192.168.1.11	192.168.1.12	ICMP	98 Echo (ping) reply	id=0x8405, seq=1/256, ttl=64 (request in 11)
13	199.370484	192.168.1.12	192.168.1.11	ECHO	98 Request	
14	199.371217	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: 0x058c (1420)
    > 000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: UDP (17)
    Header Checksum: 0xf1a5 [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: 23952, Dst Port: 7
        Source Port: 23952
        Destination Port: 7
        Length: 64
        Checksum: 0x82e4 [unverified]
        [Checksum Status: Unverified]
        [Stream index: 0]
        [Stream Packet Number: 1]
        > [Timestamps]
        UDP payload (56 bytes)
    < Echo
        Echo data: 0050796668010e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f

```

Рис. 2.7: UDP-пакет при ping в UDP-режиме

2.2.4 Анализ TCP-сообщений

При выполнении эхо-запроса в **TCP-режиме** в Wireshark наблюдается полноценная TCP-сессия. Сначала происходит установка соединения с использованием трёхстороннего рукопожатия (SYN, SYN/ACK, ACK), затем передача данных, после чего соединение корректно завершается с помощью сегментов FIN и ACK.

Данный анализ наглядно демонстрирует отличие TCP от ICMP и UDP, а также наличие механизмов установления соединения, подтверждения доставки и корректного завершения сеанса.

Рис. 2.8: TCP-сессия при ping в TCP-режиме

2.3 Моделирование простейшей сети на базе маршрутизатора FRR в GNS3

В рамках выполнения лабораторной работы в среде **GNS3** была смоделирована простейшая сеть с использованием маршрутизатора **FRR**, Ethernet-коммутатора и одного оконечного устройства **VPCS**. Целью работы являлась настройка базовой IP-адресации и анализ обмена ICMP- и ARP-сообщениями между узлом и маршрутизатором.

2.3.1 Построение топологии сети

В рабочей области GNS3 были размещены следующие устройства: - оконечное устройство **PC1-dgavdadaev**; - Ethernet-коммутатор **msk-dgavdadaev-sw-01**; - маршрутизатор **msk-dgavdadaev-gw-01**.

Устройства были соединены таким образом, что PC1 подключён к коммутатору, а коммутатор – к маршрутизатору. Таким образом, была сформирована локальная сеть с маршрутизатором в роли шлюза по умолчанию.

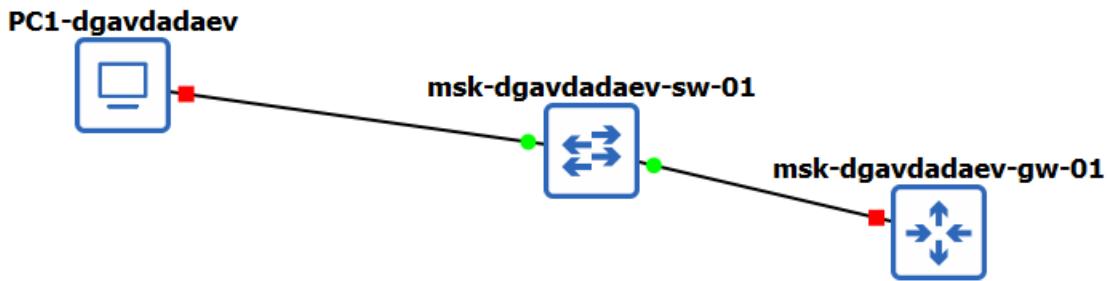


Рис. 2.9: Топология сети с маршрутизатором FRR

2.3.2 Настройка IP-адресации на оконечном устройстве

После запуска всех устройств был открыт терминал **PC1**. Оконечному устройству было назначено IP-адрес **192.168.1.10/24** с указанием шлюза по умолчанию **192.168.1.1**. После ввода параметров конфигурация была сохранена, а затем выполнен просмотр текущих IP-настроек для контроля корректности ввода.

В результате вывода команды отображаются: - IP-адрес узла – 192.168.1.10/24; - адрес шлюза – 192.168.1.1; - MAC-адрес сетевого интерфейса.

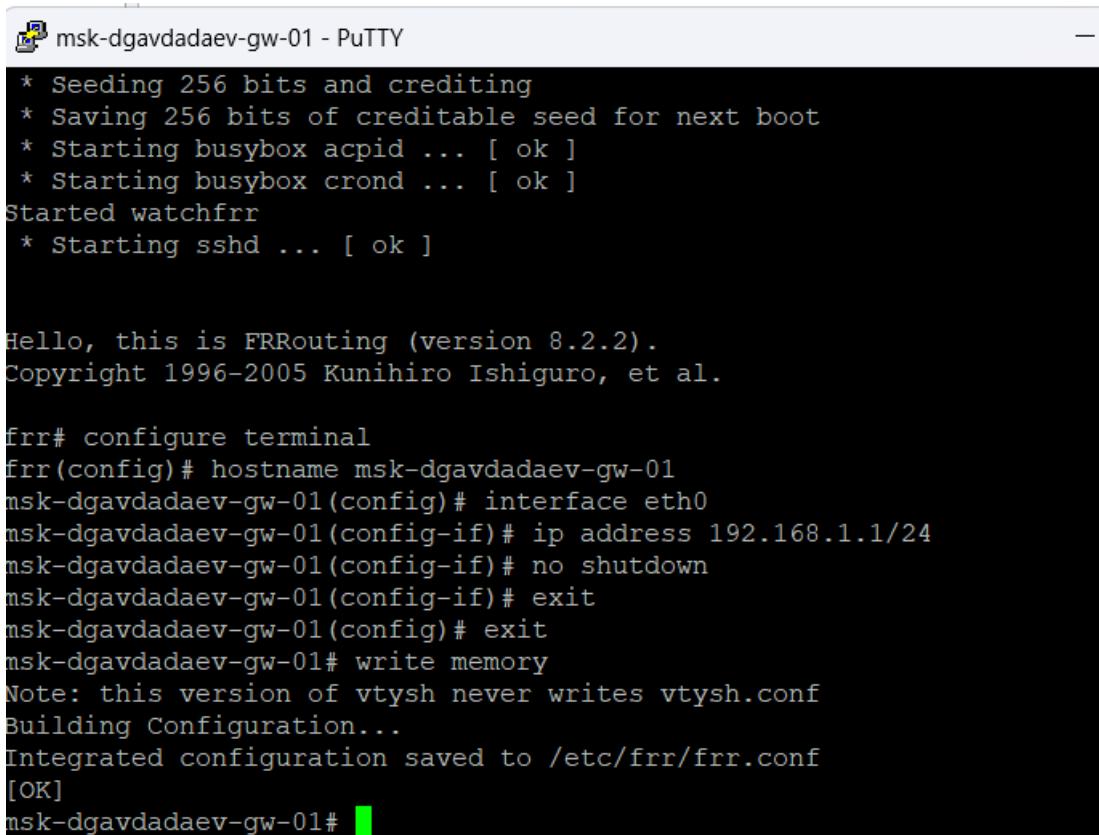
```
VPCS>
VPCS> ip 192.168.1.10/24 192.168.1.1
udChecking for duplicate address...
errVPCS : 192.168.1.10 255.255.255.0 gateway 192.168.1.1
errVPCS> save
Saving startup configuration to startup.vpc
me. done
?
VPCS> show ip
T
S NAME      : VPCS[1]
S IP/MASK   : 192.168.1.10/24
S GATEWAY   : 192.168.1.1
S DNS       :
S MAC       : 00:50:79:66:68:00
S LPORT     : 10004
S RHOST:PORT: 127.0.0.1:10005
S MTU       : 1500
VPCS>
```

Рис. 2.10: Настройка IP-адресации на PC1

2.3.3 Настройка маршрутизатора FRR

Далее была выполнена настройка маршрутизатора **FRR**. Через консоль маршрутизатора был задан hostname **msk-dgavdadaev-gw-01**, после чего произведена настройка IP-адресации интерфейса локальной сети **eth0**. Интерфейсу был присвоен адрес **192.168.1.1/24**, после чего интерфейс был переведён в активное состояние.

После завершения конфигурации параметры были сохранены в постоянную память.



```
* Seeding 256 bits and crediting
* Saving 256 bits of creditable seed for next boot
* Starting busybox acpid ... [ ok ]
* Starting busybox crond ... [ ok ]
Started watchfrr
* Starting sshd ... [ ok ]

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

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

Рис. 2.11: Настройка интерфейса eth0 на маршрутизаторе FRR

2.3.4 Проверка конфигурации маршрутизатора

Для контроля корректности выполненной настройки были просмотрены текущая конфигурация маршрутизатора и краткая информация по интерфейсам. В выводе подтверждается: - наличие настроенного интерфейса **eth0**; - его состояние **up**; - назначенный IP-адрес **192.168.1.1/24**.

```
[OK]
msk-dgavdadaev-gw-01# show running-config
Building configuration...

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

Рис. 2.12: Проверка конфигурации и интерфейсов FRR

2.3.5 Проверка сетевой связности

Для проверки работоспособности сети с оконечного устройства **PC1** был выполнен эхо-запрос к IP-адресу маршрутизатора **192.168.1.1**. В результате были получены ответы на все ICMP-запросы, что подтверждает: - корректную IP-адресацию узла и маршрутизатора; - исправную работу Ethernet-коммутатора; - наличие сетевой связности между PC1 и маршрутизатором.

```
All devices PC1-dgavdadaev - PuTTY VPCS> ping 192.168.1.1 84 bytes from 192.168.1.1 icmp_seq=1 ttl=64 time=3.627 ms 84 bytes from 192.168.1.1 icmp_seq=2 ttl=64 time=4.582 ms 84 bytes from 192.168.1.1 icmp_seq=3 ttl=64 time=2.124 ms 84 bytes from 192.168.1.1 icmp_seq=4 ttl=64 time=1.078 ms 84 bytes from 192.168.1.1 icmp_seq=5 ttl=64 time=2.538 ms VPCS>
```

Рис. 2.13: Ping с PC1 на маршрутизатор

2.3.6 Анализ трафика в Wireshark

Во время проверки связности был включён захват трафика на соединении между коммутатором и маршрутизатором. В окне **Wireshark** зафиксированы ARP- и ICMP-сообщения.

В начале обмена наблюдается ARP-запрос от узла **192.168.1.10** с целью определения MAC-адреса узла **192.168.1.1**. В ответ маршрутизатор отправляет ARP-ответ, содержащий свой MAC-адрес. После разрешения ARP выполняется обмен ICMP Echo Request и Echo Reply между PC1 и маршрутизатором.

В деталях ICMP-пакета отображаются стандартные поля протоколов Ethernet, IPv4 и ICMP, включая тип сообщения, идентификатор, номер последовательности и время отклика.

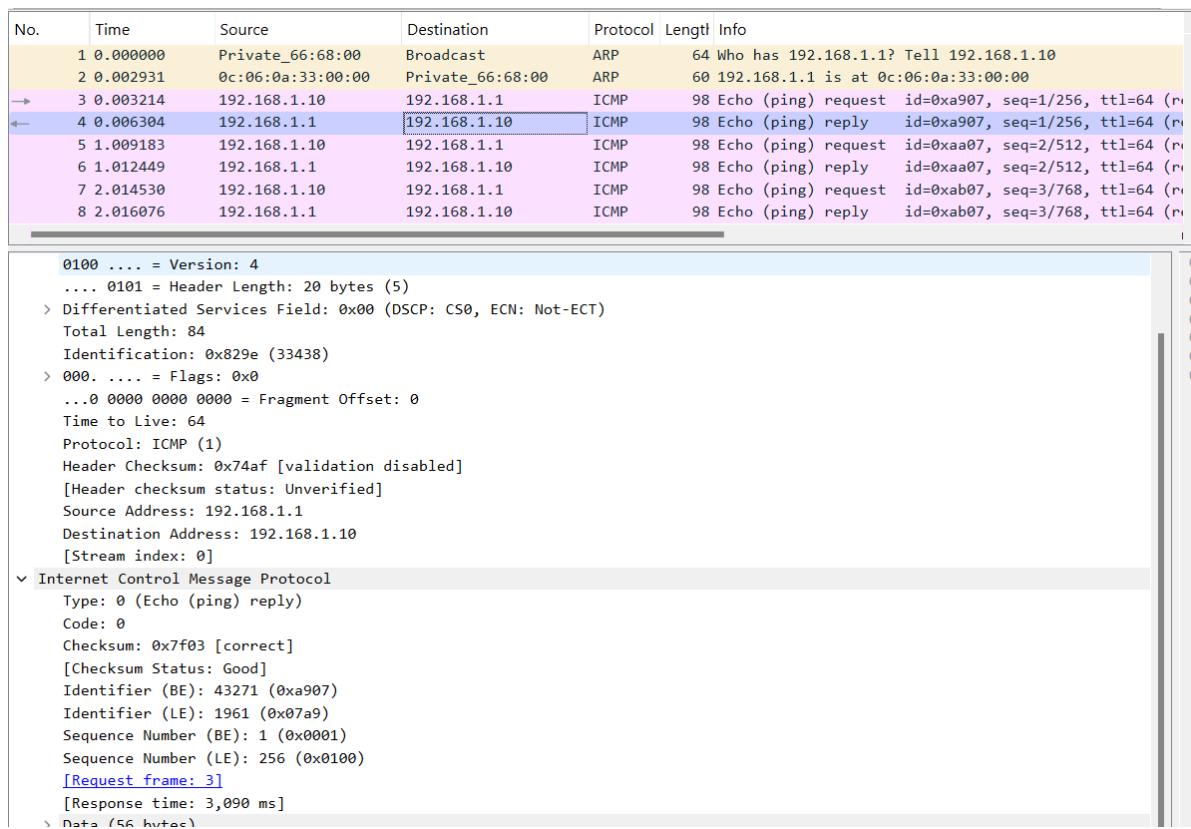


Рис. 2.14: Анализ ARP и ICMP-трафика в Wireshark

2.4 Моделирование простейшей сети на базе маршрутизатора VyOS в GNS3

В рамках выполнения лабораторной работы в среде **GNS3** была смоделирована простейшая сеть с использованием маршрутизатора **VyOS**, Ethernet-коммутатора и одного оконечного устройства **VPCS**. Целью работы являлась настройка базовой IP-адресации, проверка сетевой связности и анализ трафика с использованием анализатора **Wireshark**.

2.4.1 Построение топологии сети

В рабочей области GNS3 были размещены следующие устройства: - оконечное устройство **PC1-dgavdadaev**; - Ethernet-коммутатор **msk-dgavdadaev-sw-01**; - маршрутизатор **msk-dgavdadaev-gw-01** на базе VyOS.

Оконечное устройство было подключено к коммутатору, а коммутатор — к маршрутизатору. Таким образом, была сформирована локальная сеть с маршрутизатором VyOS в роли шлюза по умолчанию.

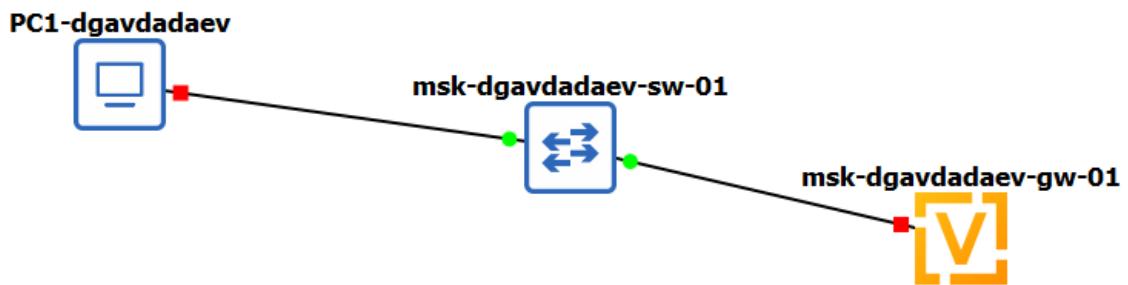


Рис. 2.15: Топология сети с маршрутизатором VyOS

2.4.2 Настройка IP-адресации на оконечном устройстве

После запуска всех устройств проекта была открыта консоль **PC1**. Оконечному устройству был назначен IP-адрес **192.168.1.10/24** с указанием шлюза по умолчанию **192.168.1.1**. После ввода параметров конфигурация была сохранена, а затем выполнен просмотр текущих IP-настроек для контроля корректности конфигурации.

В выводе отображаются назначенный IP-адрес, маска сети, адрес шлюза и MAC-адрес интерфейса.

2.4.3 Настройка маршрутизатора VyOS

После загрузки маршрутизатора VyOS был выполнен вход под стандартной учётной записью. Далее произведён переход в режим конфигурирования. В конфигурации маршрутизатора были выполнены следующие действия: - изменено имя устройства на **msk-dgavdadaev-gw-01**; - на интерфейсе **eth0** задан IP-адрес **192.168.1.1/24**; - удалена настройка получения адреса по DHCP; - просмотрены изменения в конфигурации; - применены и сохранены внесённые изменения.

После сохранения конфигурации был выполнен просмотр информации об интерфейсах маршрутизатора, что позволило убедиться в наличии назначенного IP-адреса и корректном состоянии интерфейса **eth0**.

Процесс настройки маршрутизатора VyOS и вывод информации об интерфейсах представлен на рисунке ниже.

```
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:~$ configure
[edit]
vyos@vyos# set system host-name msk-dgavdadaev-gw-01
[edit]
vyos@vyos# set interfaces ethernet eth0 address 192.168.1.1/24
[edit]
vyos@vyos# delete interfaces ethernet eth0 address dhcp
[edit]
vyos@vyos# compare
[edit interfaces ethernet eth0]
-address dhcp
+address 192.168.1.1/24
[edit system]
>host-name msk-dgavdadaev-gw-01
[edit]
vyos@vyos# commit
[edit]
vyos@vyos# save
Saving configuration to '/config/config.boot'...
Done
[edit]
vyos@vyos# show interfaces
ethernet eth0 {
    address 192.168.1.1/24
    hw-id 0c:9c:10:73:00:00
}
ethernet eth1 {
    hw-id 0c:9c:10:73:00:01
}
ethernet eth2 {
    hw-id 0c:9c:10:73:00:02
}
loopback lo {
}
[edit]
vyos@vyos#
```

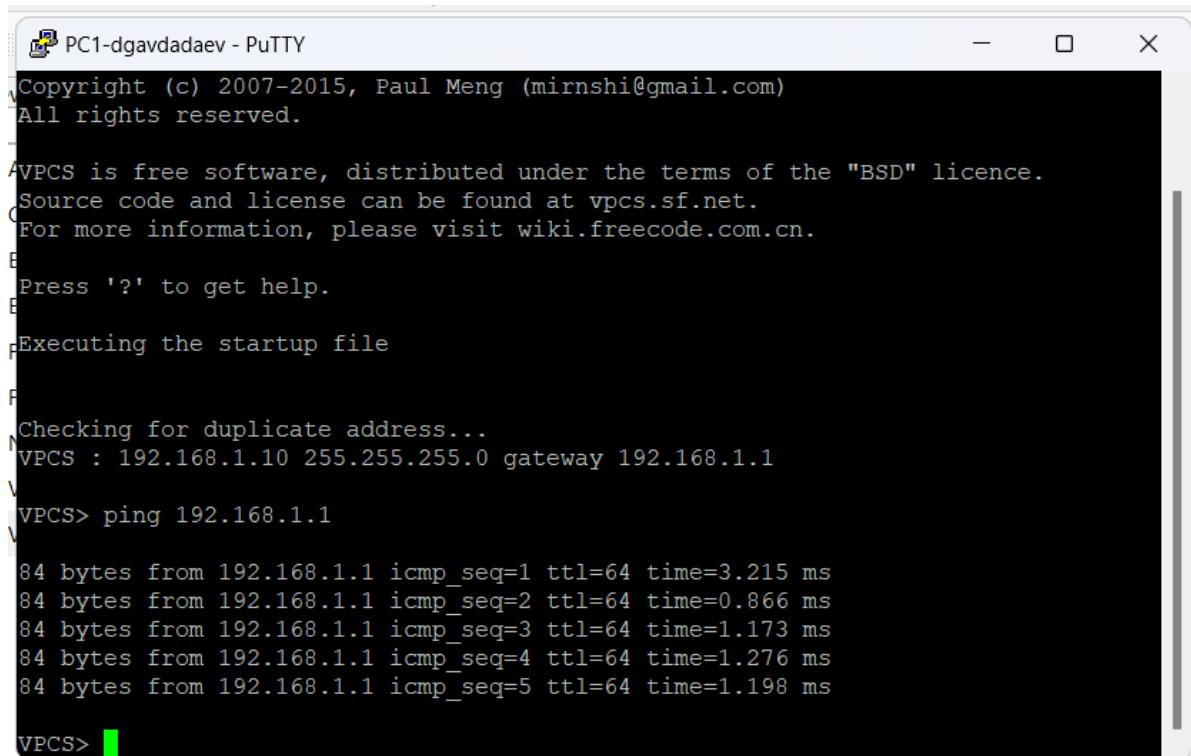
Рис. 2.16: Настройка и проверка интерфейсов маршрутизатора VyOS

2.4.4 Проверка сетевой связности

Для проверки работоспособности сети с оконечного устройства **PC1** был выполнен эхо-запрос к IP-адресу маршрутизатора **192.168.1.1**. В результате были получены ответы на все ICMP-запросы, что подтверждает корректную настрой-

ку IP-адресации и работоспособность сетевого взаимодействия между узлом и маршрутизатором.

Результат выполнения проверки связности представлен на рисунке ниже.



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

VPCS is free software, distributed under the terms of the "BSD" licence.
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

F
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=3.215 ms
84 bytes from 192.168.1.1 icmp_seq=2 ttl=64 time=0.866 ms
84 bytes from 192.168.1.1 icmp_seq=3 ttl=64 time=1.173 ms
84 bytes from 192.168.1.1 icmp_seq=4 ttl=64 time=1.276 ms
84 bytes from 192.168.1.1 icmp_seq=5 ttl=64 time=1.198 ms

VPCS>
```

Рис. 2.17: Ping с PC1 на маршрутизатор VyOS

2.4.5 Анализ трафика в Wireshark

На соединении между коммутатором и маршрутизатором был включён захват трафика. В окне **Wireshark** были зафиксированы ARP- и ICMP-сообщения, возникающие в процессе проверки связности.

В начале обмена наблюдается ARP-запрос от узла **192.168.1.10**, направленный на определение MAC-адреса узла с IP **192.168.1.1**. В ответ маршрутизатор VyOS отправляет ARP-ответ, содержащий свой MAC-адрес. После завершения ARP-разрешения в трассировке фиксируются ICMP Echo Request и Echo Reply, подтверждающие успешный обмен данными.

В деталях ICMP-пакета отображаются основные поля протоколов Ethernet, IPv4

и ICMP, включая тип сообщения, идентификатор, номер последовательности и время отклика.

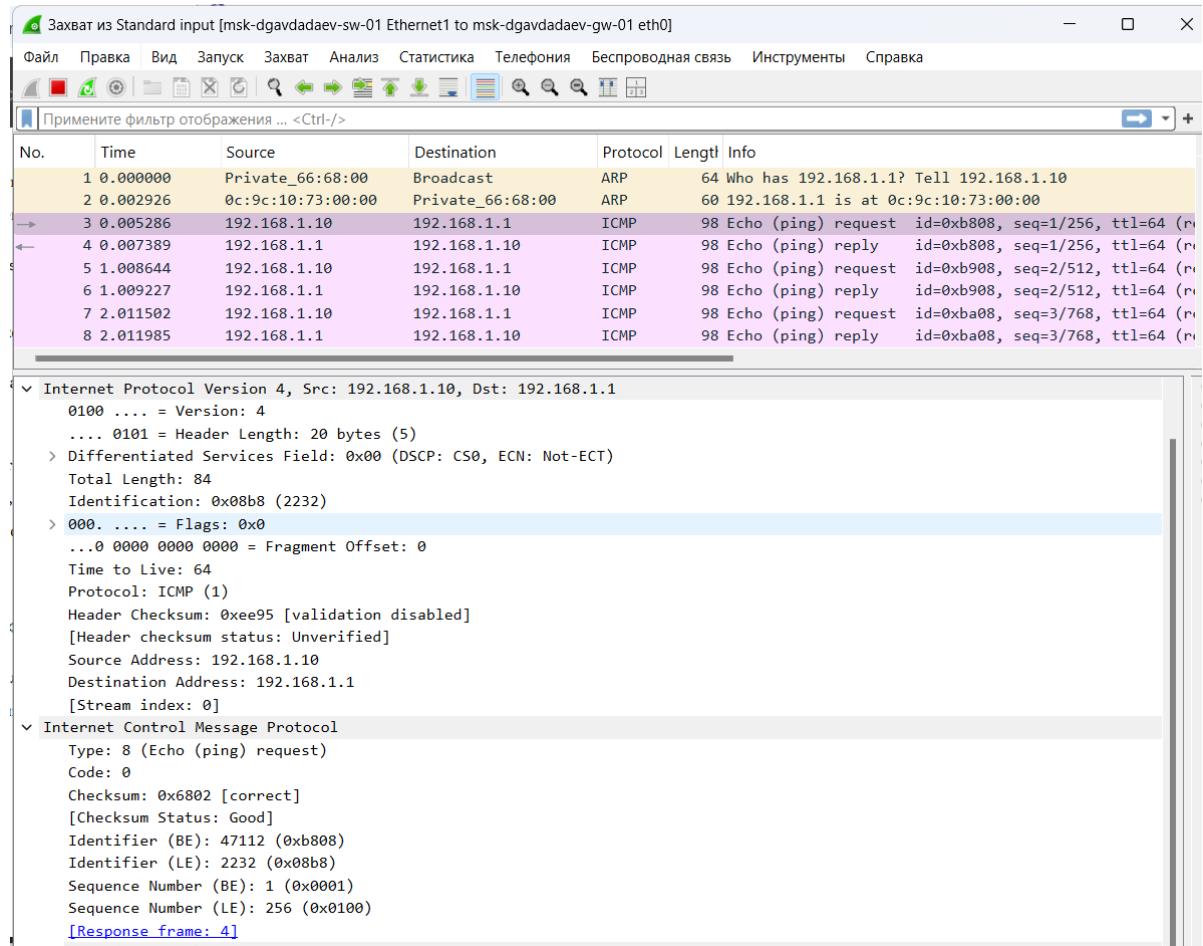


Рис. 2.18: Ping с PC1 на маршрутизатор VyOS

3 Заключение

В ходе выполнения лабораторной работы была освоена базовая настройка и моделирование простейших локальных сетей в среде GNS3 с использованием Ethernet-коммутаторов, оконечных устройств VPCS, а также маршрутизаторов на базе FRR и VyOS. Были построены сетевые топологии различной сложности, выполнена настройка IP-адресации и проверена корректность сетевого взаимодействия между узлами.

В процессе работы подтверждена работоспособность протоколов ARP и ICMP, а также на практике изучены особенности обмена данными при использовании различных режимов эхо-запросов. С помощью анализатора Wireshark был выполнен захват и разбор сетевого трафика, что позволило наглядно проследить процесс разрешения MAC-адресов и передачи ICMP-сообщений.