

Отчёт по лабораторной работе 5

Простые сети в GNS3. Анализ трафика

Гафоров Нурмухаммад

Содержание

1 Введение	5
1.1 Цель работы	5
2 Ход выполнения	6
2.1 Анализ трафика в GNS3 посредством Wireshark	6
2.2 Моделирование простейшей сети на базе маршрутизатора FRR в GNS3	12
2.3 Моделирование простейшей сети на базе маршрутизатора VyOS в GNS3	16
3 Вывод	19

Список иллюстраций

2.1 Топология сети в GNS3	6
2.2 Просмотр справки по командам VPCS	7
2.3 Настройка IP-адресов и проверка связи между узлами	8
2.4 Захват ARP-пакетов в Wireshark	8
2.5 Выполнение ping в различных режимах на PC2	9
2.6 ICMP-запрос и ответ между узлами	10
2.7 Захват UDP-пакетов в Wireshark	11
2.8 Анализ TCP-пакетов в Wireshark	12
2.9 Настройка IP-адреса на ПК	13
2.10 Настройка интерфейса маршрутизатора FRR	14
2.11 Просмотр текущей конфигурации маршрутизатора	14
2.12 Интерфейс маршрутизатора активен	15
2.13 Проверка связи между ПК и маршрутизатором	15
2.14 Анализ ICMP-трафика в Wireshark	16
2.15 Настройка маршрутизатора VyOS	17
2.16 Проверка настроек интерфейсов VyOS	17
2.17 Проверка связи с маршрутизатором VyOS	18
2.18 Захват ICMP и ARP трафика в Wireshark	18

Список таблиц

1 Введение

1.1 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 Анализ трафика в GNS3 посредством Wireshark

В среде **GNS3** была создана простейшая сеть, состоящая из двух виртуальных компьютеров (**PC1-ngaforov**, **PC2-ngaforov**) и коммутатора **msk-ngaforov-sw-01**. Устройства были соединены между собой стандартными Ethernet-кабелями.

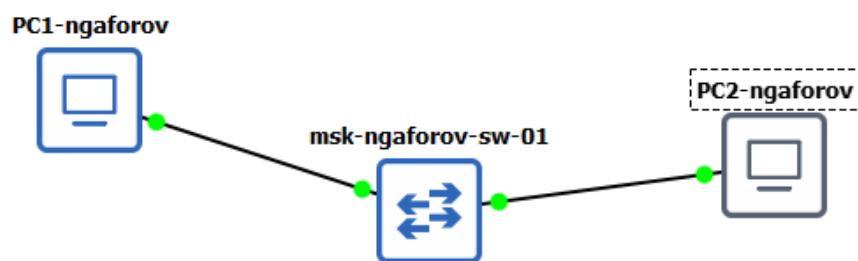
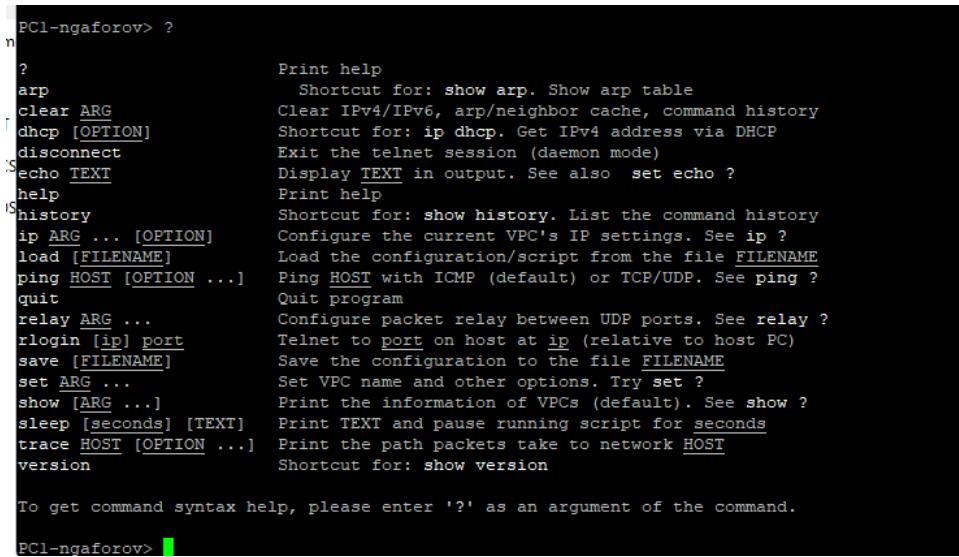


Рис. 2.1: Топология сети в GNS3

После запуска каждого из узлов были открыты консоли **VPCS**. Для проверки до-

ступных команд на терминале **PC1-ngaforov** введена команда `?`, которая вывела справку по доступным функциям, включая работу с IP, ARP, ping и сохранением конфигурации.



```
PC1-ngaforov> ?
?
arp Print help
    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.
PC1-ngaforov>
```

Рис. 2.2: Просмотр справки по командам VPCS

На узле **PC1-ngaforov** был задан IP-адрес **192.168.1.11/24** с указанием шлюза **192.168.1.1** командой `ip 192.168.1.11/24 192.168.1.1`. После проверки отсутствия дублирующего адреса конфигурация была успешно сохранена с помощью команды `save`. Аналогичные действия были выполнены для **PC2-ngaforov**, которому присвоен IP-адрес **192.168.1.12/24** и шлюз **192.168.1.1**.

Далее была выполнена проверка связности между узлами с помощью команды `ping`. Узлы успешно обменялись ICMP-эхо-запросами и ответами, что подтвердило корректную настройку адресов и функционирование коммутатора.

```

PC1-ngaforov> arp
arp [OPTION]           Shortcut for: show arp. Show arp table
clear ARG              Clear IPv4/IPv6, arp/neighbor cache, command history
dumppc [OPTION]         Shortcut for: ip dchp. Get IPv4 address via DHCP
dumpnbia [OPTION]       Shortcut for: ip nbia. Dump neighbor table
echo TEXT              Display TEXT in output. See also set echo ?
help                  Print help
history [ARG]           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                  Exit program
set ARG                Set VPC parameter options. Try SET ?
telnet [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 ...]          Show configuration information. 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.

PC1-ngaforov>
PC1-ngaforov> ip
PC1-ngaforov : ip 192.168.1.11/24 192.168.1.1
Checking for duplicate address...
PC1-ngaforov : 192.168.1.11 255.255.255.0 gateway 192.168.1.1

PC1-ngaforov> save
Saving startup configuration to startup.vpc
. done

PC1-ngaforov> ping 192.168.1.12
64 bytes from 192.168.1.12 icmp_seq=1 ttl=64 time=2.058 ms
64 bytes from 192.168.1.12 icmp_seq=2 ttl=64 time=0.784 ms
64 bytes from 192.168.1.12 icmp_seq=3 ttl=64 time=0.784 ms
64 bytes from 192.168.1.12 icmp_seq=4 ttl=64 time=0.766 ms
64 bytes from 192.168.1.12 icmp_seq=5 ttl=64 time=1.154 ms

PC1-ngaforov> ping 192.168.1.12
64 bytes from 192.168.1.12 icmp_seq=1 ttl=64 time=2.058 ms
64 bytes from 192.168.1.12 icmp_seq=2 ttl=64 time=0.784 ms
64 bytes from 192.168.1.12 icmp_seq=3 ttl=64 time=0.784 ms
64 bytes from 192.168.1.12 icmp_seq=4 ttl=64 time=0.766 ms
64 bytes from 192.168.1.12 icmp_seq=5 ttl=64 time=1.154 ms

PC1-ngaforov> ping 192.168.1.12
64 bytes from 192.168.1.12 icmp_seq=1 ttl=64 time=1.861 ms
64 bytes from 192.168.1.12 icmp_seq=2 ttl=64 time=1.204 ms
64 bytes from 192.168.1.12 icmp_seq=3 ttl=64 time=0.739 ms
64 bytes from 192.168.1.12 icmp_seq=4 ttl=64 time=0.737 ms
64 bytes from 192.168.1.12 icmp_seq=5 ttl=64 time=0.730 ms

PC2-ngaforov> ping 192.168.1.11

```

Рис. 2.3: Настройка IP-адресов и проверка связи между узлами

Для анализа сетевого обмена был запущен **Wireshark** на соединении между коммутатором и ПК **PC1**. В момент активации узлов зафиксированы **ARP**-кадры типа **Gratuitous ARP**, указывающие, что устройства объявляют свои IP-адреса в сети (192.168.1.11 и 192.168.1.12).

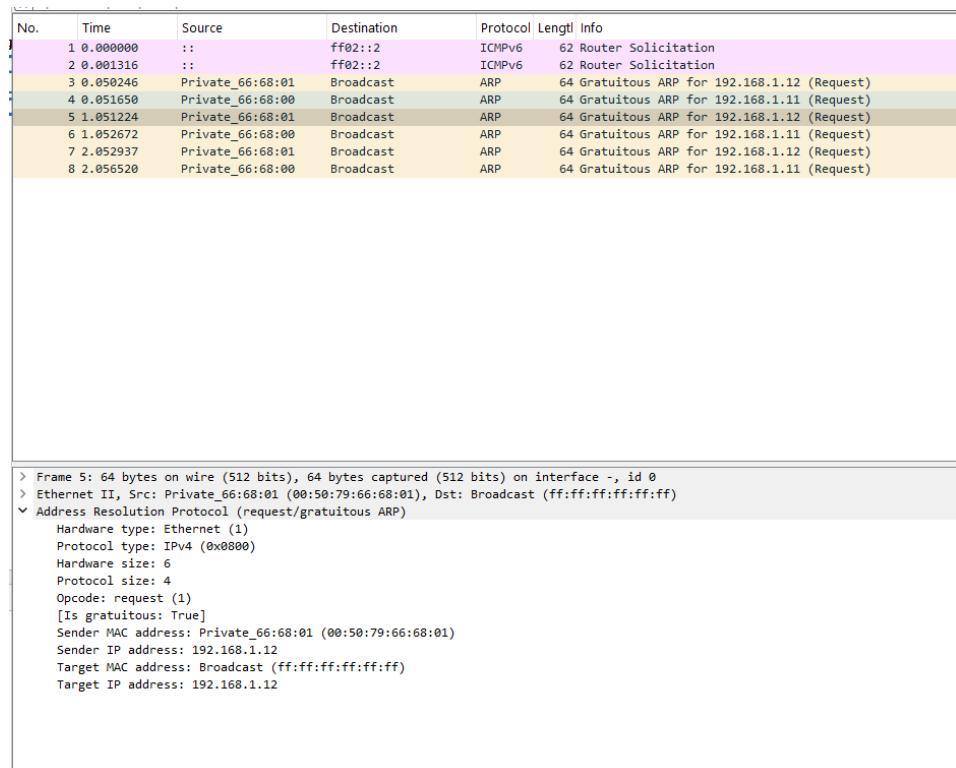


Рис. 2.4: Захват ARP-пакетов в Wireshark

Далее на **PC2-ngaforov** были протестированы различные режимы работы команды ping. После вывода справки (ping /?) последовательно были выполнены эхо-запросы в режимах **ICMP**, **UDP** и **TCP** к адресу **192.168.1.11**.

```
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.

PC2-ngaforov> ping 192.168.1.1 -c 1
host (192.168.1.1) not reachable

PC2-ngaforov> ping 192.168.1.11 -c 1
84 bytes from 192.168.1.11 icmp_seq=1 ttl=64 time=1.826 ms

PC2-ngaforov> ping 192.168.1.11 -c 1 -2
84 bytes from 192.168.1.11 udp_seq=1 ttl=64 time=1.523 ms

PC2-ngaforov> ping 192.168.1.11 -c 1 -3
3 Connect  7@192.168.1.11 seq=1 ttl=64 time=2.827 ms
wSendData  7@192.168.1.11 seq=1 ttl=64 time=2.533 ms
mClose     7@192.168.1.11 seq=1 ttl=64 time=4.444 ms
PC2-ngaforov>
```

Рис. 2.5: Выполнение ping в различных режимах на PC2

В **Wireshark** отобразились соответствующие типы пакетов. При ICMP-запросе наблюдался стандартный обмен пакетами **Echo Request** и **Echo Reply** между узлами **192.168.1.11** и **192.168.1.12**, что подтверждает корректную реализацию базового ICMP-протокола.

No.	Time	Source	Destination	Protocol	Length	Info
3	0.050246	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.12 (Request)
4	0.051650	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.11 (Request)
5	1.051224	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.12 (Request)
6	1.052672	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.11 (Request)
7	2.052937	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.12 (Request)
8	2.056520	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.11 (Request)
9	54.030194	Private_66:68:01	Broadcast	ARP	64	Who has 192.168.1.1? Tell 192.168.1.12
10	55.030911	Private_66:68:01	Broadcast	ARP	64	Who has 192.168.1.1? Tell 192.168.1.12
11	56.031626	Private_66:68:01	Broadcast	ARP	64	Who has 192.168.1.1? Tell 192.168.1.12
12	61.614174	Private_66:68:01	Broadcast	ARP	64	Who has 192.168.1.11? Tell 192.168.1.12
13	61.614599	Private_66:68:00	Private_66:68:01	ARP	64	192.168.1.11 is at 00:50:79:66:68:00
→	14 61.616298	192.168.1.12	192.168.1.11	ICMP	98	Echo (ping) request id=0x3341, seq=1/256, ttl=64 (reply in 15)
←	15 61.616941	192.168.1.11	192.168.1.12	ICMP	98	Echo (ping) reply id=0x3341, seq=1/256, ttl=64 (request in 14)
16	74.453887	192.168.1.12	192.168.1.11	ECHO	98	Request
17	74.454788	192.168.1.12	192.168.1.11	ECHO	98	Response
18	78.111416	192.168.1.12	192.168.1.11	TCP	74	17298 → 7 [SYN] Seq=0 Win=2920 Len=0 MSS=1460 TSeqval=1761558852 TSecr=0
19	78.112282	192.168.1.11	192.168.1.12	TCP	54	7 → 17298 [SYN, ACK] Seq=0 Ack=1 Win=2920 Len=0
20	78.113918	192.168.1.12	192.168.1.11	TCP	66	17298 → 7 [ACK] Seq=1 Ack=1 Win=2920 Len=0 TSeqval=1761558852 TSecr=0
21	78.115400	192.168.1.12	192.168.1.11	ECHO	122	Request
22	78.115995	192.168.1.11	192.168.1.12	TCP	54	7 → 17298 [ACK] Seq=1 Ack=57 Win=2920 Len=0
23	78.119390	192.168.1.12	192.168.1.11	TCP	66	17298 → 7 [FIN, PSH, ACK] Seq=57 Ack=1 Win=0 TSeqval=1761558852 TSecr=0
24	78.121334	192.168.1.11	192.168.1.12	TCP	54	7 → 17298 [ACK] Seq=1 Ack=58 Win=2920 Len=0
25	78.121380	192.168.1.11	192.168.1.12	TCP	54	7 → 17298 [FIN, ACK] Seq=1 Ack=58 Win=2920 Len=0 TSeqval=1761558852 TSecr=0
26	78.123690	192.168.1.12	192.168.1.11	TCP	66	17298 → 7 [ACK] Seq=58 Ack=2 Win=2920 Len=0 TSeqval=1761558852 TSecr=0

> Frame 14: 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
▼ Internet Control Message Protocol
 Type: 8 (Echo (ping) request)
 Code: 0
 Checksum: 0xecc9 [correct]
 [Checksum Status: Good]
 Identifier (BE): 13121 (0x3341)
 Identifier (LE): 16691 (0x4133)
 Sequence Number (BE): 1 (0x0001)
 Sequence Number (LE): 256 (0x0100)
 [Response frame: 15]
 Data (56 bytes)
 Data: 08090a0b0c0d0e0f101111311415161718191a1b1c1d1e1f20212223242526272829a2b2c2d2e2f303132333435363738393a3b3c3d3e3f
 [Length: 56]

0000 00 50 7
0010 00 54 4
0020 01 00 0
0030 0e 0f 1
0040 1e 1f 2
0050 2e 2f 3
0060 3e 3f

Рис. 2.6: ICMP-запрос и ответ между узлами

В режиме **UDP** зафиксированы пакеты, передаваемые с порта **16417** на порт **7**, содержащие эхо-данные. Эти кадры имеют протокол **UDP** и длину 64 байта, что соответствует эхо-запросу без подтверждения на транспортном уровне.

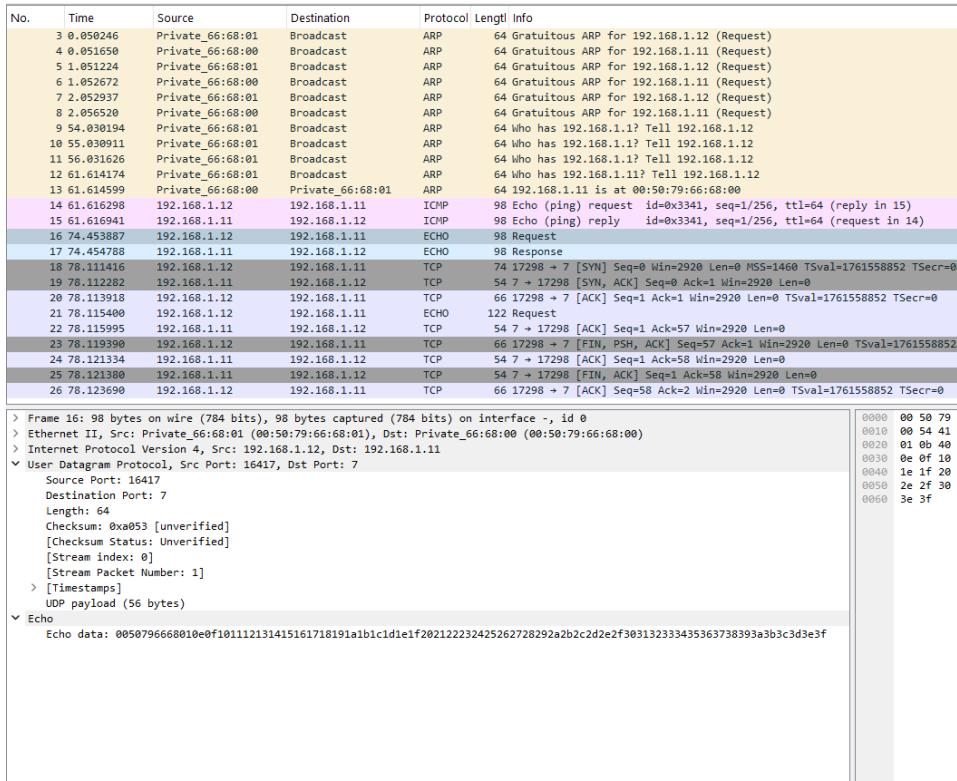


Рис. 2.7: Захват UDP-пакетов в Wireshark

При использовании **TCP** наблюдалось установление соединения (последовательность **SYN, SYN-ACK, ACK**) и передача эхо-данных в рамках установленной сессии. В окне **Wireshark** видно, что TCP-соединение было успешно завершено с обменом сегментов подтверждения (**FIN, ACK**).

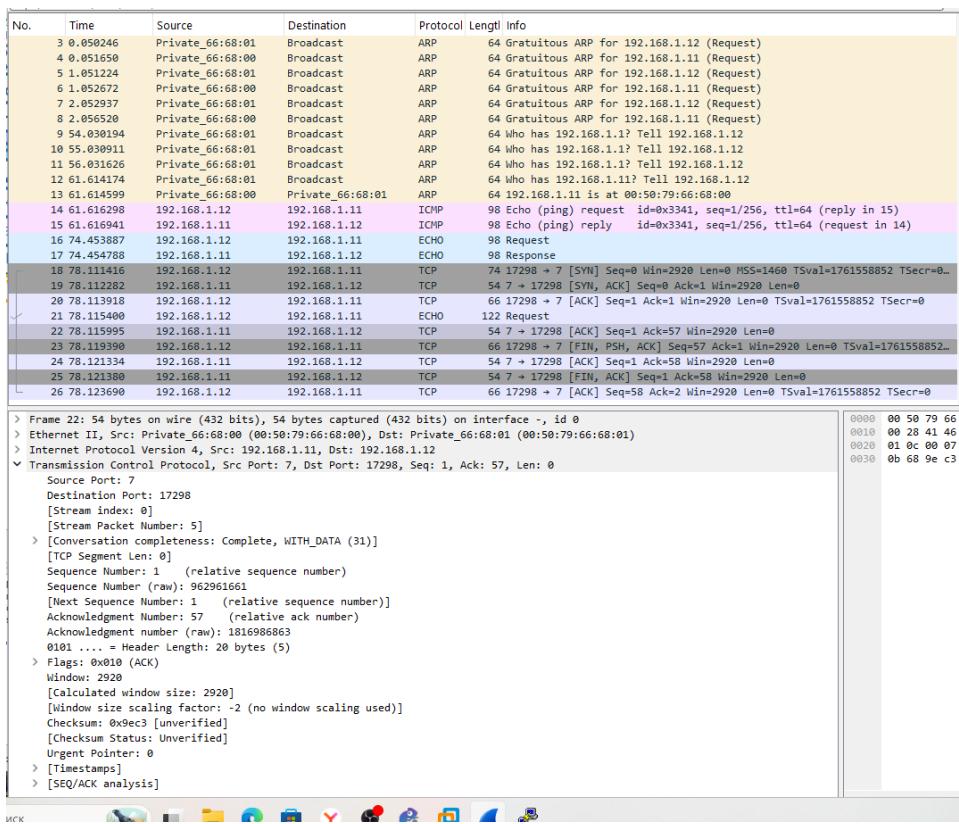


Рис. 2.8: Анализ TCP-пакетов в Wireshark

2.2 Моделирование простейшей сети на базе маршрутизатора FRR в GNS3

В среде **GNS3** была смоделирована сеть, состоящая из одного оконечного устройства **PC1-ngaforov**, коммутатора **msk-ngaforov-sw-01** и маршрутизатора **msk-ngaforov-gw-01** на базе **FRRouting**.

На узле **PC1-ngaforov** с помощью консоли **VPCS** был задан IP-адрес **192.168.1.10/24** и шлюз **192.168.1.1**, после чего конфигурация сохранена. Команда **show ip** отобразила параметры сетевого интерфейса: имя устройства, IP-адрес, маску подсети, шлюз и MAC-адрес.

```
Checking for duplicate address...
PC1-ngaforov : 192.168.1.11 255.255.255.0 gateway 192.168.1.1

PC1-ngaforov> ip 192.168.1.10/24 192.168.1.1
Checking for duplicate address...
PC1-ngaforov : 192.168.1.10 255.255.255.0 gateway 192.168.1.1

PC1-ngaforov> save
Saving startup configuration to startup.vpc
. done

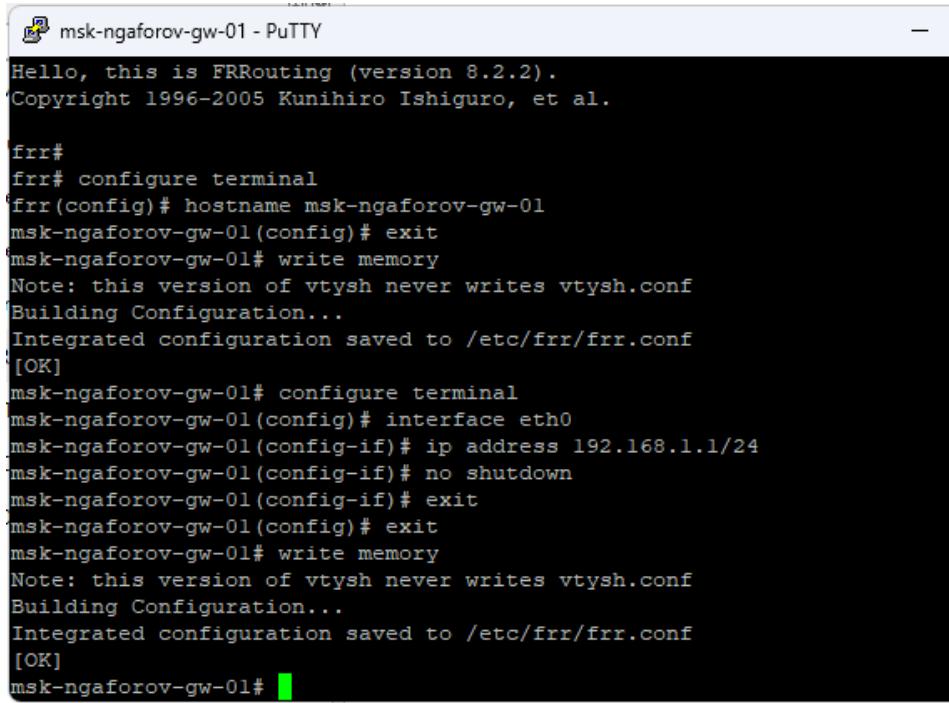
PC1-ngaforov> show ip

NAME      : PC1-ngaforov[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

PC1-ngaforov>
```

Рис. 2.9: Настройка IP-адреса на ПК

На маршрутизаторе **msk-ngaforov-gw-01** под управлением **FRR** выполнена базовая настройка. В конфигурационном режиме был задан IP-адрес **192.168.1.1/24** на интерфейсе **eth0**, интерфейс активирован, а конфигурация сохранена в файл **/etc/frr/frr.conf**.

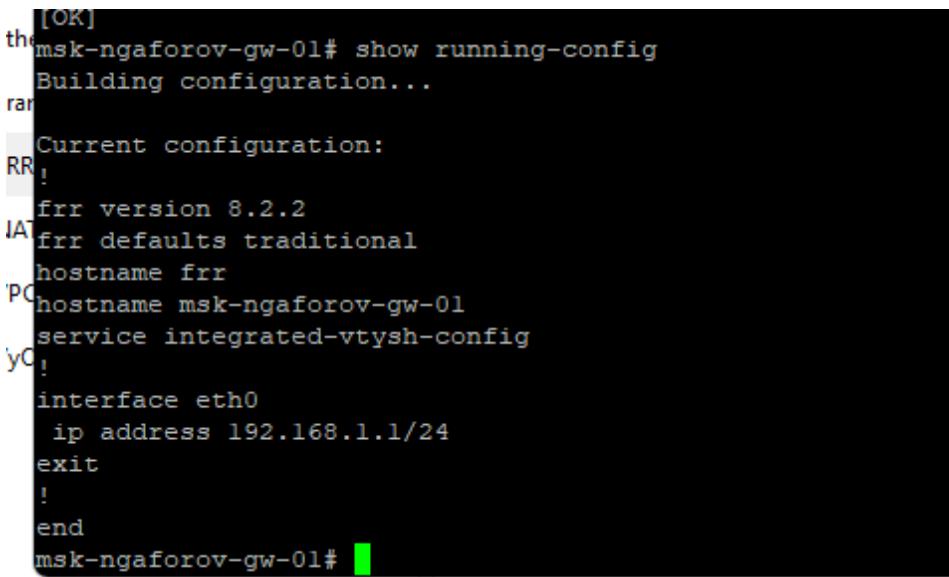


```
msk-ngaforov-gw-01 - PuTTY
Hello, this is FRRouting (version 8.2.2).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

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

Рис. 2.10: Настройка интерфейса маршрутизатора FRR

Проверка текущей конфигурации с помощью команды **show running-config** показала, что интерфейсу **eth0** успешно присвоен адрес **192.168.1.1/24**.



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

Рис. 2.11: Просмотр текущей конфигурации маршрутизатора

Команда **show interface brief** подтвердила, что интерфейс **eth0** находится в

состоянии **up** и имеет правильные параметры адресации.

```
msk-ngaforov-gw-01#  
msk-ngaforov-gw-01# show interface brief  
Interface      Status   VRF          Addresses  
R-----  
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-ngaforov-gw-01#
```

Рис. 2.12: Интерфейс маршрутизатора активен

После настройки маршрутизатора была проведена проверка связи между ПК и шлюзом с помощью ICMP-запроса. Команда **ping 192.168.1.1** показала успешный обмен эхо-пакетами без потерь, что подтвердило корректную работу сетевой конфигурации.

```
PC1-ngaforov>  
PC1-ngaforov> ping 192.168.1.1  
  
84 bytes from 192.168.1.1 icmp_seq=1 ttl=64 time=3.288 ms  
84 bytes from 192.168.1.1 icmp_seq=2 ttl=64 time=2.183 ms  
84 bytes from 192.168.1.1 icmp_seq=3 ttl=64 time=1.897 ms  
84 bytes from 192.168.1.1 icmp_seq=4 ttl=64 time=1.927 ms  
84 bytes from 192.168.1.1 icmp_seq=5 ttl=64 time=1.371 ms  
  
PC1-ngaforov>
```

Рис. 2.13: Проверка связи между ПК и маршрутизатором

В процессе анализа трафика в **Wireshark** были зафиксированы ARP-запросы, а также **ICMP Echo Request** и **Echo Reply** между узлами **192.168.1.10** и **192.168.1.1**, что демонстрирует успешную двустороннюю передачу данных.

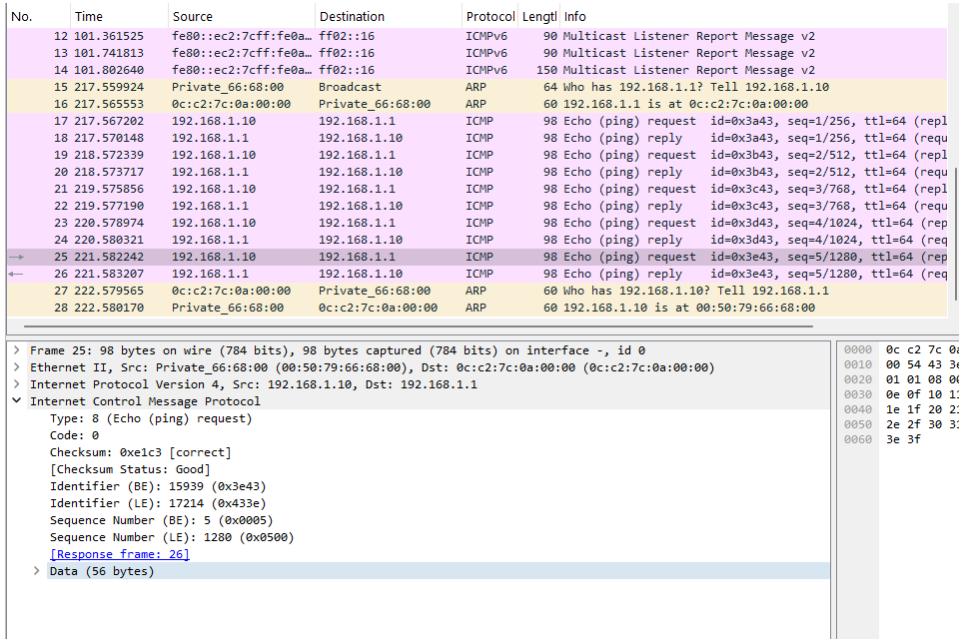


Рис. 2.14: Анализ ICMP-трафика в Wireshark

2.3 Моделирование простейшей сети на базе маршрутизатора VyOS в GNS3

Далее была смоделирована аналогичная топология с использованием маршрутизатора **VyOS**.

После загрузки устройства **VyOS** в режиме конфигурирования был изменён хостнейм на **msk-ngaforov-gw-01**, а интерфейсу **eth0** присвоен IP-адрес **192.168.1.1/24**. Параметр DHCP был удалён, изменения сохранены и применены.

```
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-ngaforov-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-ngaforov-gw-01  
[edit]  
vyos@vyos#
```

Рис. 2.15: Настройка маршрутизатора VyOS

Просмотр конфигурации интерфейсов подтвердил корректность настроек: интерфейс **eth0** активен, имеет IP-адрес **192.168.1.1/24** и аппаратный адрес **0c:99:d0:40:00:00**.

```
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:99:d0:40:00:00  
}  
ethernet eth1 {  
    hw-id 0c:99:d0:40:00:01  
}  
ethernet eth2 {  
    hw-id 0c:99:d0:40:00:02  
}  
loopback lo {  
}  
[edit]  
vyos@vyos#
```

Рис. 2.16: Проверка настроек интерфейсов VyOS

На узле **PC1-ngaforov** был успешно выполнен ICMP-тест с маршрутизатором VyOS. Все пакеты дошли без потерь, что свидетельствует о правильной настройке сетевых параметров.

```

PC1-ngaforov> ping 192.168.1.1

84 bytes from 192.168.1.1 icmp_seq=1 ttl=64 time=3.167 ms
84 bytes from 192.168.1.1 icmp_seq=2 ttl=64 time=1.166 ms
84 bytes from 192.168.1.1 icmp_seq=3 ttl=64 time=2.216 ms
84 bytes from 192.168.1.1 icmp_seq=4 ttl=64 time=3.950 ms
84 bytes from 192.168.1.1 icmp_seq=5 ttl=64 time=2.634 ms

PC1-ngaforov>

```

Рис. 2.17: Проверка связи с маршрутизатором VyOS

Анализ трафика в **Wireshark** показал обмен ICMP-пакетами между узлами **192.168.1.10** и **192.168.1.1**, а также ARP-запросы, что подтверждает корректное функционирование сетевого взаимодействия.

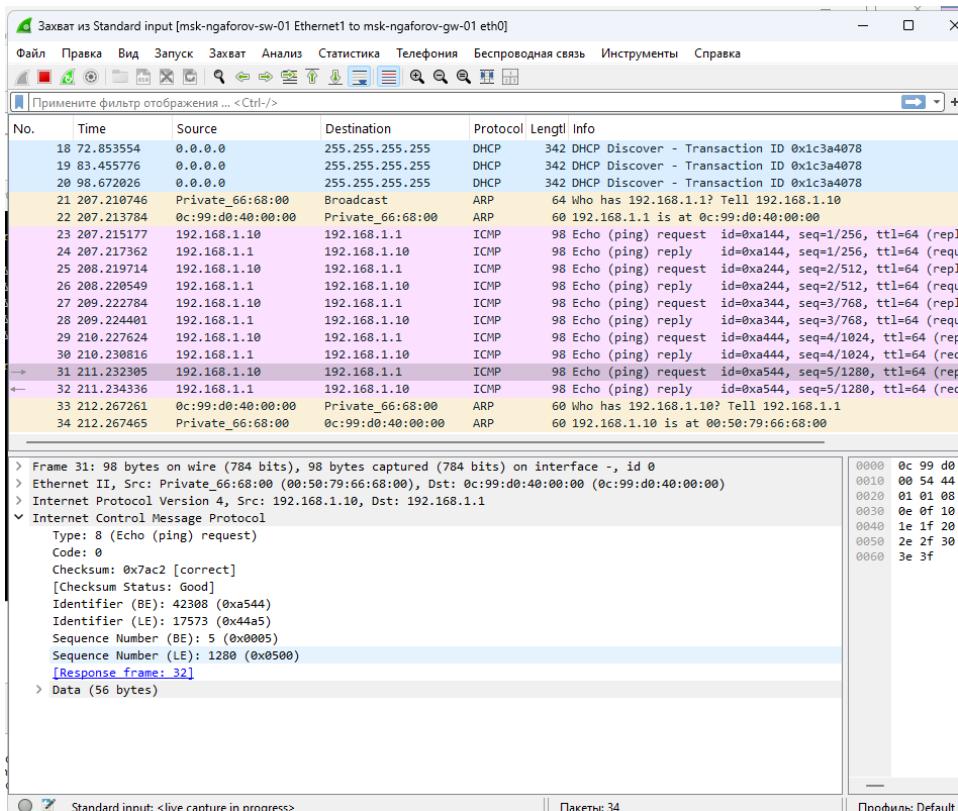


Рис. 2.18: Захват ICMP и ARP трафика в Wireshark

3 Вывод

В ходе лабораторной работы была смоделирована и исследована работа простейших сетей на базе коммутатора и маршрутизаторов **FRR** и **VyOS** в среде **GNS3**. С помощью анализатора трафика **Wireshark** были зафиксированы пакеты протоколов **ARP**, **ICMP**, **UDP** и **TCP**, что позволило изучить принципы их функционирования.

При анализе TCP-трафика подтверждена корректная реализация механизма установления соединения по схеме **Three-Way Handshake**, включающей этапы **SYN**, **SYN/ACK** и **ACK**. Это обеспечивает надёжное установление соединения перед передачей данных и синхронизацию номеров последовательностей между узлами.

Полученные результаты демонстрируют корректную работу сетевых протоколов на канальном, сетевом и транспортном уровнях модели **OSI**, а также подтверждают понимание принципов взаимодействия узлов в локальной сети и процессов обмена данными при передаче запросов и ответов.