

Лабораторная работа №5 (Простые сети в GNS3. Анализ трафика)

Шубина София Антоновна

31 октября 2025

Российский университет дружбы народов

Цель работы

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

Задание

1. Смоделировать простейшую сеть на базе коммутатора в GNS3
2. Проанализировать трафик в GNS3 посредством Wireshark
3. Смоделировать простейшую сеть на базе маршрутизатора FRR в GNS3
4. Смоделировать простейшую сеть на базе маршрутизатора VyOS в GNS3

Выполнение лабораторной работы

Моделирование простейшей сети на базе коммутатора в GNS3

Запустим GNS3 VM и GNS3 и создадим новый проект.

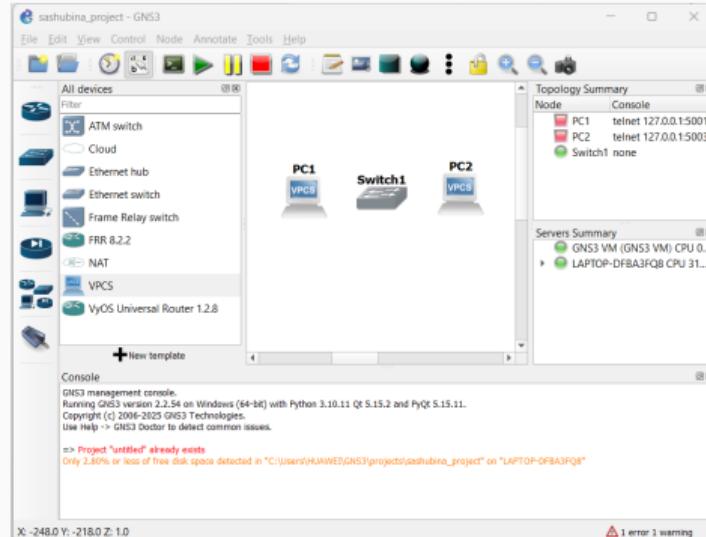


Рис. 1: Добавление устройств

изменение имен

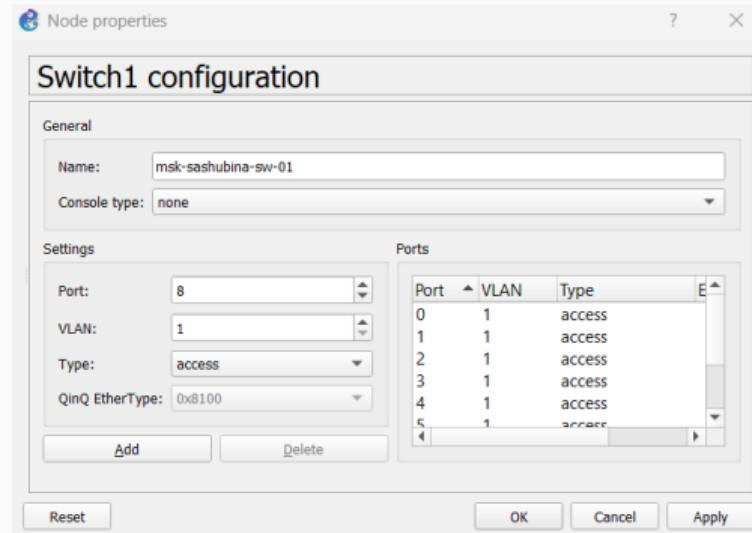


Рис. 2: изменение имен

изменение имен

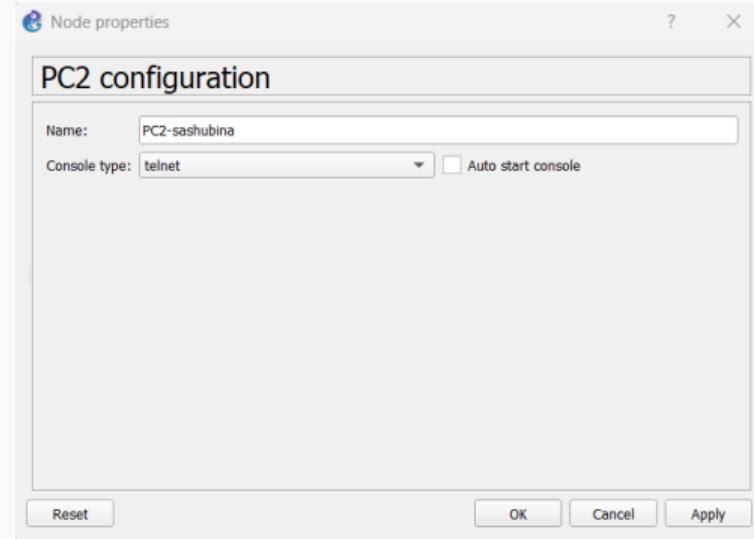


Рис. 3: изменение имен

изменение имен

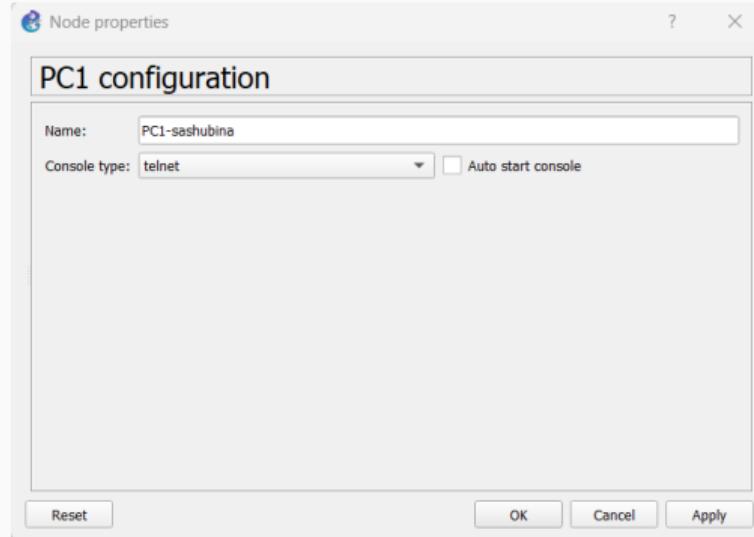


Рис. 4: изменение имен

соединение

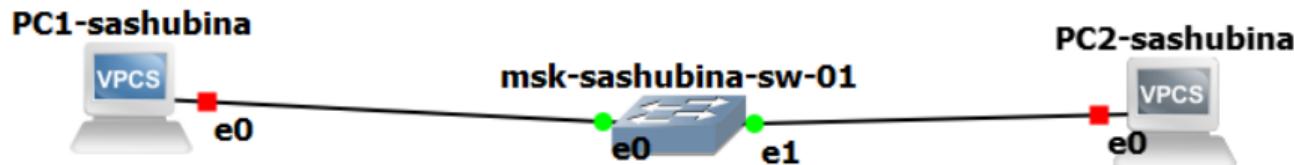
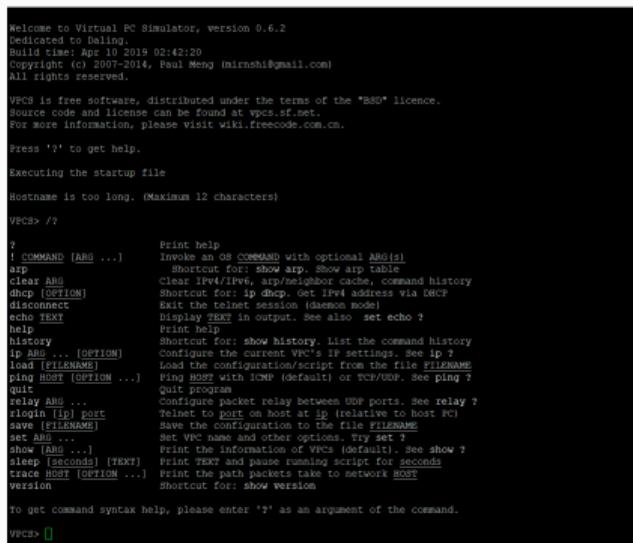


Рис. 5: соединение

Зададим IP-адреса VPCS. Для этого с помощью меню, вызываемого правой кнопкой мыши, запустим Start, PC-1, затем вызовим его терминал Console. Для просмотра синтаксиса возможных для ввода команд наберем `?`.



Welcome to Virtual PC Simulator, version 0.6.2
Dedicated to Beijing.
Build time: Apr 10 2019 02:42:20
Copyright (c) 2007-2014, 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

Hostname is too long. (Maximum 12 characters)

VPCS> ?

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

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

VPCS> []

Рис. 6: Параметры импорта

Для задания IP-адреса 192.168.1.11 в сети 192.168.1.0/24 введем:

```
VPCS> ip 192.168.1.11/24 192.168.1.1  
Checking for duplicate address...  
PC1 : 192.168.1.11 255.255.255.0 gateway 192.168.1.1
```

Рис. 7: Задание IP-адреса PC1-sashubina

А также посмотрим ip адрес, для проверки

```
PC1> show ip

NAME          : PC1[1]
IP/MASK       : 192.168.1.11/24
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> █
```

Рис. 8: show ip

Те же действия проделаем для второго VPC:

```
PC2> ip 192.168.1.12/24 192.168.1.1
Checking for duplicate address...
PC1 : 192.168.1.12 255.255.255.0 gateway 192.168.1.1

PC2> ip show
Invalid address

PC2> show ip

NAME      : PC2[1]
IP/MASK   : 192.168.1.12/24
GATEWAY   : 192.168.1.1
DNS       :
MAC       : 00:50:79:66:68:01
LPORT     : 10006
RHOST:PORT : 127.0.0.1:10007
MTU:      : 1500

PC2> █
```

Рис. 9: Задание IP-адреса PC2-sashubina

Проверим работоспособность соединения между PC-1 и PC-2 с помощью команды ping.

```
PC1> ping 192.168.1.12
84 bytes from 192.168.1.12 icmp_seq=1 ttl=64 time=0.280 ms
84 bytes from 192.168.1.12 icmp_seq=2 ttl=64 time=0.447 ms
84 bytes from 192.168.1.12 icmp_seq=3 ttl=64 time=0.372 ms
84 bytes from 192.168.1.12 icmp_seq=4 ttl=64 time=0.396 ms
84 bytes from 192.168.1.12 icmp_seq=5 ttl=64 time=0.431 ms
```

Рис. 10: Проверка соединения между PC-1 и PC-2

В конце остановим в проекте все узлы(меню GNS3 Control Stop all nodes).

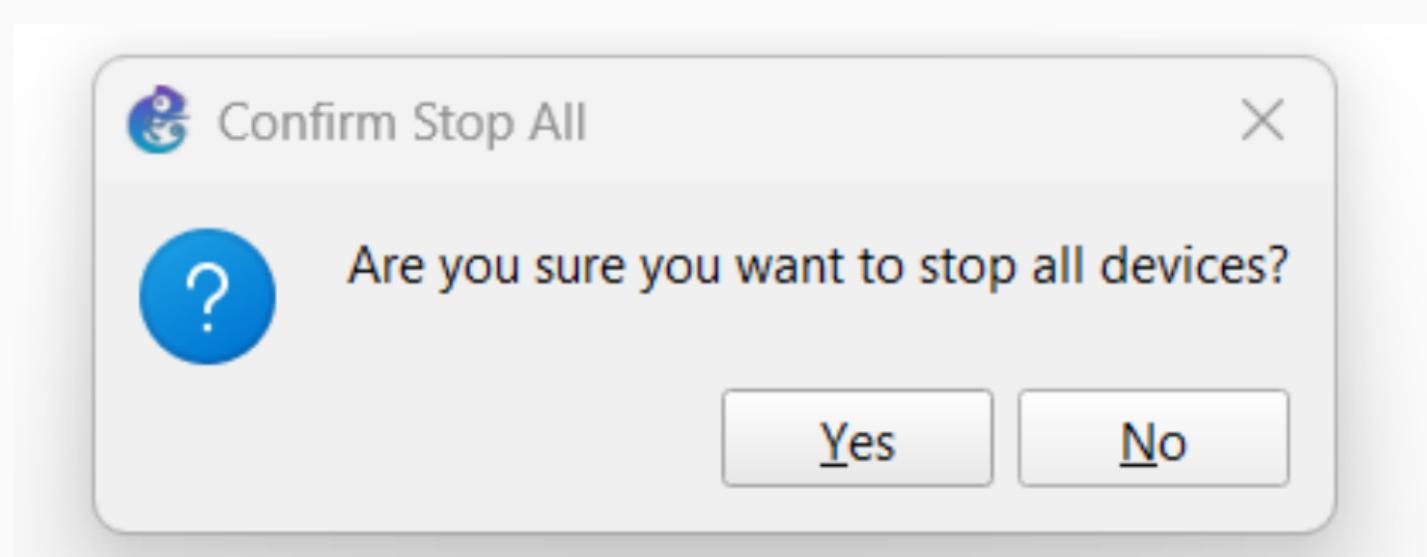


Рис. 11: Остановка всех узлов

Анализ трафика в GNS3 посредством Wireshark

Запустим на соединении между PC-1 и коммутатором анализатор трафика. Для этого щёлкнём правой кнопкой мыши на соединении, выберем в меню Start capture.

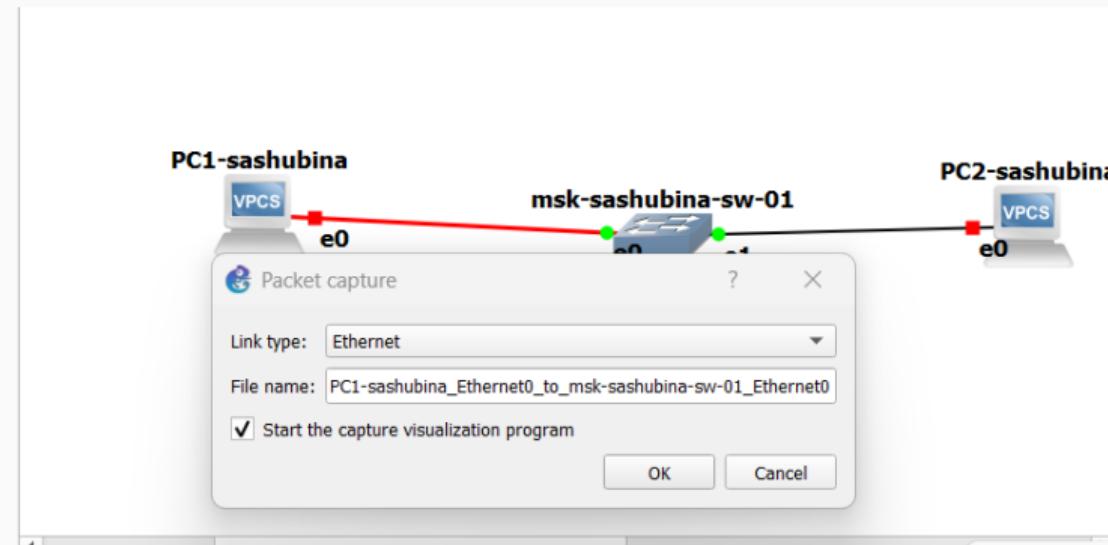


Рис. 12: Запуск анализатора трафика

Запустился Wireshark, а в проекте GNS3 на соединении появился знак.

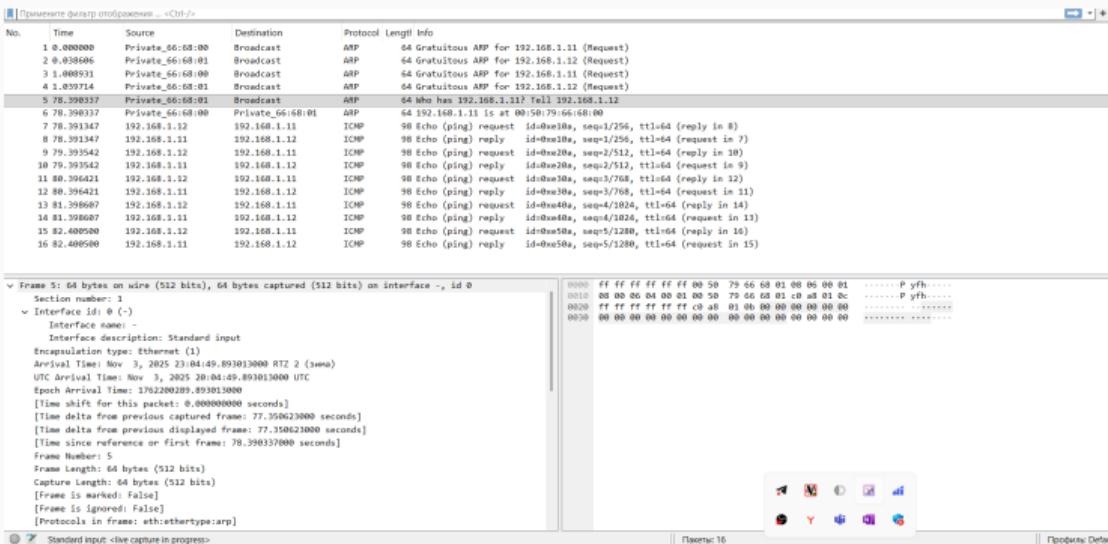


Рис. 13: ARP пакеты

ARP пакеты

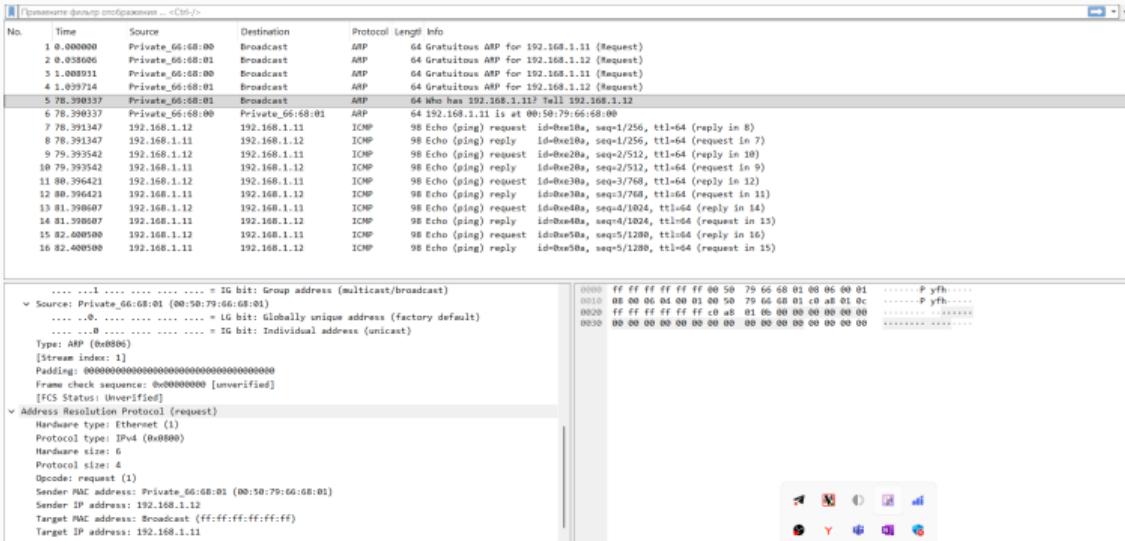


Рис. 14: ARP пакеты

ARP пакеты

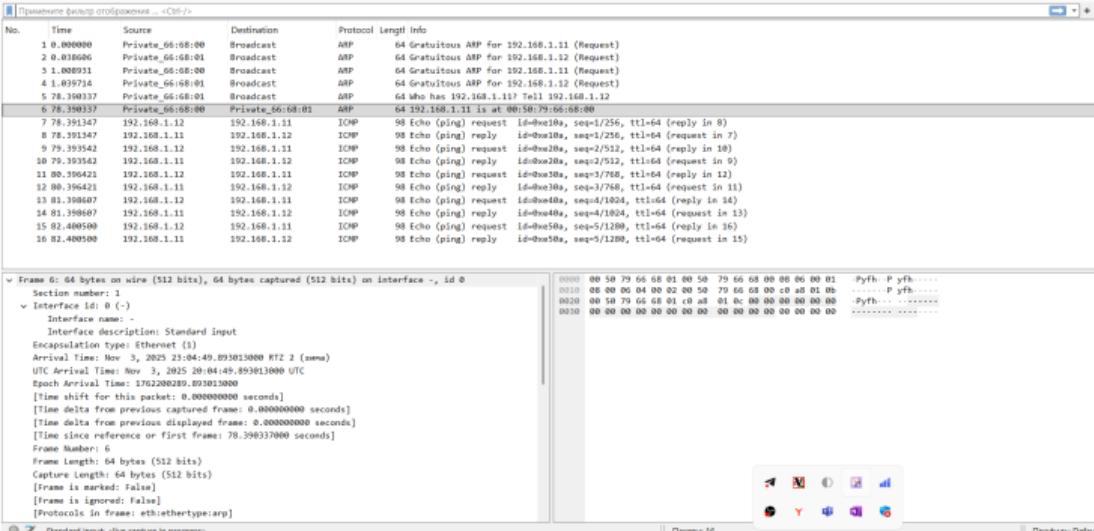


Рис. 15: ARP пакеты

ARP пакеты

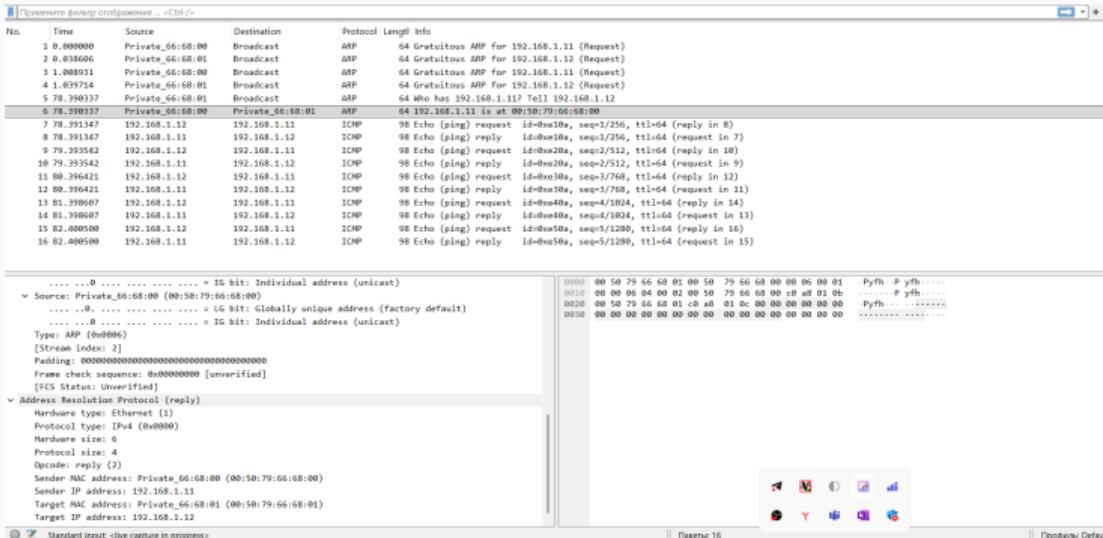


Рис. 16: ARP пакеты

В терминале PC-2 посмотрим информацию по опциям команды ping, введя ping /?.

```
PC2> 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|R|S|R|
                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> █
```

Рис. 17: опции команды ping

Затем сделаем один эхо-запрос в ICMP-моде к узлу PC-1. Изучим эхо-запрос и эхо-ответ ICMP в программе Wireshark. В обоих случаях длина кадра равняется 98 байт. В случае эхо-запроса точка назначения – PC-1, а источник – PC-2, в случае же эхо-ответа – наоборот.

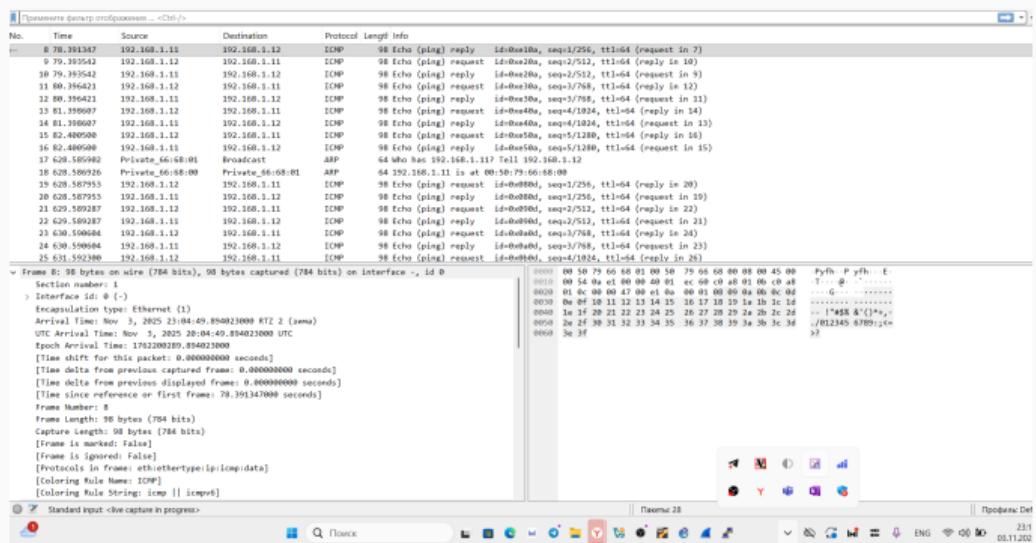


Рис. 18: Эхо-ответ в ICMP-моде

Эхо-ответ в ICMP-моде

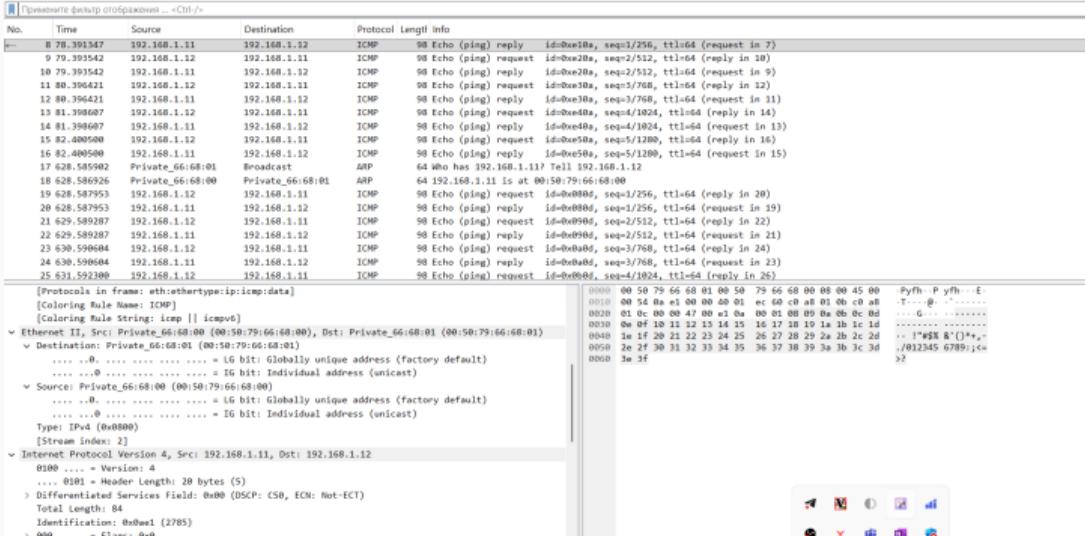


Рис. 19: Эхо-ответ в ICMP-моде

Эхо-запрос в ICMP-моде

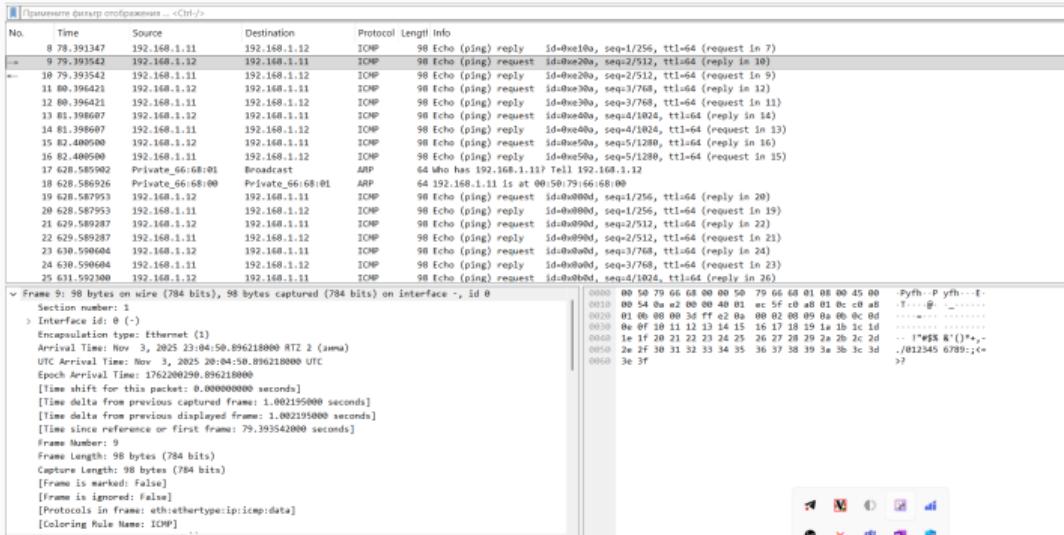
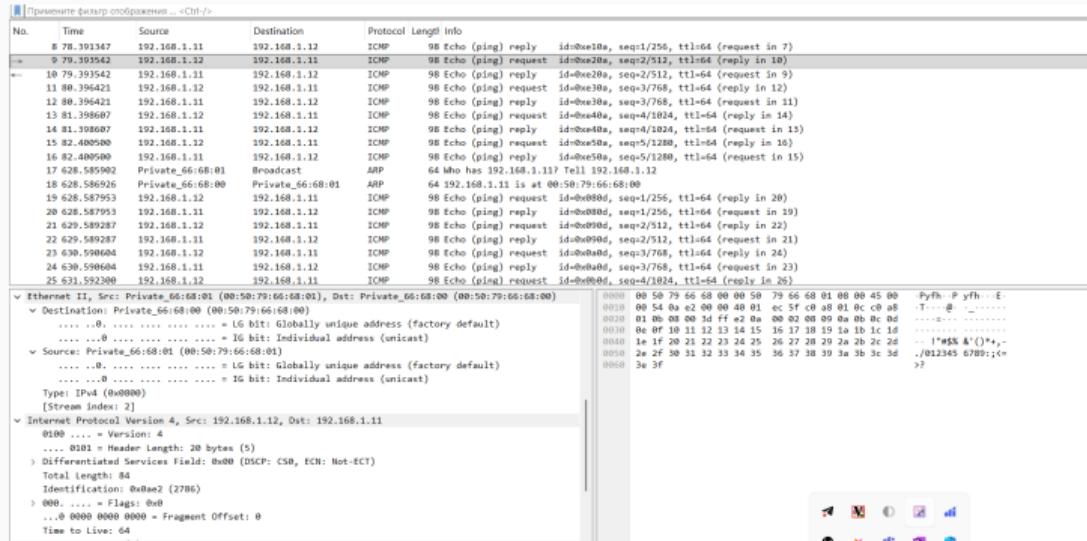


Рис. 20: Эхо-запрос в ICMP-моде

Эхо-запрос в ICMP-моде



Сделаем один эхо-запрос в UDP-моде к узлу PC-1. В окне Wireshark проанализируем полученную информацию. В обоих случаях длина кадра равняется 98 байт. В случае эхо-запроса точка назначения – PC-1, а источник – PC-2, в случае же эхо-ответа – наоборот.

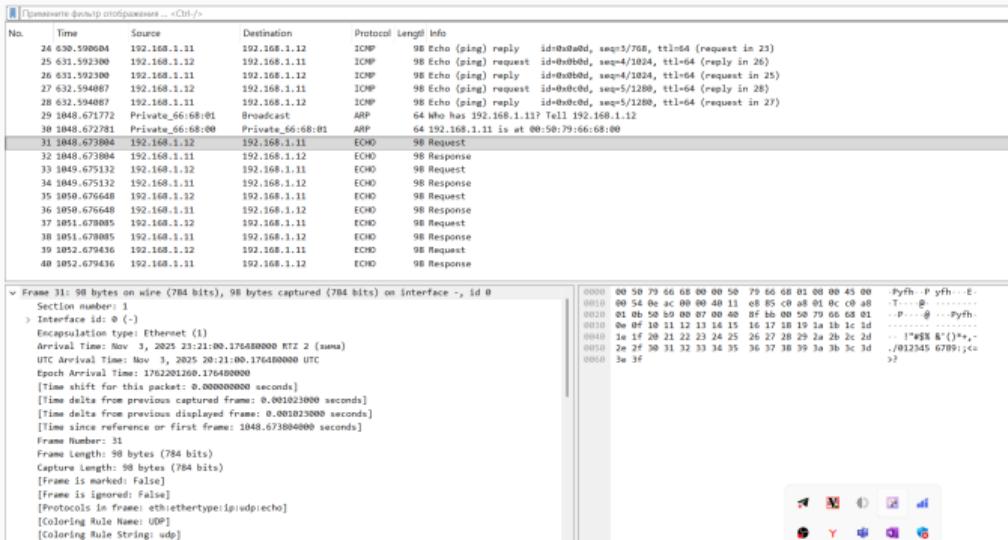


Рис. 22: Эхо-запрос в UDP-моде

Эхо-запрос в UDP-моде

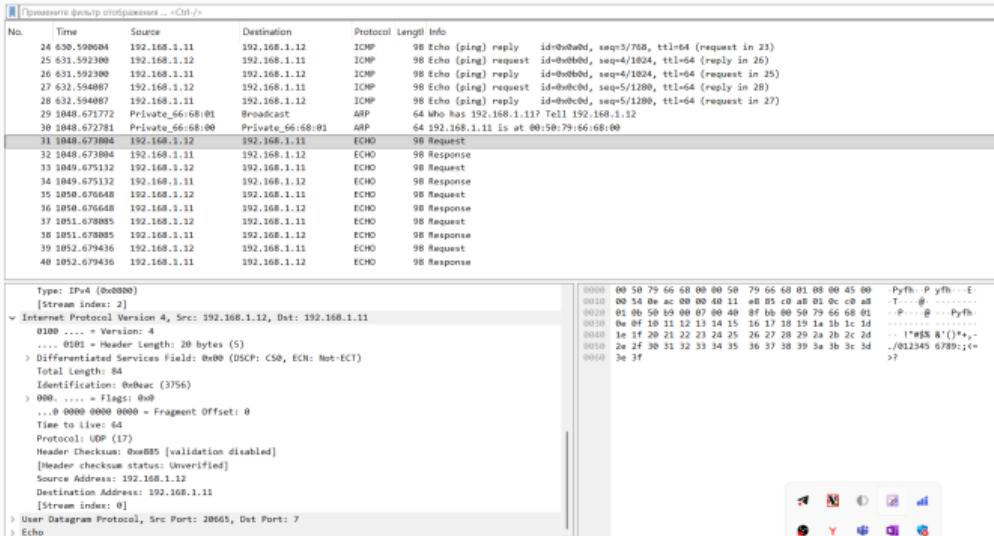


Рис. 23: Эхо-запрос в UDP-моде

Эхо-ответ в UDP-моде

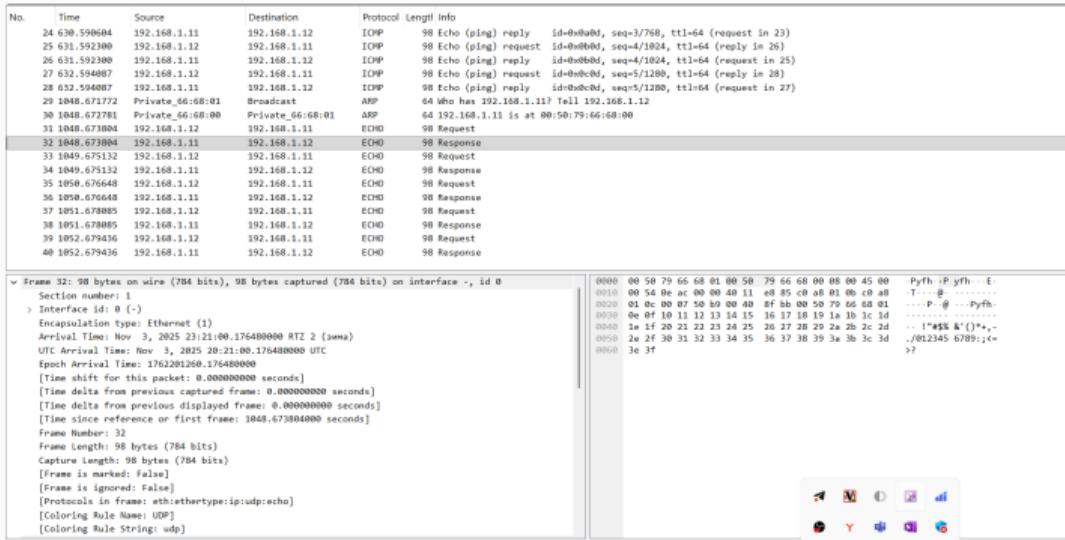


Рис. 24: Эхо-ответ в UDP-моде

Эхо-ответ в UDP-моде

No.	Time	Source	Destination	Protocol	Length	Info
26	630.590684	192.168.1.11	192.168.1.12	ICMP	98	Echo (ping) reply id=0xa0d, seq=5/768, ttl=64 (request in 23)
25	631.592308	192.168.1.12	192.168.1.11	ICMP	98	Echo (ping) request id=0xb0d, seq=4/1024, ttl=64 (reply in 26)
26	631.592308	192.168.1.11	192.168.1.12	ICMP	98	Echo (ping) reply id=0xb0d, seq=4/1024, ttl=64 (request in 25)
27	632.594087	192.168.1.12	192.168.1.11	ICMP	98	Echo (ping) request id=0xc0d, seq=5/1280, ttl=64 (reply in 28)
28	632.594087	192.168.1.11	192.168.1.12	ICMP	98	Echo (ping) reply id=0xc0d, seq=5/1280, ttl=64 (request in 27)
29	1648.671772	Private_66:68:01	Broadcast	ARP	64	Who has 192.168.1.1? Tell 192.168.1.20
30	1648.672781	Private_66:68:00	Private_66:68:00	ARP	64	192.168.1.11 is at 00:50:79:66:68:00
31	1648.673804	192.168.1.12	192.168.1.11	ECHO	98	Request
32	1648.673804	192.168.1.11	192.168.1.12	ECHO	98	Response
33	1649.675132	192.168.1.12	192.168.1.11	ECHO	98	Request
34	1649.675132	192.168.1.11	192.168.1.12	ECHO	98	Response
35	1650.676648	192.168.1.12	192.168.1.11	ECHO	98	Request
36	1650.676648	192.168.1.11	192.168.1.12	ECHO	98	Response
37	1651.678885	192.168.1.12	192.168.1.11	ECHO	98	Request
38	1651.678885	192.168.1.11	192.168.1.12	ECHO	98	Response
39	1652.679436	192.168.1.12	192.168.1.11	ECHO	98	Request
40	1652.679436	192.168.1.11	192.168.1.12	ECHO	98	Response

Type: IPv4 (0x0800)
[Stream index: 2]
v Internet Protocol Version 4, Src: 192.168.1.11, Dst: 192.168.1.12
0100 = Version: 4
.... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCH: CS0, ECN: Not-ECT)
Total Length: 84
Identification: 0x0eac (3756)
> 0800 = Flags: 0x0
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 64
Protocol: UDP [17]
Header checksum: 0xed85 [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.1.11
Destination Address: 192.168.1.12
[Stream index: 8]
> User Datagram Protocol, Src Port: 7, Dst Port: 20665
> Echo

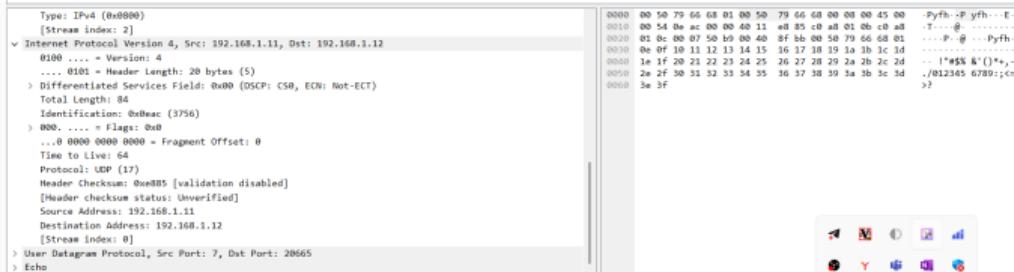


Рис. 25: Эхо-ответ в UDP-моде

Сделаем один эхо-запрос в TCP-моде к узлу PC-1.

```
PC2> ping 192.168.1.11 -1
84 bytes from 192.168.1.11 icmp_seq=1 ttl=64 time=0.445 ms
84 bytes from 192.168.1.11 icmp_seq=2 ttl=64 time=0.657 ms
84 bytes from 192.168.1.11 icmp_seq=3 ttl=64 time=0.694 ms
84 bytes from 192.168.1.11 icmp_seq=4 ttl=64 time=0.688 ms
84 bytes from 192.168.1.11 icmp_seq=5 ttl=64 time=0.655 ms

PC2> ping 192.168.1.11 -2
84 bytes from 192.168.1.11 udp_seq=1 ttl=64 time=0.527 ms
84 bytes from 192.168.1.11 udp_seq=2 ttl=64 time=0.610 ms
84 bytes from 192.168.1.11 udp_seq=3 ttl=64 time=0.655 ms
84 bytes from 192.168.1.11 udp_seq=4 ttl=64 time=0.659 ms
84 bytes from 192.168.1.11 udp_seq=5 ttl=64 time=0.876 ms

PC2> ping 192.168.1.11 -3
Connect 7@192.168.1.11 seq=1 ttl=64 time=1.089 ms
SendData 7@192.168.1.11 seq=1 ttl=64 time=1.051 ms
Close    7@192.168.1.11 seq=1 ttl=64 time=3.116 ms
Connect 7@192.168.1.11 seq=2 ttl=64 time=1.509 ms
SendData 7@192.168.1.11 seq=2 ttl=64 time=1.903 ms
Close    7@192.168.1.11 seq=2 ttl=64 time=2.434 ms
Connect 7@192.168.1.11 seq=3 ttl=64 time=1.342 ms
SendData 7@192.168.1.11 seq=3 ttl=64 time=1.102 ms
Close    7@192.168.1.11 seq=3 ttl=64 time=2.341 ms
Connect 7@192.168.1.11 seq=4 ttl=64 time=2.213 ms
SendData 7@192.168.1.11 seq=4 ttl=64 time=1.072 ms
Close    7@192.168.1.11 seq=4 ttl=64 time=3.386 ms
Connect 7@192.168.1.11 seq=5 ttl=64 time=1.020 ms
SendData 7@192.168.1.11 seq=5 ttl=64 time=1.987 ms
Close    7@192.168.1.11 seq=5 ttl=64 time=2.039 ms

PC2> █
```

Эхо-запрос в TCP-моде

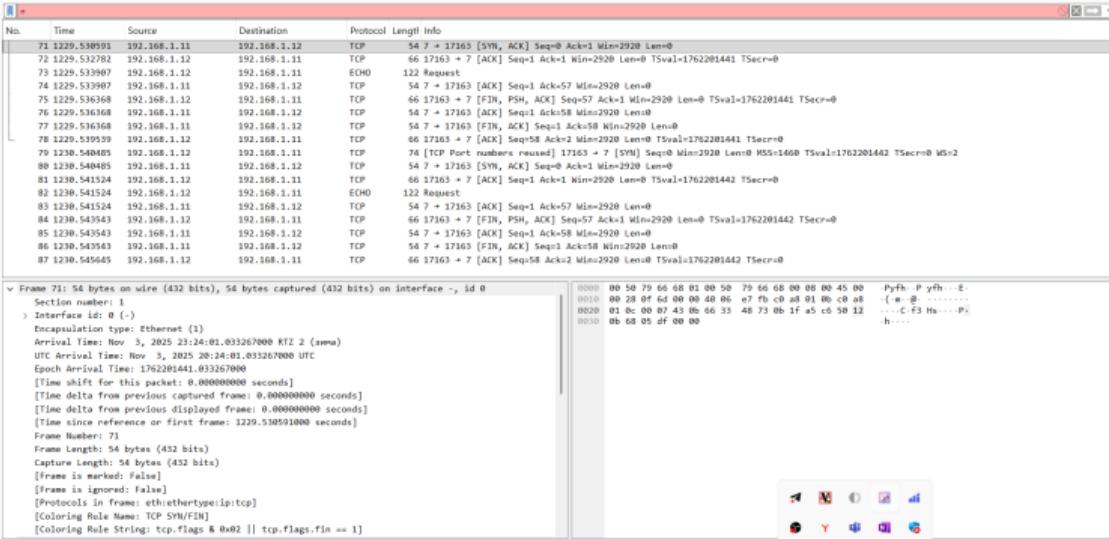


Рис. 28: TCP-моде

Эхо-запрос в TCP-моде

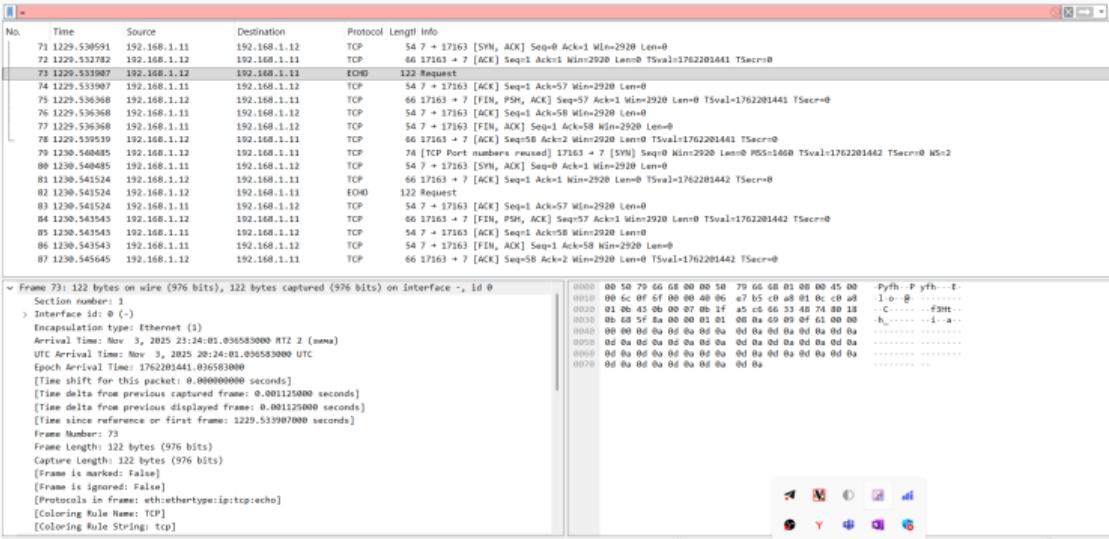


Рис. 29: TCP-моде

Эхо-запрос в TCP-моде

Рис. 30: TCP-моде

Моделирование простейшей сети на базе маршрутизатора FRR в GNS3

Запустим GNS3 VM и GNS3. Создадим новый проект.

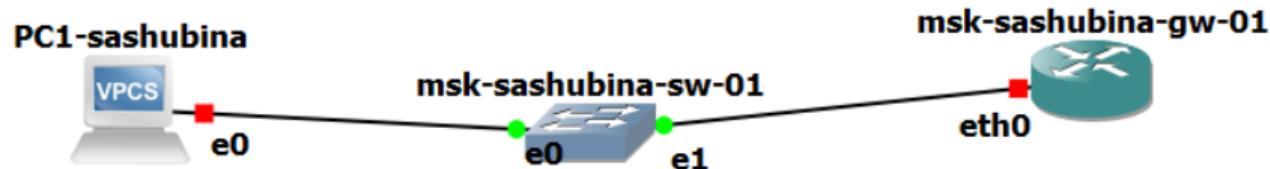


Рис. 31: Создание проекта

Настроим IP-адресацию для интерфейса узла PC1:

```
VPCS> ip 192.168.1.10/24 192.168.1.1
Checking for duplicate address...
PC1 : 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     : 10003
RHOST:PORT: 127.0.0.1:10004
MTU:      : 1500

VPCS> █
```

Рис. 32: Настройка IP-адресации для интерфейса узла PC-1

Настроим IP-адресацию для интерфейса локальной сети маршрутизатора:

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

Рис. 33: Настройка IP-адресации для интерфейса локальной сети маршрутизатора

Проверим конфигурацию маршрутизатора и настройки IP-адресации:

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

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

Рис. 34: Проверка конфигурации маршрутизатора и настройки IP-адресации

Проверим подключение. Узел PC1 успешно отправляет эхо-запросы ICMP на адрес маршрутизатора 192.168.1.1.

```
VPCS> ping 192.168.1.1
84 bytes from 192.168.1.1 icmp_seq=1 ttl=64 time=3.044 ms
84 bytes from 192.168.1.1 icmp_seq=2 ttl=64 time=1.169 ms
84 bytes from 192.168.1.1 icmp_seq=3 ttl=64 time=0.948 ms
84 bytes from 192.168.1.1 icmp_seq=4 ttl=64 time=1.026 ms
84 bytes from 192.168.1.1 icmp_seq=5 ttl=64 time=0.957 ms

VPCS> □
```

Рис. 35: Проверка подключения

В окне Wireshark проанализируем полученную информацию.

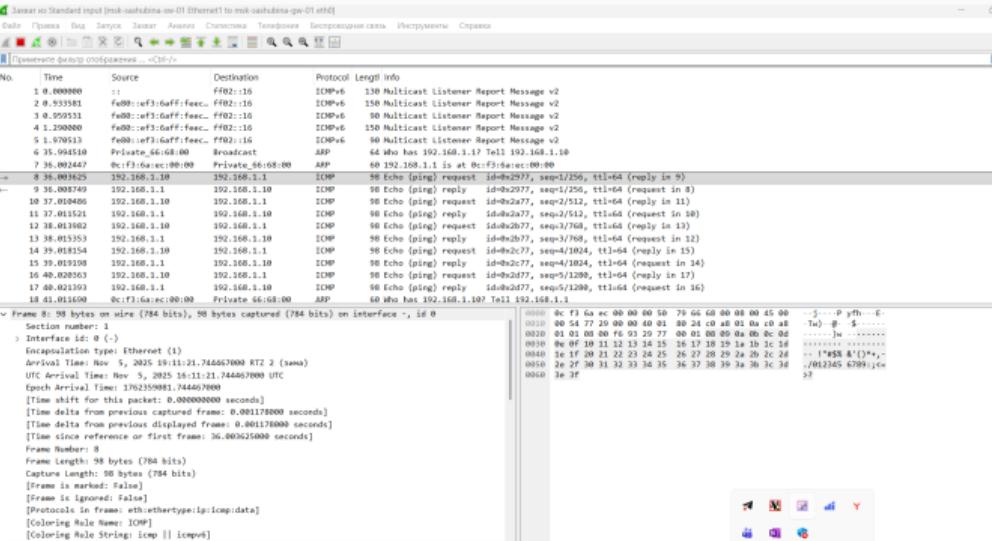


Рис. 36: Эхо-запрос

Эхо-запрос

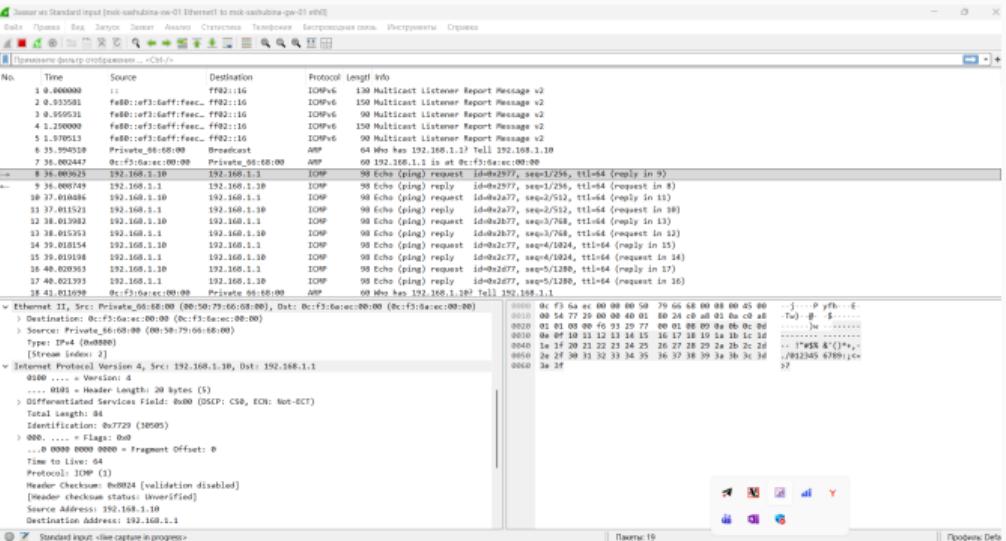


Рис. 37: Эхо-запрос

Эхо-запрос

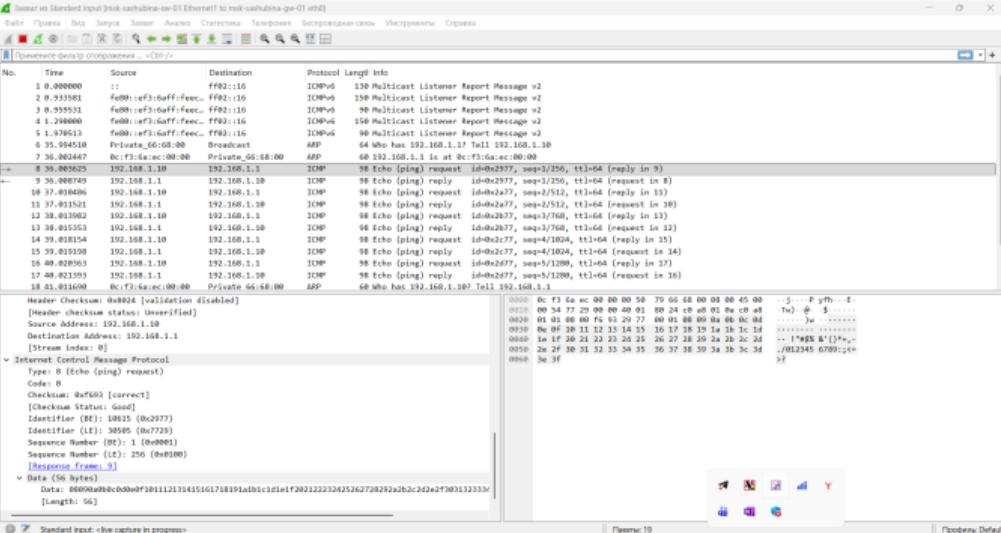


Рис. 38: Эхо-запрос

Эхо-ответ

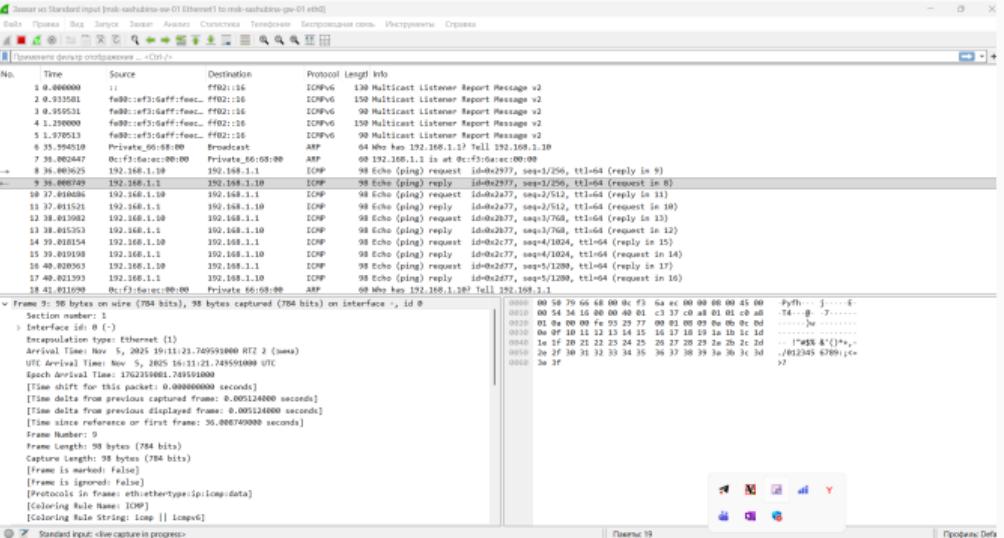


Рис. 39: Эхо-ответ

Эхо-ответ

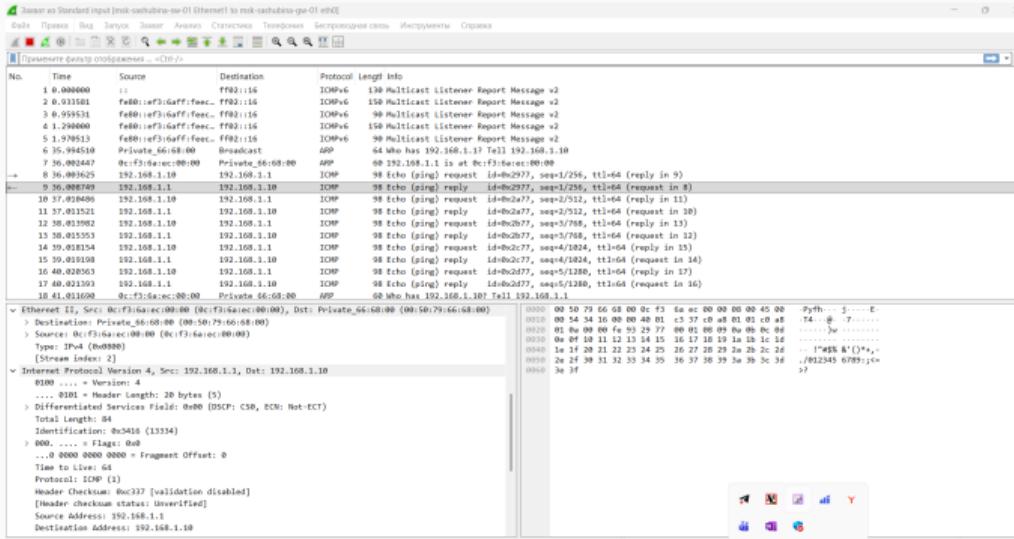


Рис. 40: Эхо-ответ

Эхо-ответ

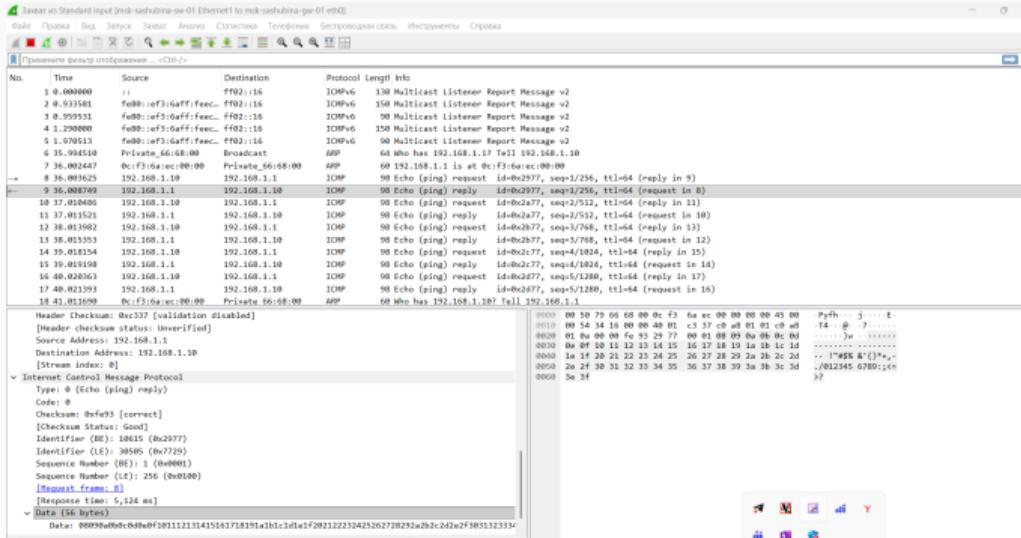


Рис. 41: Эхо-ответ

Моделирование простейшей сети на базе маршрутизатора VyOS в GNS3

Запустим GNS3 VM и GNS3. Создадим новый проект. В рабочей области GNS3 разместим VPCS, коммутатор Ethernet и маршрутизатор VyOS.

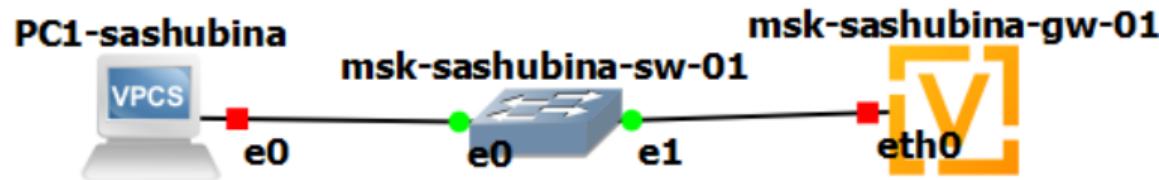


Рис. 42: Создание проекта

Настроим IP-адресацию для интерфейса узла PC1.

```
VPCS> ip 192.168.1.10/24 192.168.1.1
Checking for duplicate address...
PC1 : 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     : 10003
RHOST:PORT: 127.0.0.1:10004
MTU:      : 1500
```

Рис. 43: Настройка IP-адресации для интерфейса узла PC-1

Настроим маршрутизатор VyOS. После загрузки введем логин vyos и пароль vyos: В рабочем режиме в командной строке отображается символ \$.

```
vyos login: vyos
Password:
Linux vyos 4.19.195-amd64-vyos #1 SMP Sat Jun 19 08:48:00 UTC 2021 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
vyos@vyos:~$
```

Рис. 44: ввод логина и пароля

Установим систему на диск:

```
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...
```

Рис. 45: система установлена

Перезагрузим.

```
vyos@vyos:~$ reboot
Are you sure you want to reboot this system? [y/N] y
[ 436.991780] reboot: Restarting system

Welcome to VyOS - vyos ttyS0
```

Рис. 46: перезагрузка

Для настройки маршрутизатора VyOS необходимо выполнить последовательность команд.

```
vyos@vyos:~$ configure
[edit]
vyos@vyos# set system host-name msk-sashubina-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-sashubina-gw-01
[edit]
vyos@vyos# commit
[edit]
vyos@vyos# save
Saving configuration to '/config/config.boot'...
Done
[edit]
```

Рис. 47: настройка

Для проверки настроек интерфейсов выполняем `show interfaces`. Завершаем работу в режиме конфигурирования командой `exit`, возвращаясь в обычный режим с приглашением `$`. Как мы видим, `eth0` успешно настроен

```
vyos@vyos# show interfaces
ethernet eth0 {
    address 192.168.1.1/24
    hw-id 0c:fa:53:ea:00:00
}
ethernet eth1 {
    hw-id 0c:fa:53:ea:00:01
}
ethernet eth2 {
    hw-id 0c:fa:53:ea:00:02
}
ethernet eth3 {
    hw-id 0c:fa:53:ea:00:03
}
ethernet eth4 {
    hw-id 0c:fa:53:ea:00:04
}
ethernet eth5 {
    hw-id 0c:fa:53:ea:00:05
}
ethernet eth6 {
    hw-id 0c:fa:53:ea:00:06
}
ethernet eth7 {
    hw-id 0c:fa:53:ea:00:07
}
ethernet eth8 {
    hw-id 0c:fa:53:ea:00:08
}
ethernet eth9 {
    hw-id 0c:fa:53:ea:00:09
}
loopback lo {
}
[edit]
vyos@vyos#
```

Рис. 48: проверка настроек

Проверим подключение. Узел PC1 успешно отправлять эхо-запросы на адрес маршрутизатора 192.168.1.1.

```
VPCS> ping 192.168.1.1
84 bytes from 192.168.1.1 icmp_seq=1 ttl=64 time=0.866 ms
84 bytes from 192.168.1.1 icmp_seq=2 ttl=64 time=5.337 ms
84 bytes from 192.168.1.1 icmp_seq=3 ttl=64 time=0.895 ms
84 bytes from 192.168.1.1 icmp_seq=4 ttl=64 time=1.060 ms
84 bytes from 192.168.1.1 icmp_seq=5 ttl=64 time=0.940 ms
```

```
VPCS> █
```

Рис. 49: Проверка соединения

В окне Wireshark проанализируем полученную информацию.

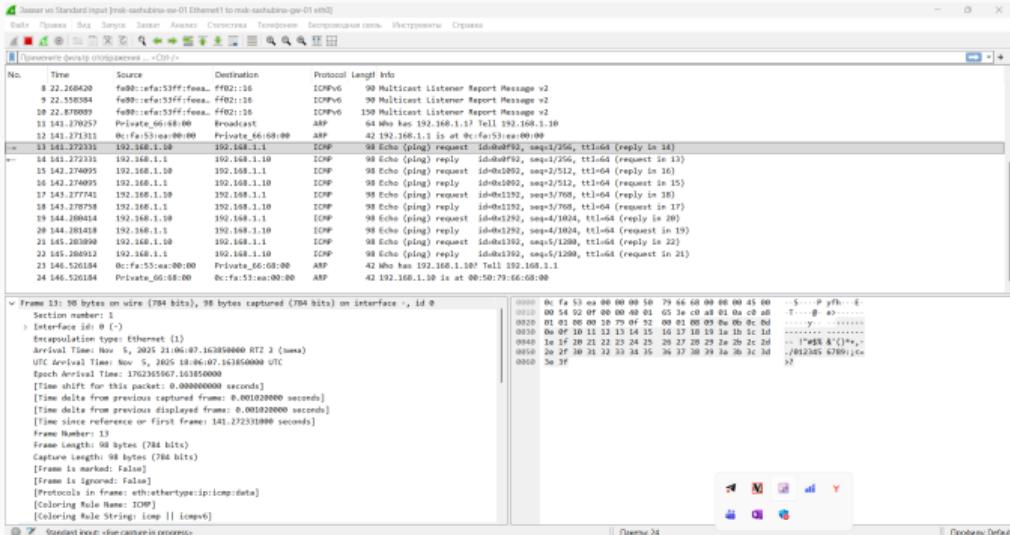


Рис. 50: Анализ трафика Wireshark

Анализ трафика Wireshark

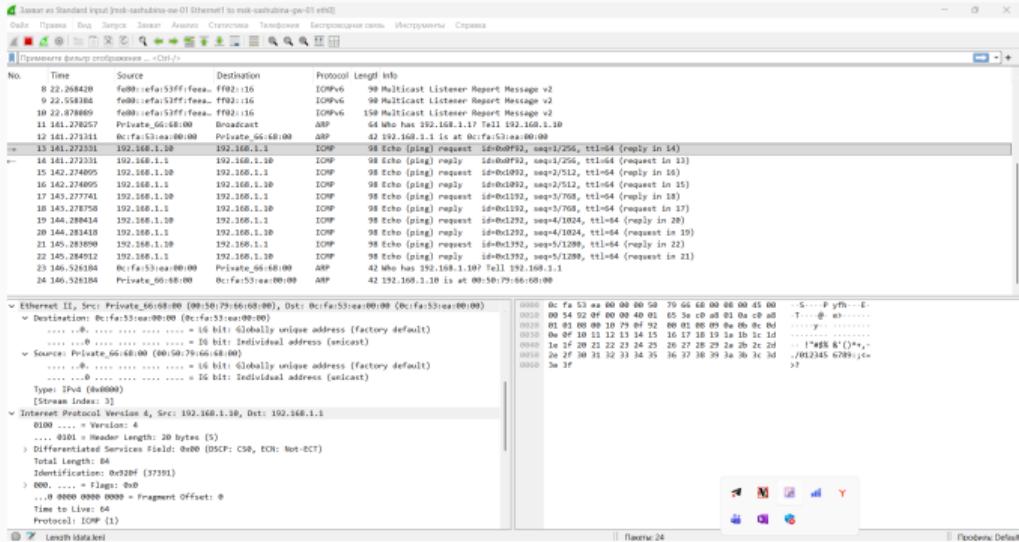


Рис. 51: Анализ трафика Wireshark

Анализ трафика Wireshark

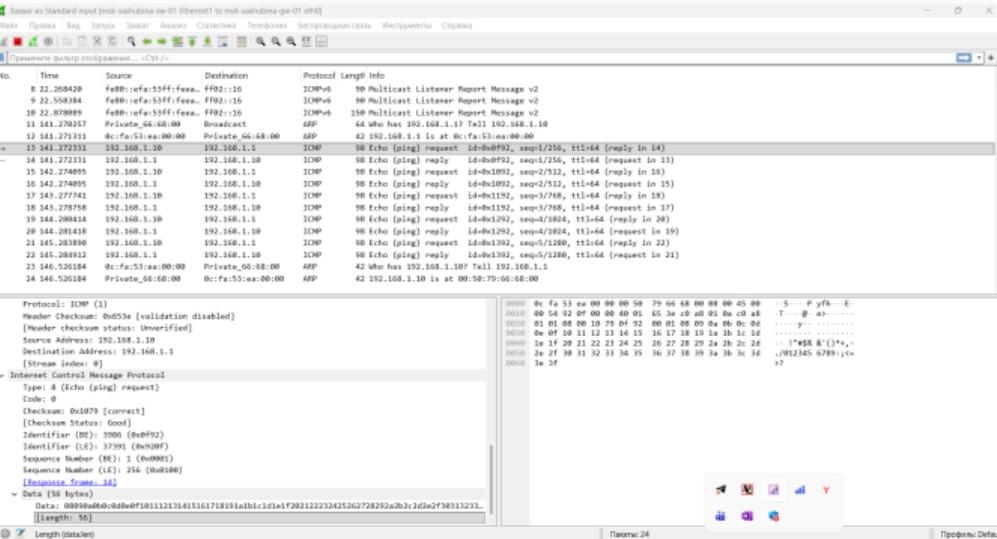


Рис. 52: Анализ трафика Wireshark

Анализ трафика Wireshark

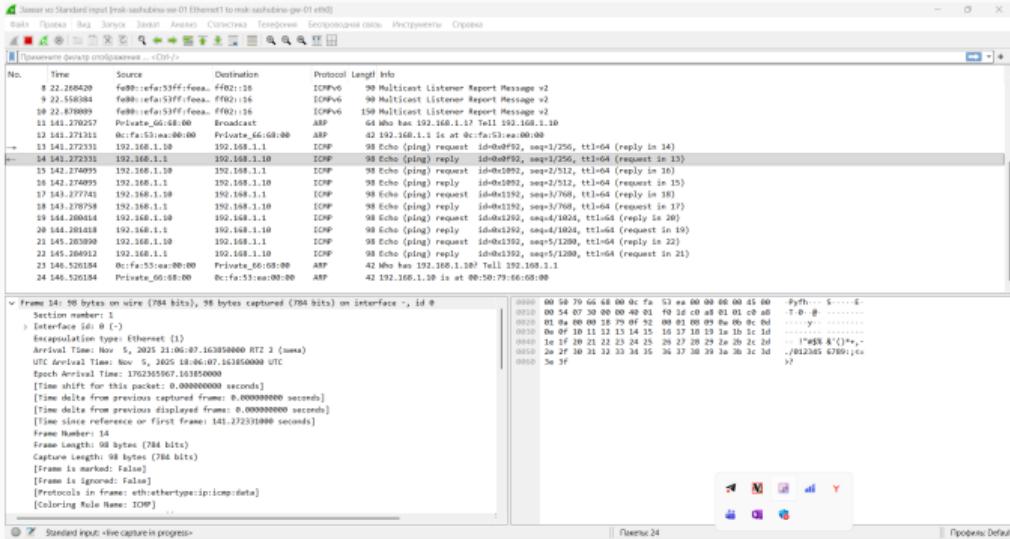


Рис. 53: Анализ трафика Wireshark

Анализ трафика Wireshark

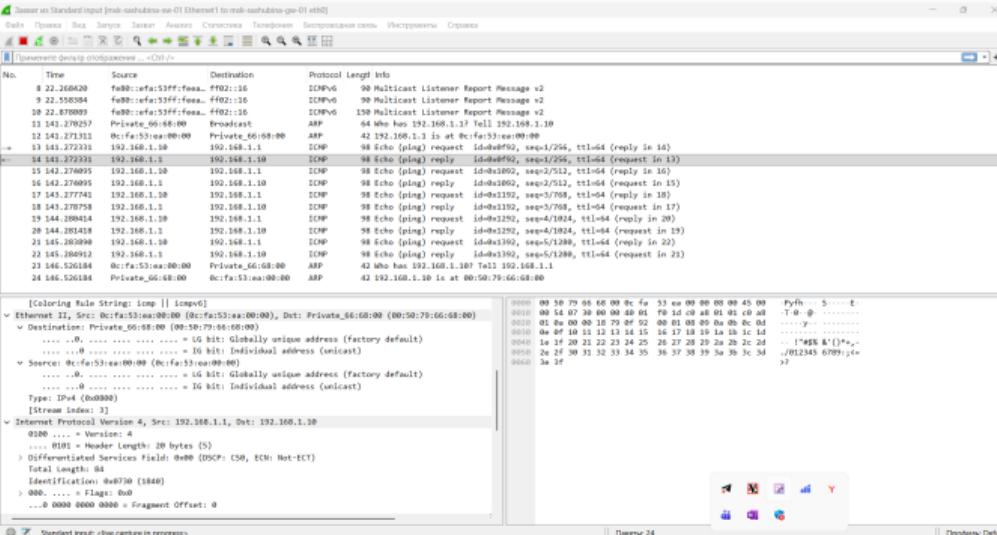


Рис. 54: Анализ трафика Wireshark

Анализ трафика Wireshark

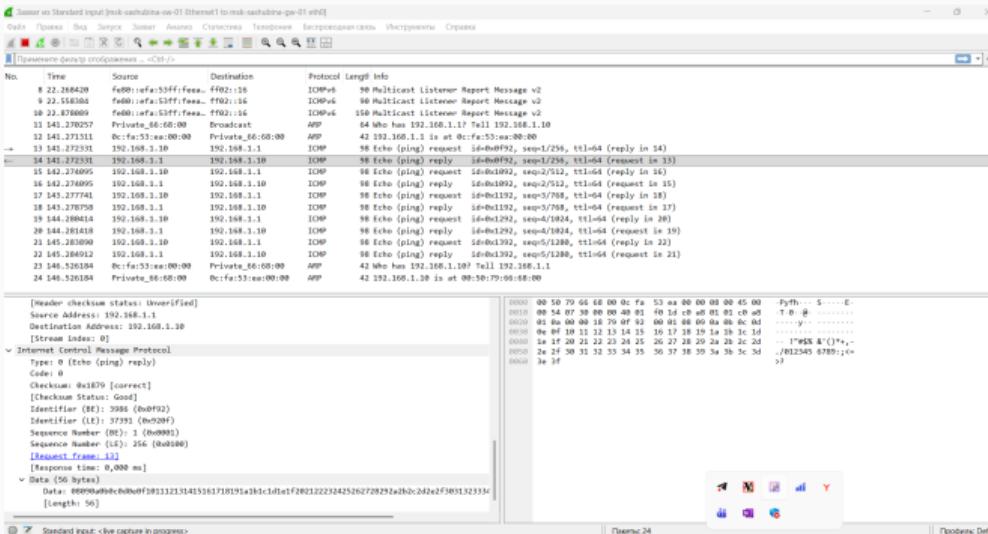


Рис. 55: Анализ трафика Wireshark

Выводы

В результате выполнения лабораторной работы были построены простейшие модели сети на базе коммутатора и маршрутизаторов FRR и VyOS в GNS3, а также проанализирован трафик посредством Wireshark.