

Лабораторная работа №5

Простые сети в GNS3. Анализ трафика

Шубина София Антоновна

Содержание

1 Цель работы	5
2 Задание	6
3 Выполнение лабораторной работы	7
3.1 Моделирование простейшей сети на базе коммутатора в GNS3	7
3.2 Анализ трафика в GNS3 посредством Wireshark	13
3.3 Моделирование простейшей сети на базе маршрутизатора FRR в GNS3	22
3.4 Моделирование простейшей сети на базе маршрутизатора VyOS в GNS3	29
4 Выводы	36

Список иллюстраций

3.1 Добавление устройств	8
3.2 изменение имен	8
3.3 изменение имен	9
3.4 изменение имен	9
3.5 соединение	10
3.6 Параметры импорта	10
3.7 Задание IP-адреса PC1-sashubina	11
3.8 show ip	11
3.9 Задание IP-адреса PC2-sashubina	12
3.10 Проверка соединения между PC-1 и PC-2	12
3.11 Проверка соединения между PC-1 и PC-2	12
3.12 Остановка всех узлов	13
3.13 Запуск анализатора трафика	13
3.14 Запуск анализатора трафика	13
3.15 ARP пакеты	14
3.16 ARP пакеты	14
3.17 ARP пакеты	14
3.18 ARP пакеты	15
3.19 опции команды ping	16
3.20 Эхо-ответ в ICMP-моде	17
3.21 Эхо-ответ в ICMP-моде	17
3.22 Эхо-запрос в ICMP-моде	17
3.23 Эхо-запрос в ICMP-моде	18
3.24 Эхо-запрос в UDP-моде	18
3.25 Эхо-запрос в UDP-моде	19
3.26 Эхо-ответ в UDP-моде	19
3.27 Эхо-ответ в UDP-моде	19
3.28 ping	20
3.29 TCP-моде	20
3.30 TCP-моде	21
3.31 TCP-моде	21
3.32 TCP-моде	21
3.33 Создание проекта	23
3.34 Настройка IP-адресации для интерфейса узла PC-1	23
3.35 Настройка IP-адресации для интерфейса локальной сети маршрутизатора	24
3.36 Проверка конфигурации маршрутизатора и настройки IP-адресации	25

3.37 Проверка подключения	26
3.38 Эхо-запрос	27
3.39 Эхо-запрос	27
3.40 Эхо-запрос	27
3.41 Эхо-ответ	28
3.42 Эхо-ответ	28
3.43 Эхо-ответ	28
3.44 Создание проекта	29
3.45 Настройка IP-адресации для интерфейса узла PC-1	30
3.46 ввод логина и пароля	30
3.47 система установлена	30
3.48 перезагрузка	31
3.49 настройка	31
3.50 проверка настроек	32
3.51 Проверка соединения	32
3.52 Анализ трафика Wireshark	33
3.53 Анализ трафика Wireshark	33
3.54 Анализ трафика Wireshark	34
3.55 Анализ трафика Wireshark	34
3.56 Анализ трафика Wireshark	34
3.57 Анализ трафика Wireshark	35

1 Цель работы

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

2 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

3 Выполнение лабораторной работы

3.1 Моделирование простейшей сети на базе коммутатора в GNS3

1. Построить в GNS3 топологию сети, состоящей из коммутатора Ethernet и двух оконечных устройств (персональных компьютеров).
2. Задать оконечным устройствам IP-адреса в сети 192.168.1.0/24. Проверить связь.

Запустим GNS3 VM и GNS3 и создадим новый проект. В рабочей области GNS3 разместим коммутатор Ethernet и два VPCS. Щёлкнув на устройстве правой кнопкой мыши в меню Configure изменим название устройства, включив в имя устройства имя своей учётной записи. Коммутатору присвоим название msk-sashubina-sw-01. Затем соединим VPCS с коммутатором и отобразим обозначение интерфейсов соединения.

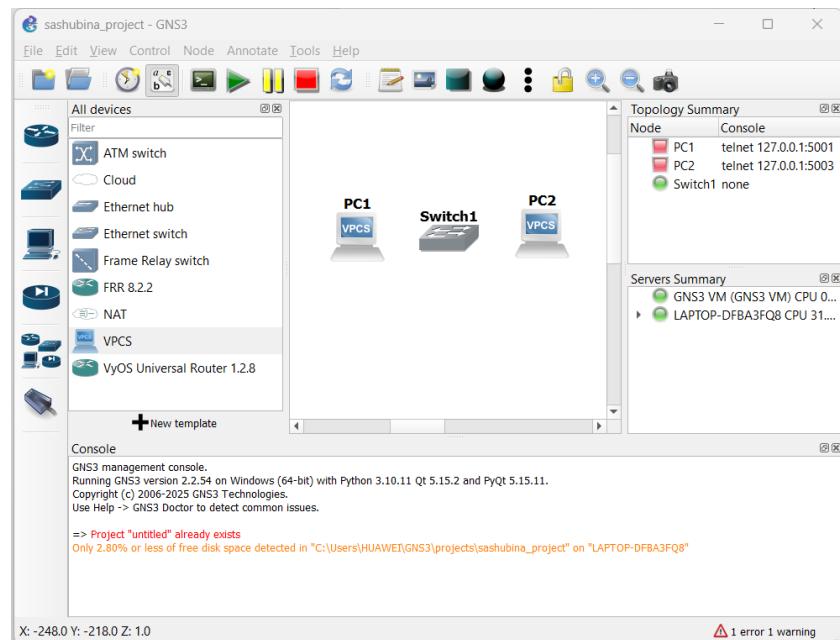


Рис. 3.1: Добавление устройств

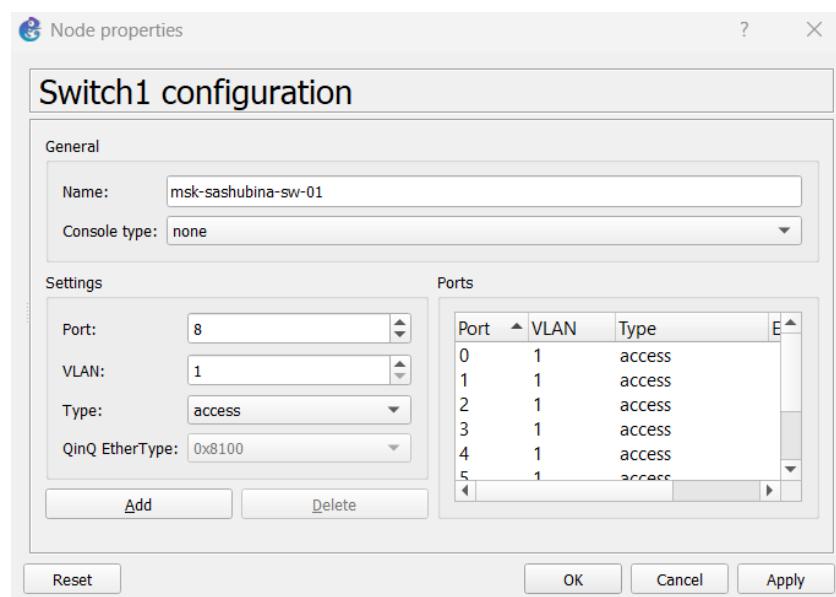


Рис. 3.2: изменение имен

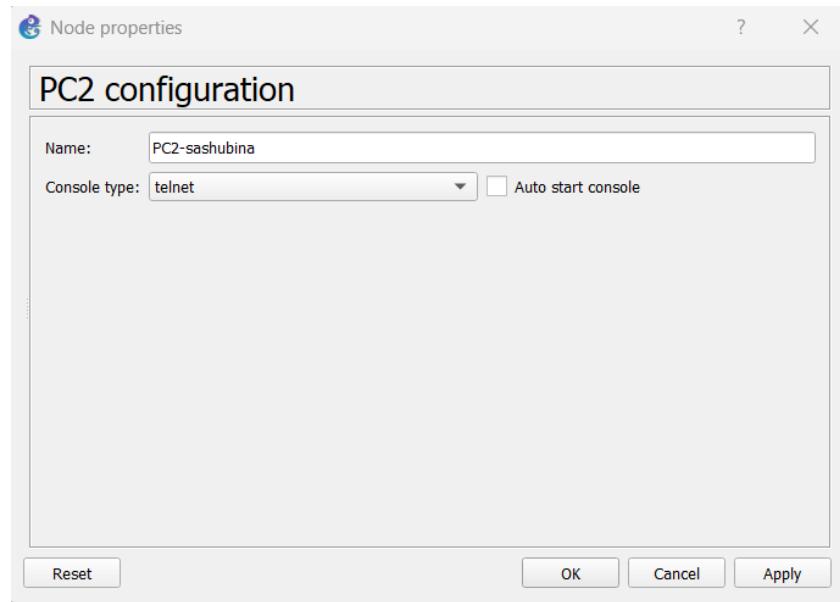


Рис. 3.3: изменение имен

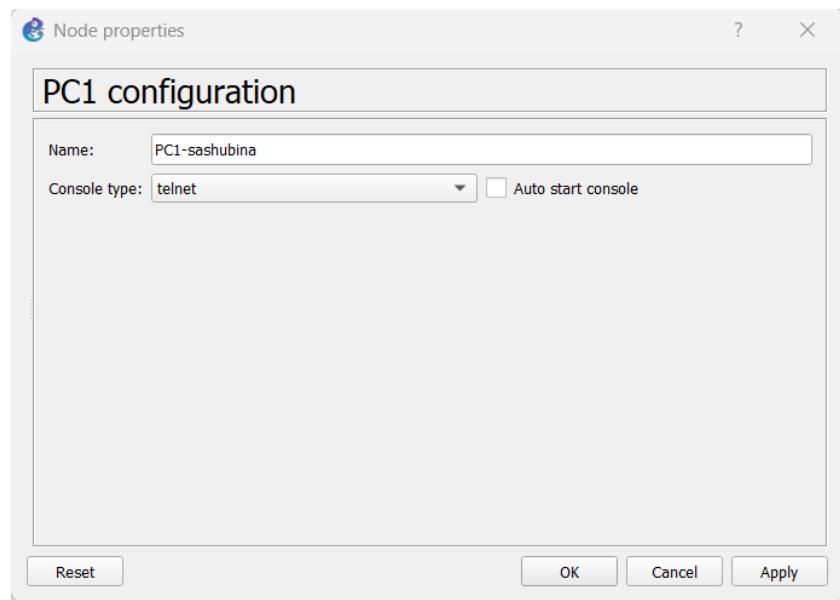


Рис. 3.4: изменение имен

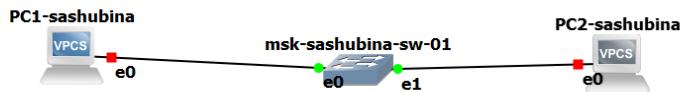


Рис. 3.5: соединение

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 Daling.
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> /?

?
! COMMAND [ARG ...]      Print help
arp                         Invoke an OS COMMAND with optional ARG(s)
clear                        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)
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. 3.6: Параметры импорта

Ключевые команды эмулятора Virtual PC Simulator и их назначение включают в себя: команда ip является самой важной, так как позволяет настраивать IP-адрес, маску сети и шлюз для виртуального компьютера. Команда ping используется для проверки сетевой связности с другим хостом по IP-адресу или имени. С помощью команды trace можно выполнить трассировку маршрута до

указанного хоста (это аналог команды tracert в Windows). Команда dhcp обеспечивает автоматическое получение IP-адреса от DHCP-сервера. Для просмотра различной информации, такой как текущие настройки IP, таблица ARP или история команд, используется команда show. Установка параметров, например имени виртуального ПК (hostname), выполняется командой set. Команда arp показывает ARP-таблицу, то есть соответствие IP-адресов MAC-адресам в локальной сети. Сохранение текущей конфигурации в файл или её загрузка из файла осуществляется командами save и load соответственно. Команда clear позволяет очистить настройки IP, кэш ARP или историю команд. Для выхода из программы используется команда quit.

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

```
ip 192.168.1.11/24 192.168.1.1
```

А для сохранения конфигураций введём команду save.

```
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. 3.7: 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. 3.8: 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. 3.9: 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. 3.10: Проверка соединения между PC-1 и PC-2

```
PC2> ping 192.168.1.11
84 bytes from 192.168.1.11 icmp_seq=1 ttl=64 time=0.437 ms
84 bytes from 192.168.1.11 icmp_seq=2 ttl=64 time=0.434 ms
84 bytes from 192.168.1.11 icmp_seq=3 ttl=64 time=0.399 ms
84 bytes from 192.168.1.11 icmp_seq=4 ttl=64 time=0.365 ms
84 bytes from 192.168.1.11 icmp_seq=5 ttl=64 time=0.433 ms
```

Рис. 3.11: Проверка соединения между PC-1 и PC-2

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

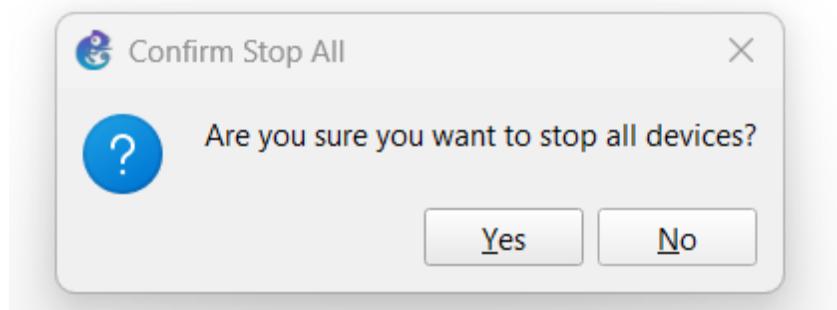


Рис. 3.12: Остановка всех узлов

3.2 Анализ трафика в GNS3 посредством Wireshark

1. С помощью Wireshark захватить и проанализировать ARP-сообщения.
2. С помощью Wireshark захватить и проанализировать ICMP-сообщения.

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

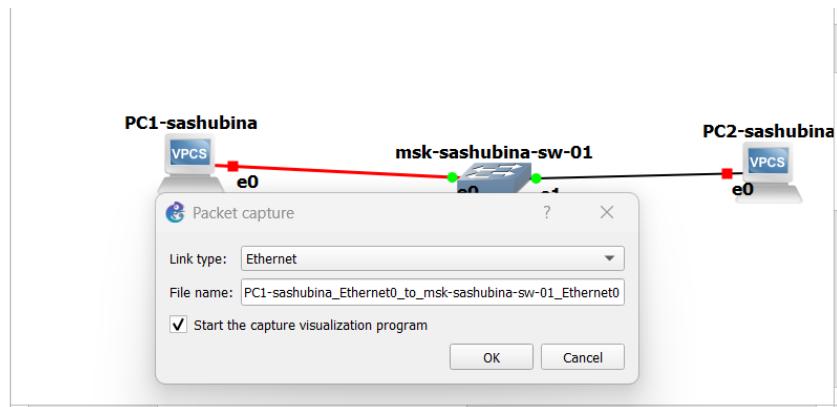


Рис. 3.13: Запуск анализатора трафика

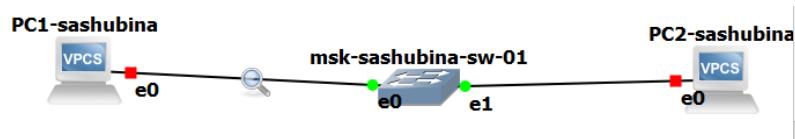


Рис. 3.14: Запуск анализатора трафика

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

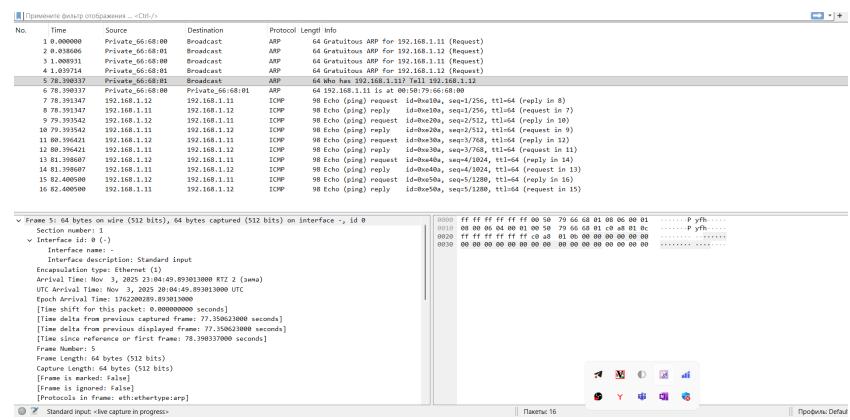


Рис. 3.15: ARP пакеты

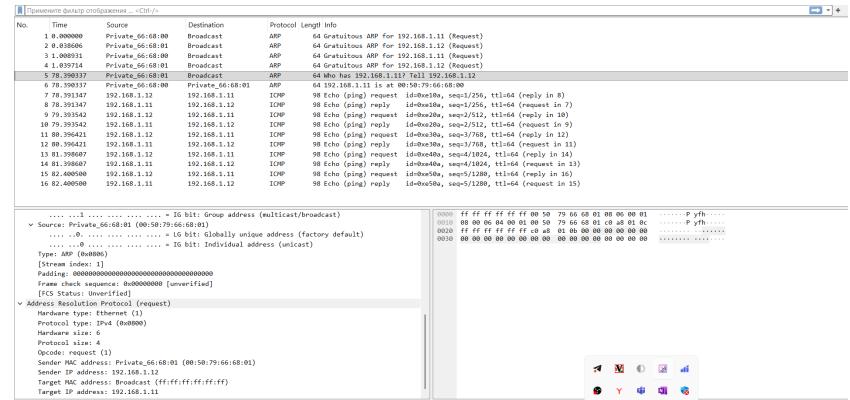


Рис. 3.16: ARP пакеты

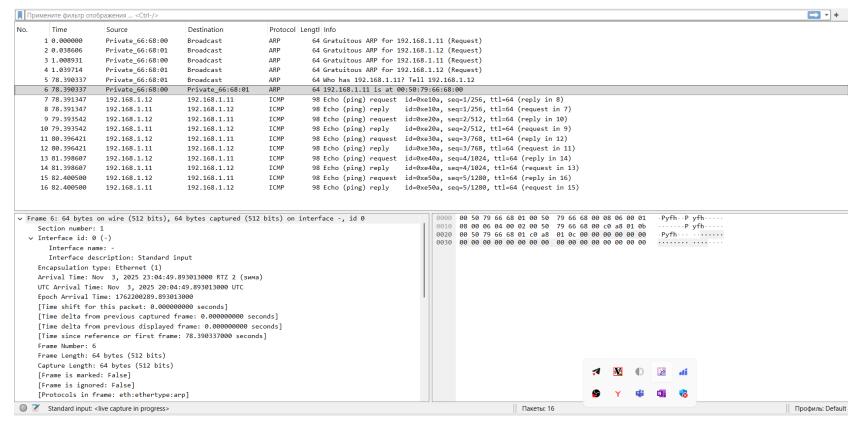


Рис. 3.17: ARP пакеты

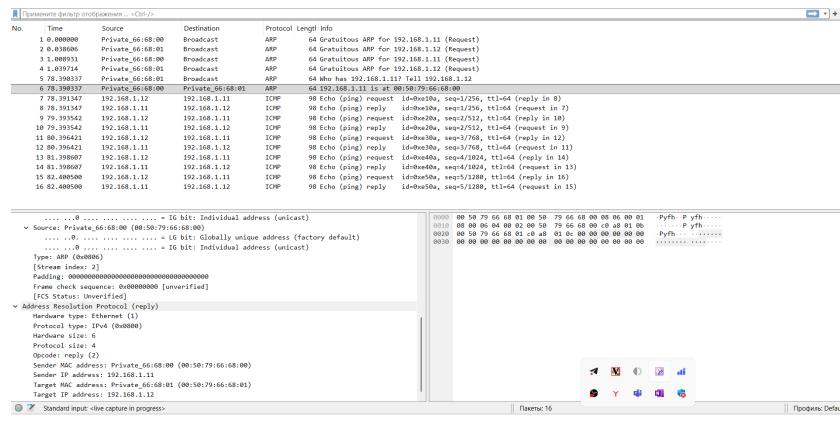


Рис. 3.18: ARP пакеты

В проекте GNS3 стартуем все узлы (меню GNS3 Control Start/Resume all nodes).

В окне Wireshark отобразилась информация по протоколу ARP.

Изучим запрос и ответ ARP в программе Wireshark. В обоих случаях длина кадра равняется 64 байт. В начале сформировались запросы безвоздмездных пакетов ARP для PC-1(в этом случае источник – Private_66:68:00, а пункт назначения - Broadcast) и для PC-2(в этом случае источник – Private_66:68:01, а пункт назначения - Broadcast). Затем был сформирован запрос от PC-2 на передачу MAC-адреса PC-1 и получен ответ - MAC-адрес.

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|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> 

```

Рис. 3.19: опции команды ping

Основной синтаксис команды — ping HOST [OPTION ...], где HOST может быть IP-адресом или именем узла. Ключевой особенностью этой утилиты является поддержка различных режимов работы, которые задаются с помощью опций: по умолчанию используется ICMP-режим (-1), но также доступны UDP (-2) и TCP (-3) режимы, что позволяет проводить более гибкое тестирование доступности сетевых служб. Для управления количеством отправляемых пакетов служит опция -c count (по умолчанию 5), а опция -t позволяет отправлять пакеты непрерывно до принудительной остановки пользователем комбинацией клавиш Ctrl+C. Тонкая настройка передаваемых пакетов осуществляется с помощью дополнительных параметров: -l size задаёт размер данных, -T ttl устанавливает время жизни пакета (TTL), а -D включает бит, запрещающий фрагментацию. Для управления портами предназначены опции -p port (указание порта назначения) и -s port (указание исходного порта). Скорость опроса регулируется параметром -i ms, который задаёт задержку в миллисекундах между отправкой пакетов, а время ожидания ответа контролируется опцией -w ms.

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.

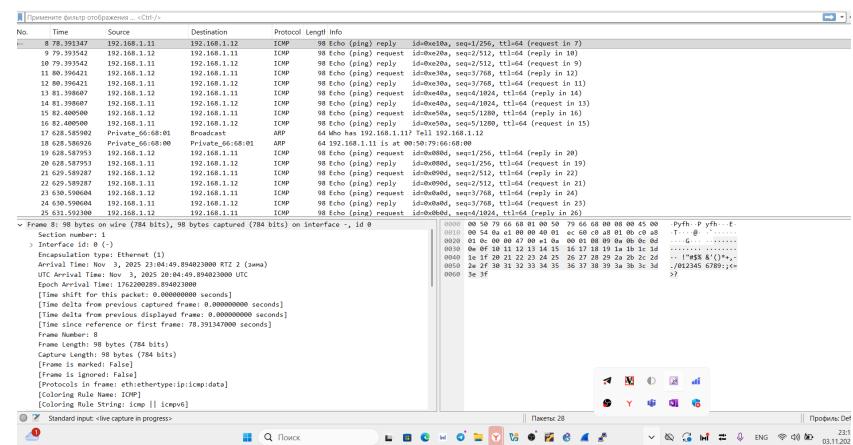


Рис. 3.20: Эхо-ответ в ICMP-моде

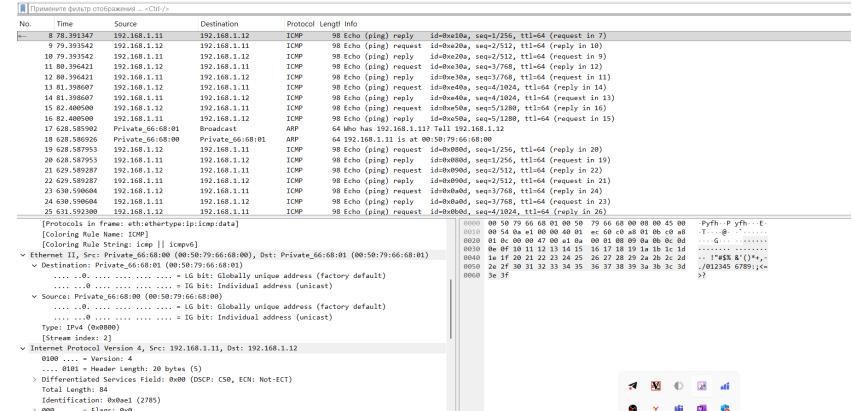


Рис. 3.21: Эхо-ответ в ICMP-моде

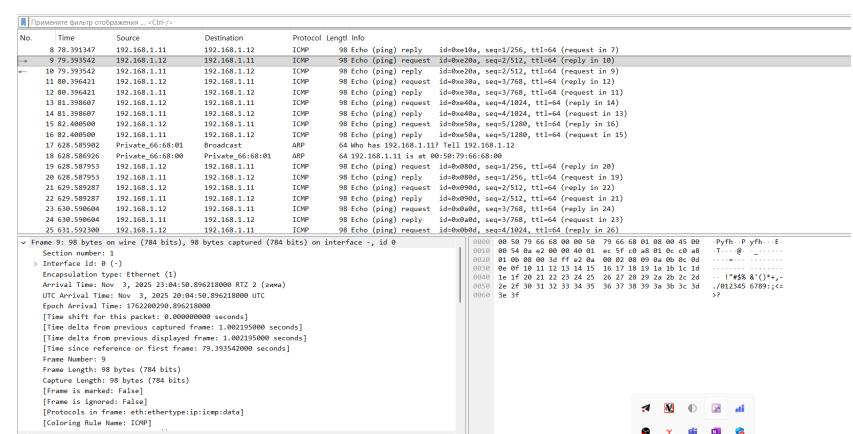


Рис. 3.22: Эхо-запрос в ICMP-моде

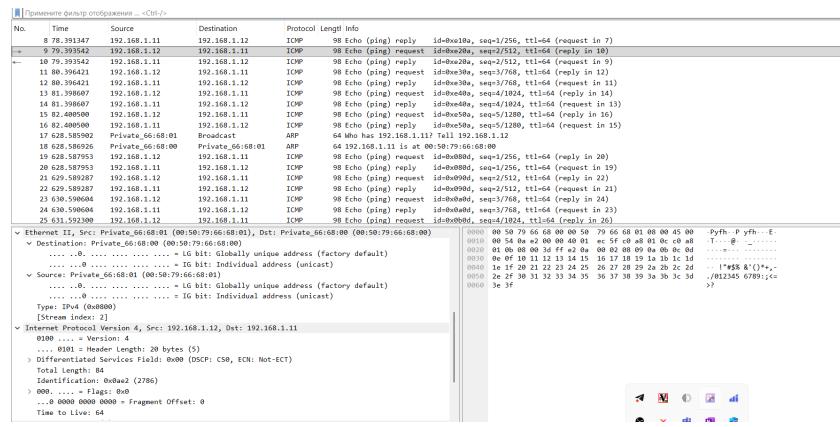


Рис. 3.23: Эхо-запрос в ICMP-моде

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.

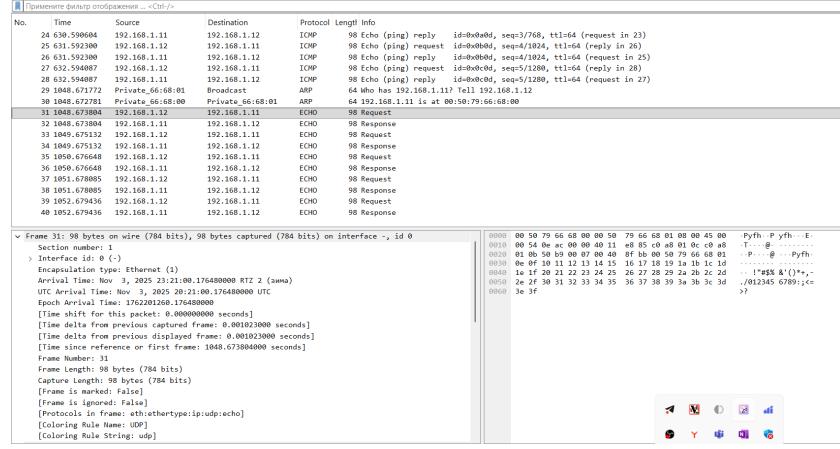


Рис. 3.24: Эхо-запрос в UDP-моде

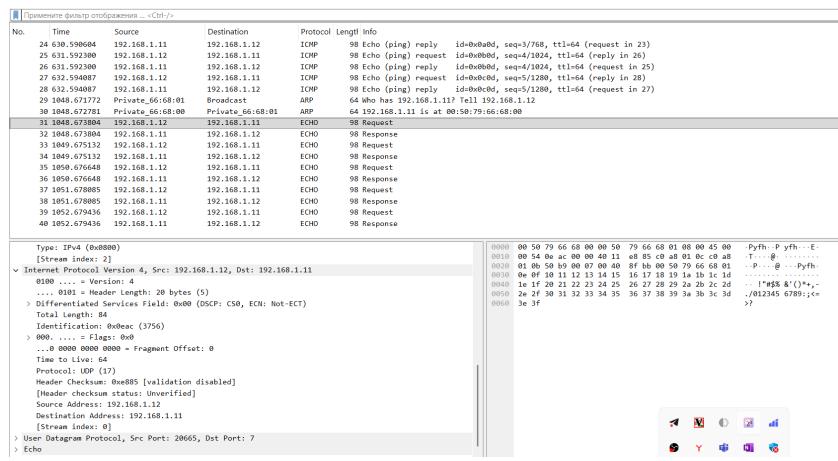


Рис. 3.25: Эхо-запрос в UDP-моде

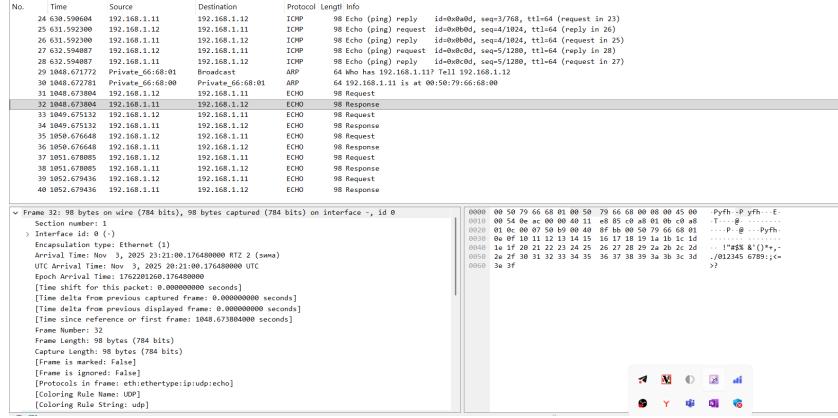


Рис. 3.26: Эхо-ответ в UDP-моде

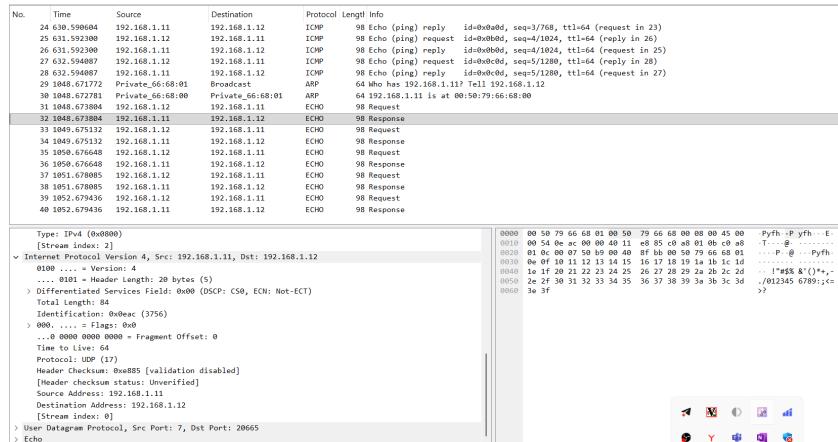


Рис. 3.27: Эхо-ответ в UDP-моде

Сделаем один эхо-запрос в TCP-моде к узлу PC-1. В окне Wireshark проанализируем полученную информацию. Порт источника задан случайно равен 20665, порт назначения равен 7. В случае ответа порты заданы наоборот. Также можно увидеть handshake протокола TCP. В первом пакете установлен бит SYN(Syn: set). Во втором пакете установлены биты SYN и ACK(Syn: set, Acknowledgment: set). А в следующем пакете установлен бит ACK(Acknowledgment: set). Также есть пакеты с битом FIN, завершающим handshake.

```

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> 

```

Рис. 3.28: ping

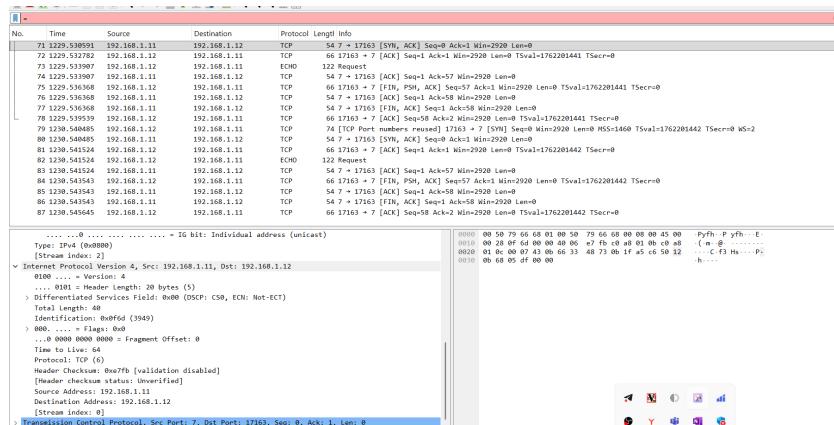


Рис. 3.29: TCP-моде

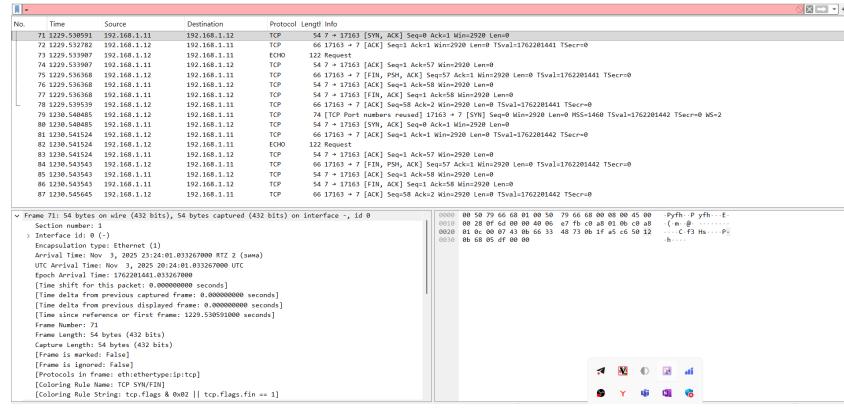


Рис. 3.30: TCP-моде

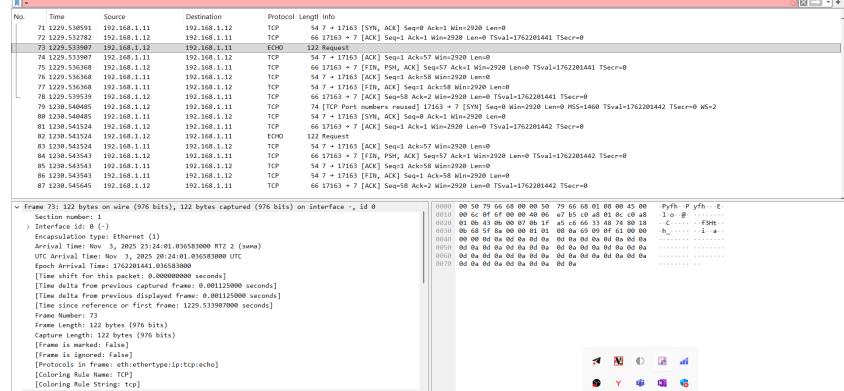


Рис. 3.31: TCP-моде

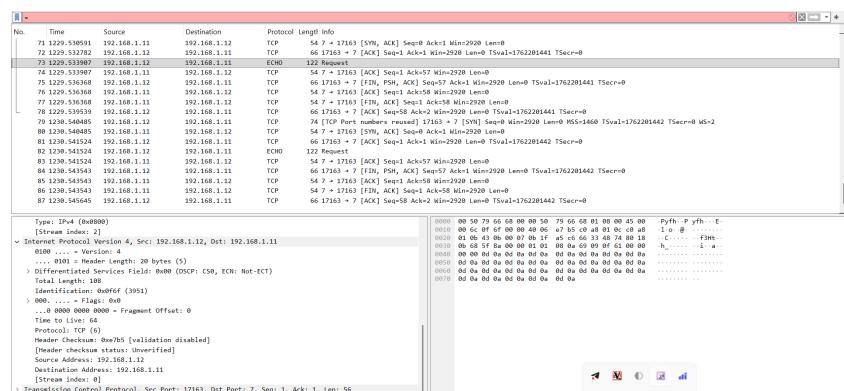


Рис. 3.32: TCP-моде

При выполнении TCP “пинга” наблюдается полный процесс установления TCP-соединения. Анализ в Wireshark показывает: Пакет 71 (SYN-ACK): Ис-

точник: 192.168.1.11 (PC1), порт 7 Назначение: 192.168.1.12 (PC2), порт 17163
Направление: PC1 → PC2 (ответ на запрос соединения)

Пакет 73 (ECHO Request): Источник: 192.168.1.12 (PC2), порт 17163 Назначение: 192.168.1.11 (PC1), порт 7 Направление: PC2 → PC1 (передача данных)

В пакете 71 PC1 отправляет PC2 сегмент с флагами [SYN, ACK] - это ответ на первоначальный запрос соединения. Пакет использует порт 7 (стандартный для echo-службы) и содержит параметры соединения.

Пакет 73 представляет собой ECHO Request, переданный через установленное TCP-соединение. В отличие от ICMP ping, здесь данные передаются поверх транспортного уровня.

Ключевое отличие от ICMP: TCP ping требует предварительного установления соединения (three-way handshake) и работает на транспортном уровне, тогда как ICMP использует сетевой уровень с простой схемой запрос-ответ без установления соединения. Это демонстрирует разницу между connectionless (ICMP) и connection-oriented (TCP) протоколами.

Остановим захват пакетов в Wireshark.

3.3 Моделирование простейшей сети на базе маршрутизатора FRR в GNS3

1. Построить в GNS3 топологию сети, состоящей из маршрутизатора FRR, коммутатора Ethernet и оконечного устройства.
2. Задать оконечному устройству IP-адрес в сети 192.168.1.0/24.
3. Присвоить интерфейсу маршрутизатора адрес 192.168.1.1/24
4. Проверить связь.

Запустим GNS3 VM и GNS3. Создадим новый проект. В рабочей области GNS3 разместим VPCS, коммутатор Ethernet и маршрутизатор FRR. Изменим отображаемые названия устройств. Включим захват трафика на соединении между

коммутатором и маршрутизатором. Затем запустим все устройства проекта. Откроем консоль всех устройств проекта.

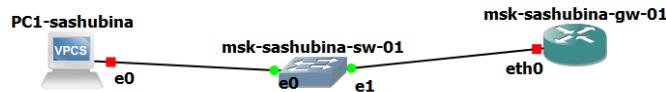


Рис. 3.33: Создание проекта

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. 3.34: 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. 3.35: Настройка IP-адресации для интерфейса локальной сети маршрутизатора

Проверим конфигурацию маршрутизатора и настройки IP-адресации:

```
msk-sashubina-gw-01# show running-config
msk-sashubina-gw-01# show interface brief
```

```

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. 3.36: Проверка конфигурации маршрутизатора и настройки IP-адресации

Конфигурация маршрутизатора FRR успешно применена, что подтверждается выводом команд `show running-config` и `show interface brief`. Маршрутизатор имеет `hostname msk-sashubina-gw-01`, соответствующий требованиям именования. На интерфейсе `eth0` назначен IP-адрес `192.168.1.1/24`, который выступает шлюзом по умолчанию для сети `192.168.1.0/24`. Статус интерфейса `eth0` показывает состояние “`up`”, что свидетельствует о корректной работе физического и канального уровней. Остальные интерфейсы (`eth1-eth7`) находятся в состоянии “`down`”, так как не настроены и не подключены к активным устройствам. Локальный интерфейс `lo` также активен, что является стандартным поведением для работы системных служб маршрутизатора. Данная конфигурация обеспечивает базовую сетевую связность и позволяет окончным устройствам использовать маршрутизатор в качестве шлюза для взаимодействия между сегментами сети.

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. 3.37: Проверка подключения

В окне Wireshark проанализируем полученную информацию. Было отправлено 5 пакетов формата ICMP. В эхо-запросе источником является IP-адрес PC-1, а пунктом назначения – IP-адрес шлюза маршрутизатора. В эхо-ответе – наоборот. Также были сформированы ARP пакеты запрашивающий MAC-адрес шлюза маршрутизатора перед пингованием его и сообщающий этот MAC-адрес PC-1, а затем запрашивающие MAC-адрес PC-1 и сообщающие его шлюзу.

Анализ трафика показывает успешную работу сети с маршрутизатором FRR. Наблюдается стабильный обмен ICMP-пакетами между PC1 (192.168.1.10) и маршрутизатором (192.168.1.1). ARP-запросы подтверждают корректное разрешение IP-адресов в MAC-адреса. Изменяющиеся идентификаторы и порядковые номера ICMP-пакетов демонстрируют стандартное поведение протокола. Трафик подтверждает корректную настройку сетевых интерфейсов и двустороннюю связность между узлам.

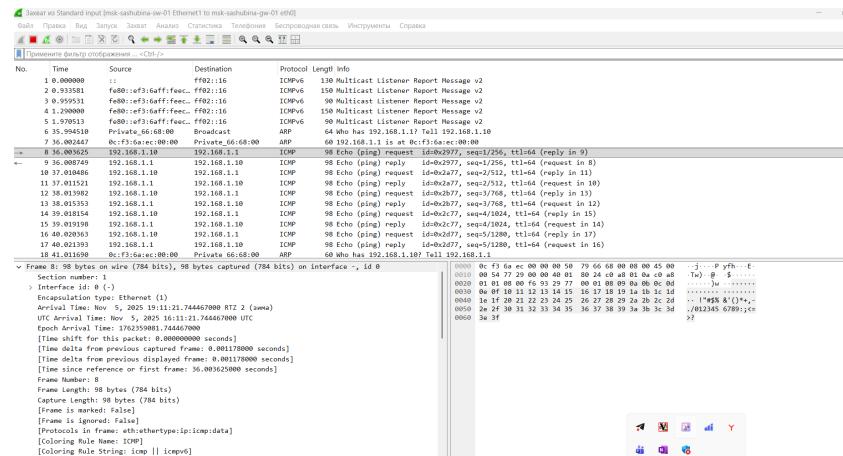


Рис. 3.38: Эхо-запрос

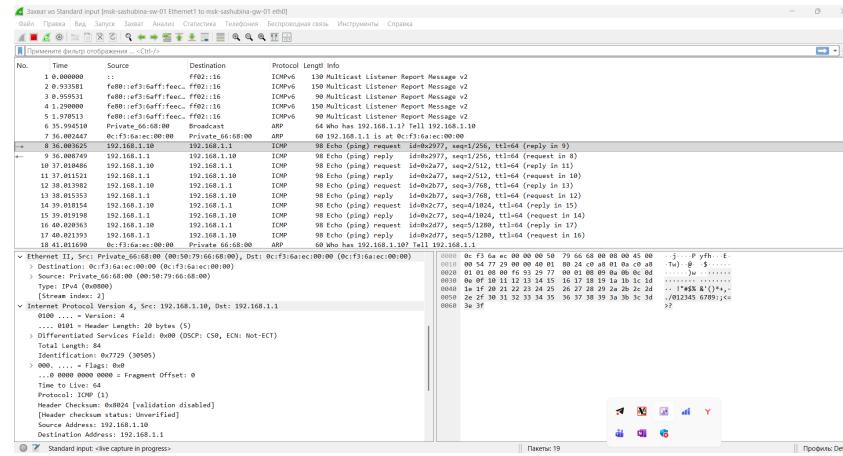


Рис. 3.39: Эхо-запрос

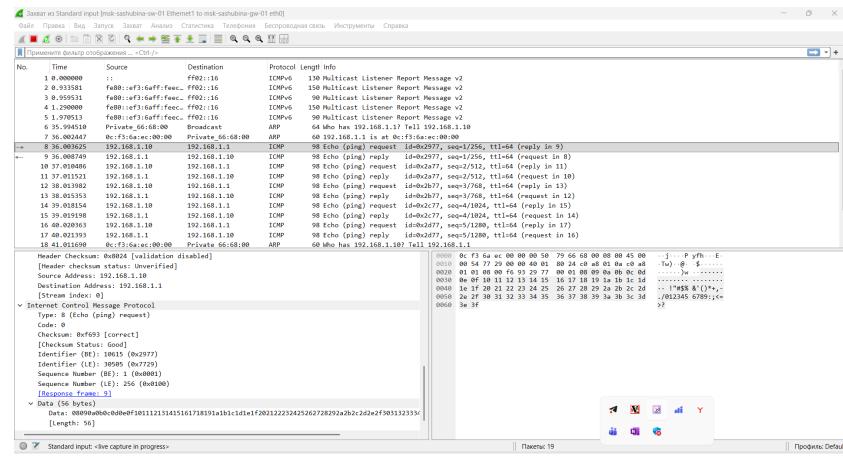


Рис. 3.40: Эхо-запрос

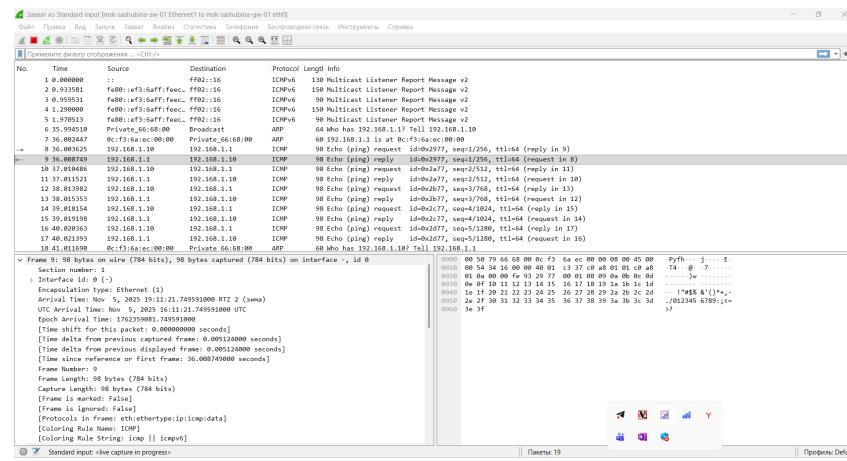


Рис. 3.41: Эхо-ответ

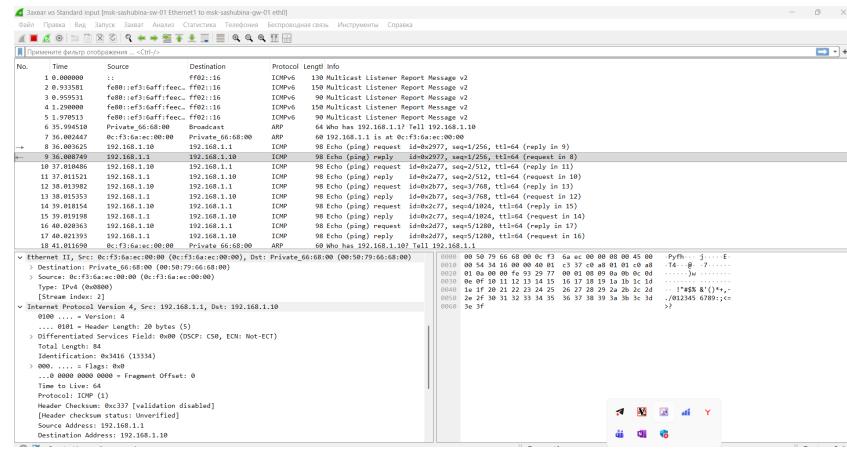


Рис. 3.42: Эхо-ответ

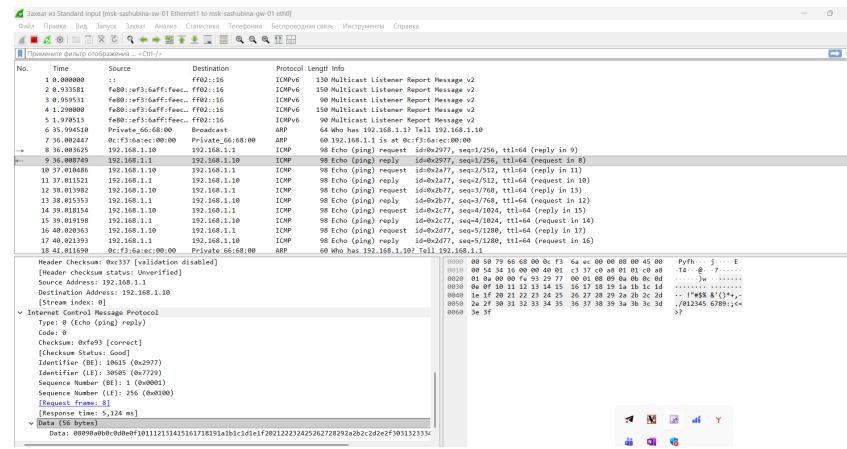


Рис. 3.43: Эхо-ответ

В конце остановим захват пакетов в Wireshark и остановим все устройства в проекте.

3.4 Моделирование простейшей сети на базе маршрутизатора VyOS в GNS3

1. Построить в GNS3 топологию сети, состоящей из маршрутизатора VyOS, коммутатора Ethernet и оконечного устройства.
2. Задать оконечному устройству IP-адрес в сети 192.168.1.0/24.
3. Присвоить интерфейсу маршрутизатора адрес 192.168.1.1/24
4. Проверить связь.

Запустим GNS3 VM и GNS3. Создадим новый проект.

В рабочей области GNS3 разместим VPCS, коммутатор Ethernet и маршрутизатор VyOS.

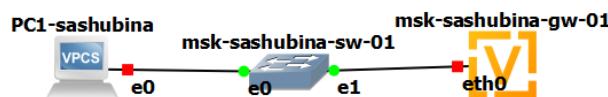


Рис. 3.44: Создание проекта

Изменим отображаемые названия устройств. Включим захват трафика на соединении между коммутатором и маршрутизатором. Запустим все устройства проекта. Откройте консоль всех устройств 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. 3.45: Настройка IP-адресации для интерфейса узла РС-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. 3.46: ввод логина и пароля

Установим систему на диск:

vyos@vyos:~\$ install image

У меня уже была система установлена

```

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. 3.47: 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. 3.48: перезагрузка

Для настройки маршрутизатора VyOS необходимо выполнить последовательность команд. Сначала переходим в режим конфигурирования с помощью команды `configure`, после чего приглашение командной строки меняется с `$` на `#`. Затем устанавливаем имя устройства командой `set system host-name msk-user-gw-01`, где следует заменить `user` на вашу учётную запись. Важно отметить, что изменения имени устройства вступят в силу только после применения и сохранения конфигурации с последующей перезагрузкой устройства. Далее настраиваем IP-адрес на интерфейсе `eth0` командой `set interfaces ethernet eth0 address 192.168.1.1/24`. Для просмотра внесённых изменений используем команду `compare`. Применяем изменения конфигурации командой `commit` и сохраняем её командой `save`.

```
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. 3.49: 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. 3.50: проверка настроек

Проверим подключение. Узел РС1 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. 3.51: Проверка соединения

В окне Wireshark проанализируем полученную информацию. Далее наблюдается обмен пятью парами ICMP-пакетов (ping). В каждой паре сначала следует запрос (Echo Request) от PC1 к маршрутизатору, после чего — ответ (Echo Reply) от маршрутизатора к PC1 с идентичными полями Identifier и Sequence Number. Этот обмен подтверждает двустороннюю IP-связность между PC1 и маршрутизатором. В завершение сессии успешно выполняется ARP-запрос, где маршрутизатор узнает MAC-адрес PC1, и PC1 отправляет ответ.

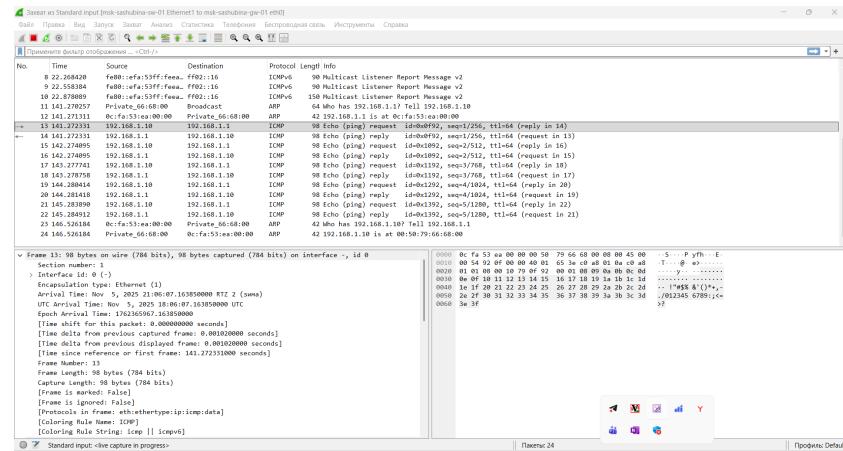


Рис. 3.52: Анализ трафика Wireshark

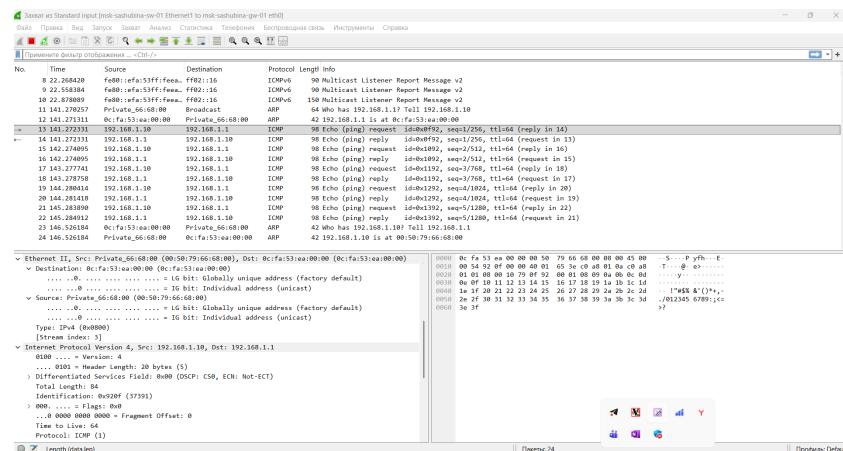


Рис. 3.53: Анализ трафика Wireshark

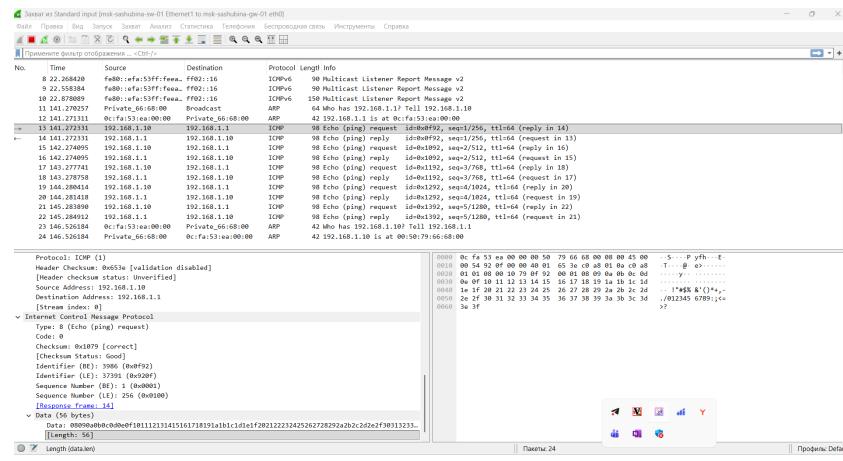


Рис. 3.54: Анализ трафика Wireshark

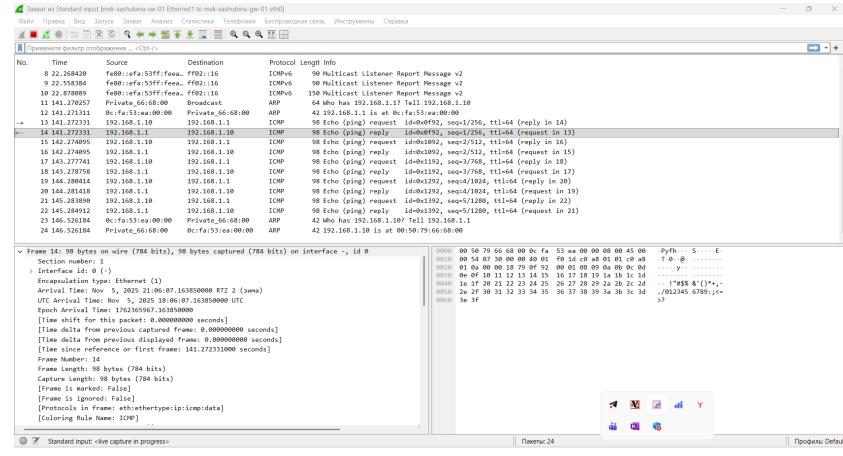


Рис. 3.55: Анализ трафика Wireshark

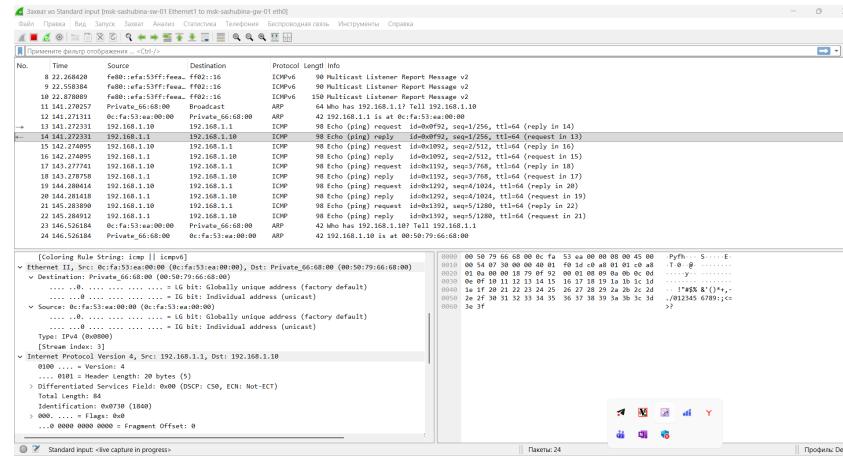


Рис. 3.56: Анализ трафика Wireshark

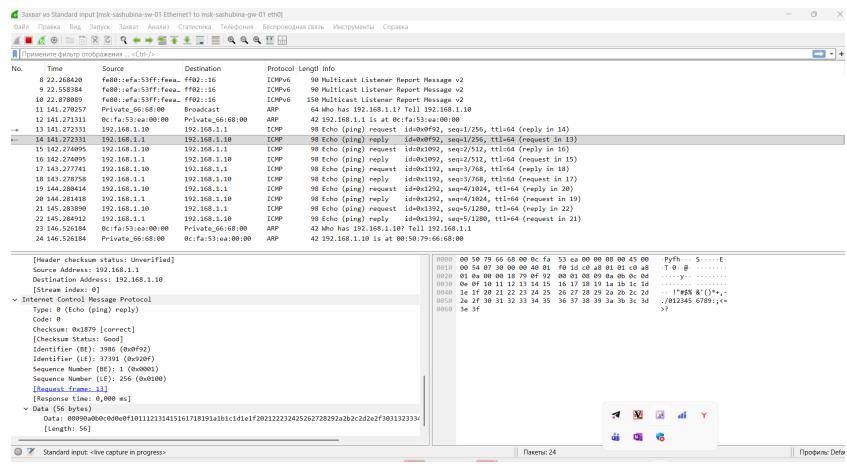


Рис. 3.57: Анализ трафика Wireshark

4 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.