

Отчёт по лабораторной работе №5

Дисциплина: Сетевые технологии

Мишина Анастасия Алексеевна

Содержание

1 Цель работы	6
2 Выполнение лабораторной работы	7
2.1 Анализ трафика в GNS3 посредством Wireshark	7
2.2 Моделирование простейшей сети на базе маршрутизатора FRR в GNS3	17
2.3 Моделирование простейшей сети на базе маршрутизатора VyOS в GNS3	21
3 Выводы	26

Список иллюстраций

2.1	Топология простейшей сети в GNS3	7
2.2	Задание IP-адреса для PC-1	8
2.3	Задание IP-адреса для PC-2	9
2.4	Пингование PC-2	9
2.5	Остановка всех узлов	10
2.6	Захват трафика, старт узлов	10
2.7	Информация по протоколу ARP	11
2.8	Эхо-запрос в ICMP-моде	11
2.9	Полученная информация по эхо-запросу в ICMP-моде к узлу PC-1	12
2.10	Эхо-ответ	12
2.11	Эхо-запрос в UDP-моде	13
2.12	Полученная информация по эхо-запросу в UDP-моде к узлу PC-1	13
2.13	Эхо-ответ	14
2.14	Эхо-запрос в TCP-моде	14
2.15	Полученная информация по эхо-запросу в TCP-моде к узлу PC-1	15
2.16	Полученная информация по эхо-запросу в TCP-моде к узлу PC-1	16
2.17	Полученная информация по эхо-запросу в TCP-моде к узлу PC-1	16
2.18	Топология сети с маршрутизатором FRR	17
2.19	Настройка IP-адресации для интерфейса узла PC-1	18
2.20	Настройка IP-адресации для интерфейса локальной сети маршрутизатора. Проверка конфигурации.	19
2.21	Отправка эхо-запросов с узла PC1 на адрес маршрутизатора	20
2.22	Полученная информация в Wireshark по ICMP-сообщениям	20
2.23	Топология сети с маршрутизатором VyOS	21
2.24	Настройка IP-адресации для интерфейса узла PC-1	22
2.25	Логин, проверка установки системы на диск	23
2.26	Режим конфигурирования: имя устройства, ip-адрес на ин- терфейсе eth0. Просмотр изменений, применение измене- ний, сохранение.	24

2.27 Пингование маршрутизатора	25
2.28 Полученная информация в Wireshark по ICMP-сообщению	25

Список таблиц

1 Цель работы

Построить простейшие модели сетей на базе коммутатора и маршрутизаторов FRR и VyOS в GNS3, проанализировать трафик посредством Wireshark.

2 Выполнение лабораторной работы

2.1 Анализ трафика в GNS3 посредством Wireshark

Для начала запустим GNS3 VM и GNS3, а также создадим новый проект. В рабочей области GNS3 разместим коммутатор Ethernet и два VPCS. В меню Configure изменим название устройства, включив в имя устройства имя моей учётной записи. Коммутатору присвоим название msk-aamishina-sw-01. Соединим VPCS с коммутатором и отобразим обозначение интерфейсов соединения (рис. fig. 2.1).

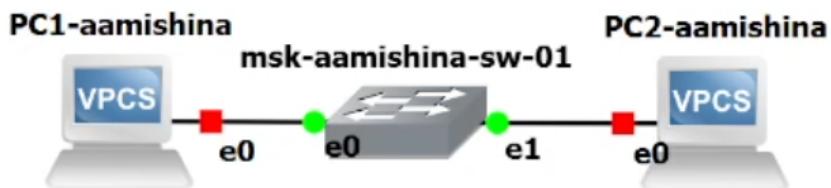


Рис. 2.1: Топология простейшей сети в GNS3

Зададим IP-адреса VPCS. Для этого с помощью меню, вызываемого правой кнопкой мыши, запустим Start PC-1, затем вызовем его терминал Console. Для просмотра синтаксиса возможных для ввода команд можно набрать ?. 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. Для сохранения конфигурации введем команду save (рис. fig. 2.2).

```

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

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

VPCS> 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'
  domain NAME                            Set local domain name to NAME

VPCS> ip 192.168.1.11/24 192.168.1.1
Checking for duplicate address...
PCI : 192.168.1.11 255.255.255.0 gateway 192.168.1
VPCS> save
Saving startup configuration to startup.vpc
. done
VPCS>

```



Рис. 2.2: Задание IP-адреса для РС-1

Аналогичным образом зададим IP-адрес 192.168.1.12 для РС-2. Пингуем соответственно IP-адрес РС-1. Получаем эхо-ответ от РС-1. Значит соединение наших РС работоспособно (рис. fig. 2.3).

```
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> 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

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=1.602 ms
84 bytes from 192.168.1.11 icmp_seq=2 ttl=64 time=1.899 ms
84 bytes from 192.168.1.11 icmp_seq=3 ttl=64 time=1.941 ms
84 bytes from 192.168.1.11 icmp_seq=4 ttl=64 time=2.136 ms
84 bytes from 192.168.1.11 icmp_seq=5 ttl=64 time=1.342 ms

VPCS>
```

Рис. 2.3: Задание IP-адреса для РС-2

Проверим работоспособность соединения между РС-1 и РС-2 с помощью команды ping. В терминале РС-1 введем команду ping и IP-адрес, присвоенный РС-2. Получаем эхо-ответ от РС-2 (возвращены 5 пакетов) (рис. fig. 2.4).

```
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

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

VPCS> ping 192.168.1.12
84 bytes from 192.168.1.12 icmp_seq=1 ttl=64 time=
84 bytes from 192.168.1.12 icmp_seq=2 ttl=64 time=
84 bytes from 192.168.1.12 icmp_seq=3 ttl=64 time=
84 bytes from 192.168.1.12 icmp_seq=4 ttl=64 time=
84 bytes from 192.168.1.12 icmp_seq=5 ttl=64 time=

VPCS>
```

Рис. 2.4: Пингование РС-2

Остановим в проекте все узлы (рис. fig. 2.5).

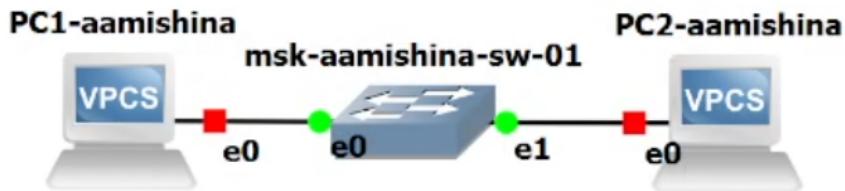


Рис. 2.5: Остановка всех узлов

Запустим на соединении между РС-1 и коммутатором анализатор трафика. Для этого щёлкнем правой кнопкой мыши на соединении, выберем в меню Start capture. После этого запустился Wireshark, а в проекте GNS3 на соединении появился значок лупы. В проекте GNS3 стартуем все узлы (рис. fig. 2.6).

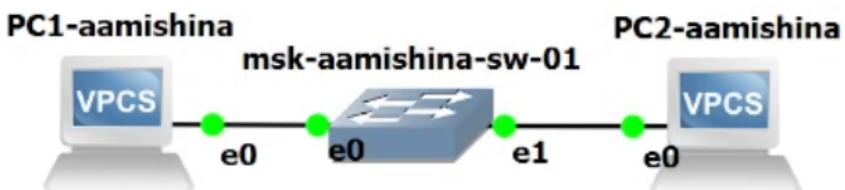


Рис. 2.6: Захват трафика, старт узлов

В окне Wireshark отобразится информация по протоколу ARP (рис. fig. 2.7). В поле кадра физического уровня мы можем узнать длину кадра (в моем случае было 64 бита). В поле канального уровня можем посмотреть мас-адреса источника и получателя. По нулевому и первому битам можем определить тип мас-адресов (получатель – локально администрируемый и широковещательный; источник – глобально администрируемый и индивидуальный).

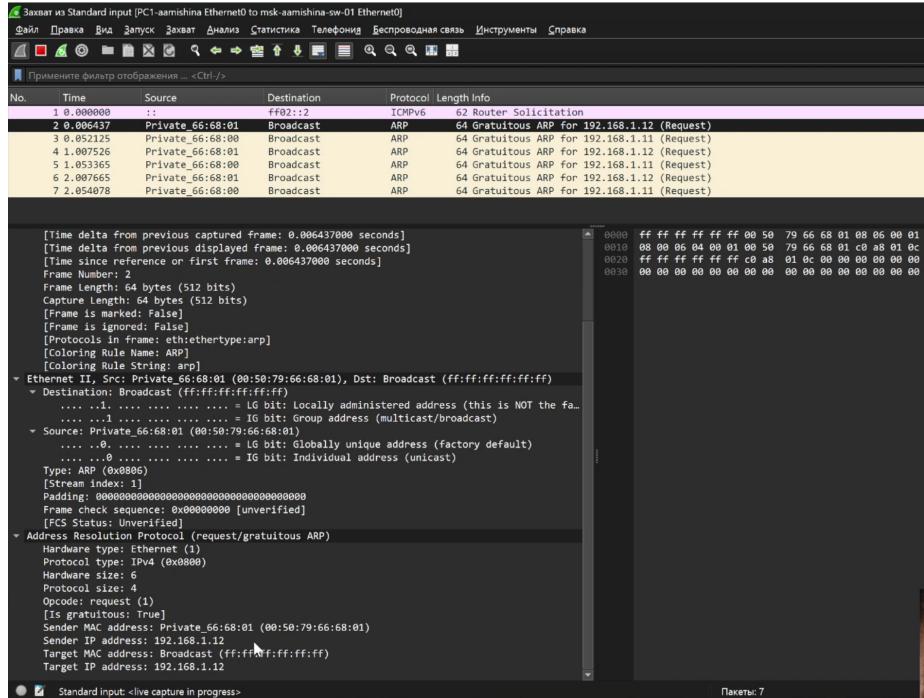


Рис. 2.7: Информация по протоколу ARP

В терминале РС-2 посмотрим информацию по опциям команды ping. Затем сделаем один эхо-запрос в ICMP-моде к узлу РС-1. Для этого используем опцию -1 (рис. fig. 2.8).

```
VPCS> ping 192.168.1.11 -1
84 bytes from 192.168.1.11 icmp_seq=1 ttl=64 time=
84 bytes from 192.168.1.11 icmp_seq=2 ttl=64 time=
84 bytes from 192.168.1.11 icmp_seq=3 ttl=64 time=
84 bytes from 192.168.1.11 icmp_seq=4 ttl=64 time=
84 bytes from 192.168.1.11 icmp_seq=5 ttl=64 time=
```

Рис. 2.8: Эхо-запрос в ICMP-моде

Далее откроем Wireshark и проанализируем эхо-запрос по протоколу ICMP (рис. fig. 2.9) и (рис. fig. 2.10). В поле канального уровня можем посмотреть mac-адреса источника и получателя. По нулевому и первому битам можем определить тип mac-адресов (получатель и источник - глобально администрируемые и одиночные, так как биты равны 0). В поле сетевого уровня указан протокол ICMP и IP-адреса источника (192.168.1.12,

то есть PC-2) и получателя (192.168.1.11, то есть PC-2).

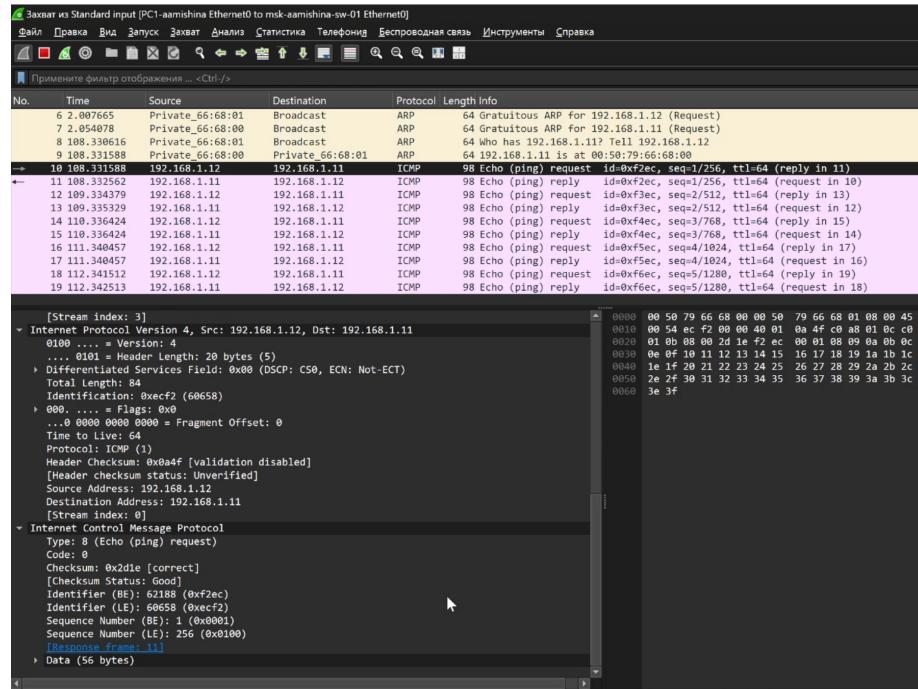


Рис. 2.9: Полученная информация по эхо-запросу в ICMP-моде к узлу PC-1

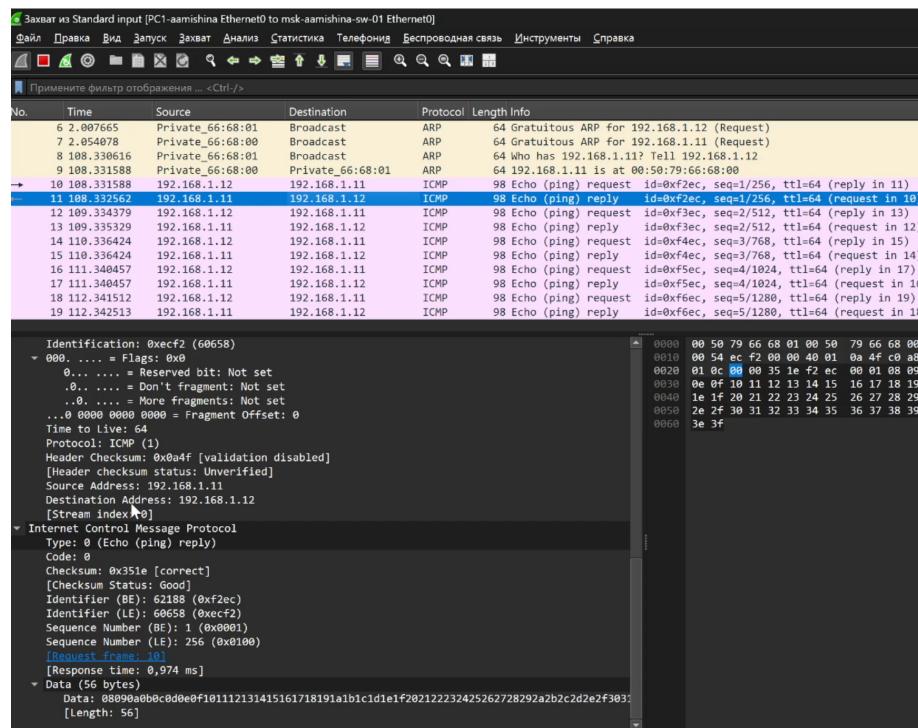


Рис. 2.10: Эхо-ответ

Сделаем один эхо-запрос в UDP-моде к узлу PC-1. Для этого используем опцию -2 (рис. fig. 2.11).

```

84 bytes from 192.168.1.11 icmp_seq=5 ttl=64 time=1
VPCS> ping 192.168.1.11 -2
84 bytes from 192.168.1.11 udp_seq=1 ttl=64 time=1
84 bytes from 192.168.1.11 udp_seq=2 ttl=64 time=1
84 bytes from 192.168.1.11 udp_seq=3 ttl=64 time=0
84 bytes from 192.168.1.11 udp_seq=4 ttl=64 time=1
84 bytes from 192.168.1.11 udp_seq=5 ttl=64 time=2

VPCS>

```

Рис. 2.11: Эхо-запрос в UDP-моде

Далее откроем Wireshark и проанализируем эхо-запрос по протоколу UDP (рис. fig. 2.12) и (рис. fig. 2.13). В поле канального уровня можем посмотреть mac-адреса источника и получателя. По нулевому и первому битам можем определить тип mac-адресов (получатель и источник - глобально администрируемые и одиночные, так как биты равны 0). В поле сетевого уровня указан протокол UDP и IP-адреса источника (192.168.1.12, то есть PC-2) и получателя (192.168.1.11, то есть PC-1). В поле протокола UDP указаны порты источника (17622) и получателя (7).

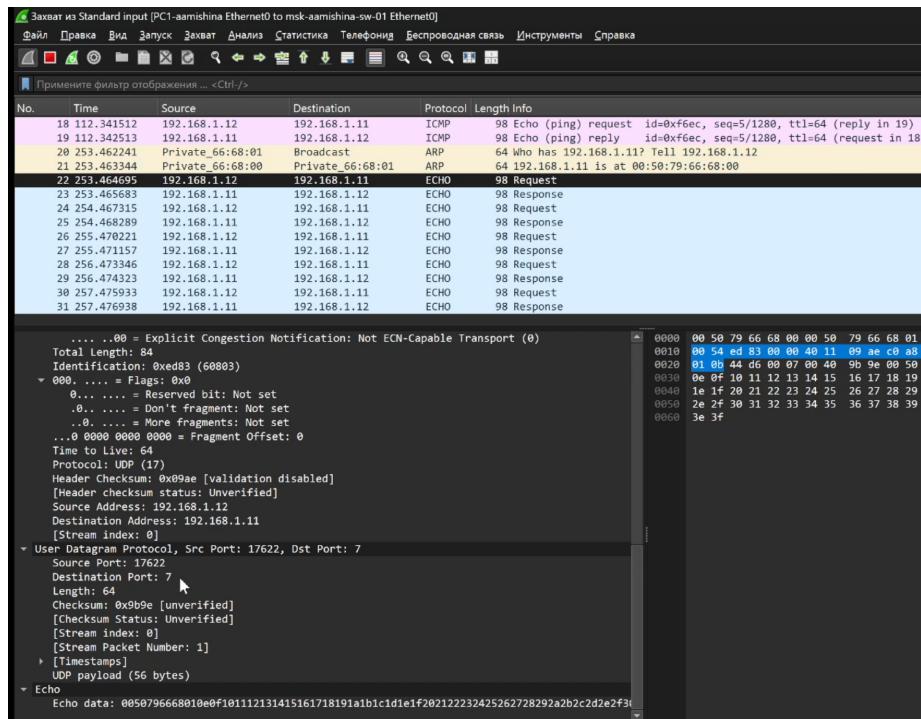


Рис. 2.12: Полученная информация по эхо-запросу в UDP-моде к узлу PC-1

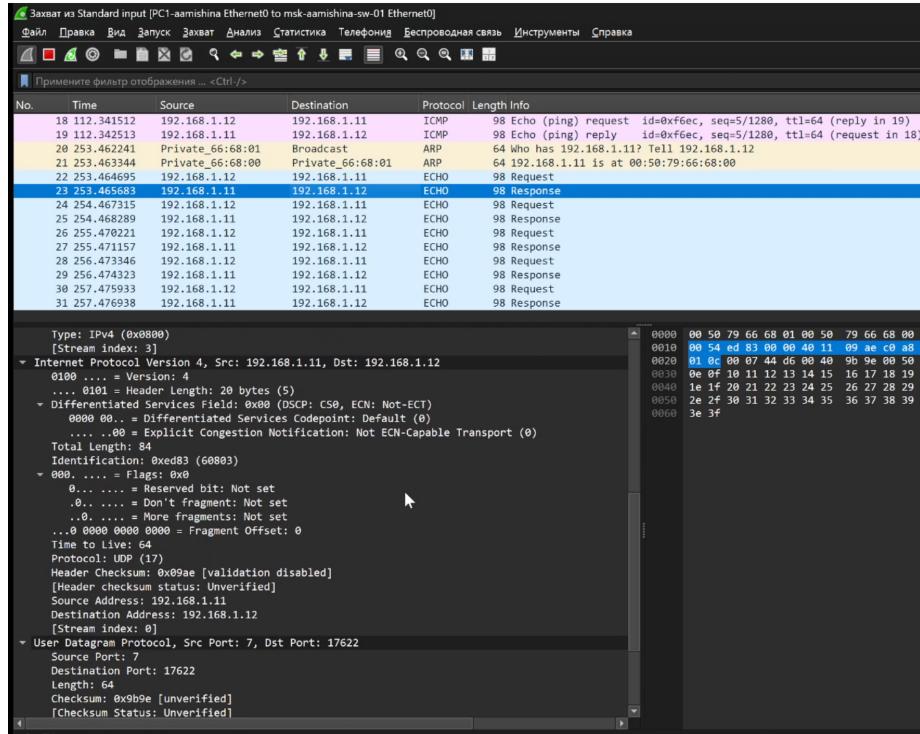


Рис. 2.13: Эхо-ответ

Сделаем один эхо-запрос в TCP-моде к узлу РС-1. Для этого используем опцию -3 (рис. fig. 2.14).

```
VPCS> ping 192.168.1.11 -3
Connect 7@192.168.1.11 seq=1 ttl=64 time=3.025 ms
SendData 7@192.168.1.11 seq=1 ttl=64 time=2.063 ms
Close 7@192.168.1.11 seq=1 ttl=64 time=2.842 ms
Connect 7@192.168.1.11 seq=2 ttl=64 time=2.998 ms
SendData 7@192.168.1.11 seq=2 ttl=64 time=2.675 ms
Close 7@192.168.1.11 seq=2 ttl=64 time=4.240 ms
Connect 7@192.168.1.11 seq=3 ttl=64 time=1.876 ms
SendData 7@192.168.1.11 seq=3 ttl=64 time=1.936 ms
Close 7@192.168.1.11 seq=3 ttl=64 time=4.037 ms
Connect 7@192.168.1.11 seq=4 ttl=64 time=2.293 ms
SendData 7@192.168.1.11 seq=4 ttl=64 time=1.544 ms
Close 7@192.168.1.11 seq=4 ttl=64 time=3.530 ms
Connect 7@192.168.1.11 seq=5 ttl=64 time=2.934 ms
SendData 7@192.168.1.11 seq=5 ttl=64 time=8.425 ms
Close 7@192.168.1.11 seq=5 ttl=64 time=3.152 ms
```

Рис. 2.14: Эхо-запрос в TCP-моде

Далее откроем Wireshark и проанализируем эхо-запрос по протоколу TCP. В поле канального уровня можем посмотреть мас-адреса источника и получателя. По нулевому и первому битам можем определить тип

mac-адресов (получатель и источник - глобально администрируемые и одиночные, так как биты равны 0). В поле сетевого уровня указан протокол TCP и IP-адреса источника (192.168.1.12, то есть PC-2) и получателя (192.168.1.11, то есть PC-1).

В поле протокола TCP можем узнать порты источника (20665) и получателя (7). А также посмотреть, как работает handshake протокола TCP. На первом шаге установлен флаг SYN (рис. fig. 2.15), а также Порядковому номеру (Sequence Number) присвоено начальное 32-битное значение ISSa (в нашем случае 884368193).

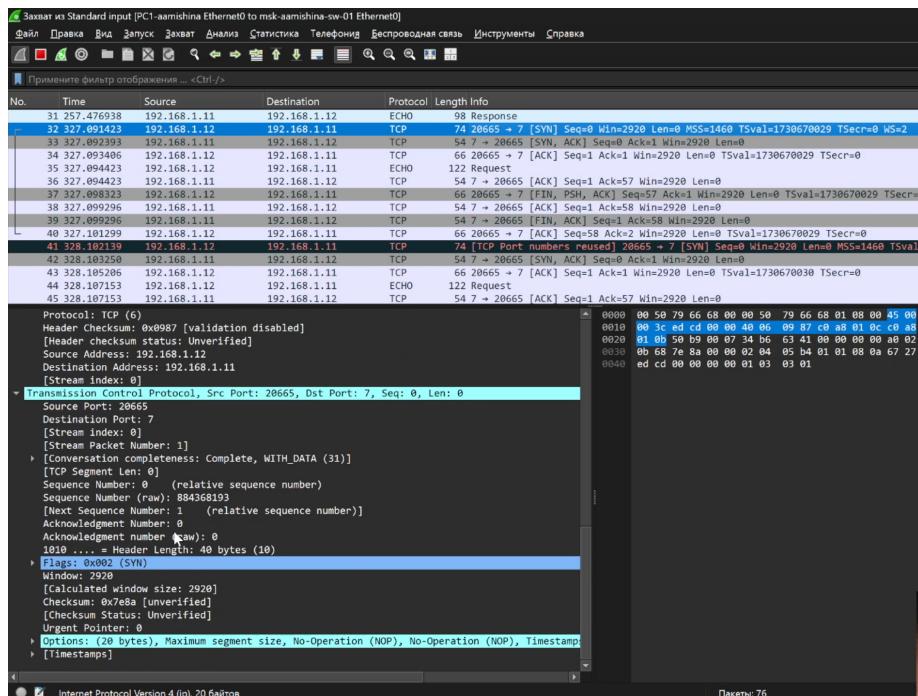


Рис. 2.15: Полученная информация по эхо-запросу в TCP-моде к узлу PC-1

На втором шаге установлены флаги SYN и ACK (рис. fig. 2.16).

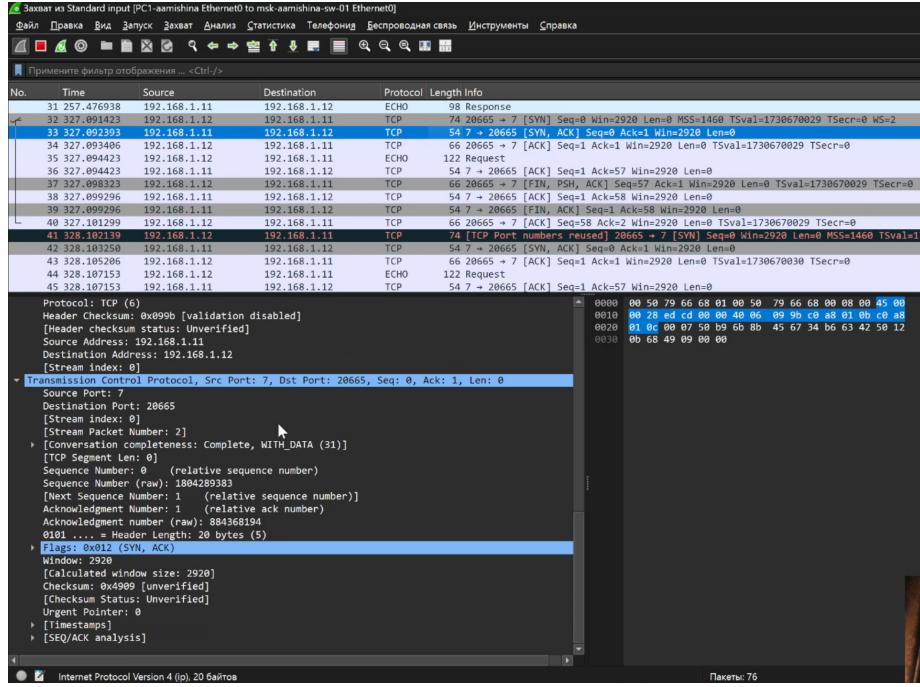


Рис. 2.16: Полученная информация по эхо-запросу в TCP-моде к узлу PC-1

На третьем шаге установлен флаг ACK (рис. fig. 2.17).

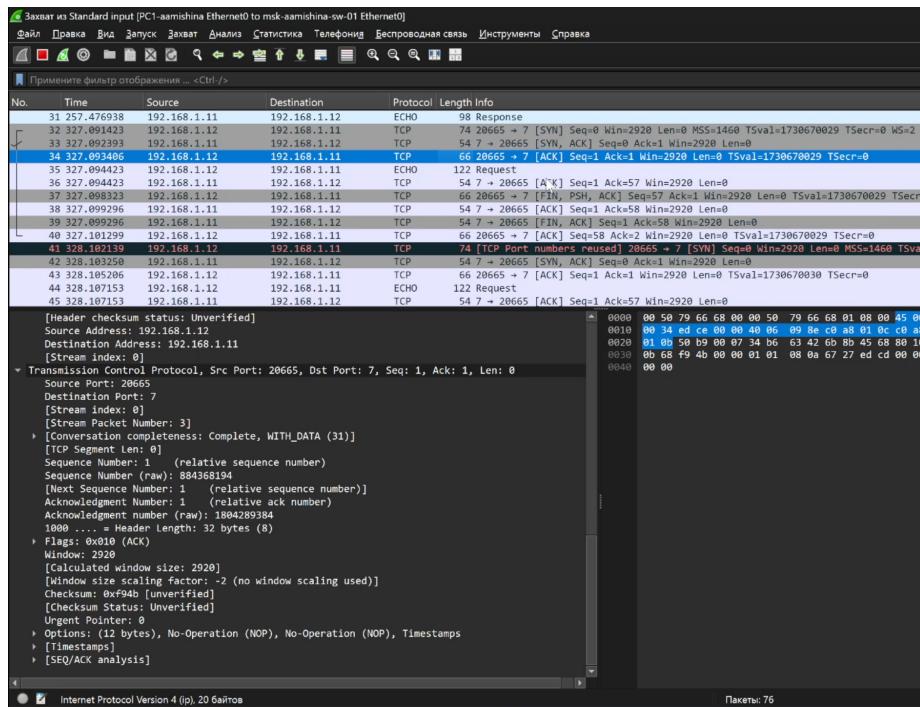


Рис. 2.17: Полученная информация по эхо-запросу в TCP-моде к узлу PC-1

Теперь можно остановить захват трафика в Wireshark.

2.2 Моделирование простейшей сети на базе маршрутизатора FRR в GNS3

Теперь нам нужно построить в GNS3 топологию сети (рис. fig. 2.18), состоящей из маршрутизатора FRR, коммутатора Ethernet и оконечного устройства. Изменим отображаемые названия устройств. Коммутатору присвоим название msk-aamishina-sw-01, маршрутизатору – по принципу msk-aamishina-gw-01, VPCS – по принципу PC1-aamishina. Включим захват трафика на соединении между коммутатором и маршрутизатором. Запустим все устройства проекта, а также откроем консоль всех устройств проекта.

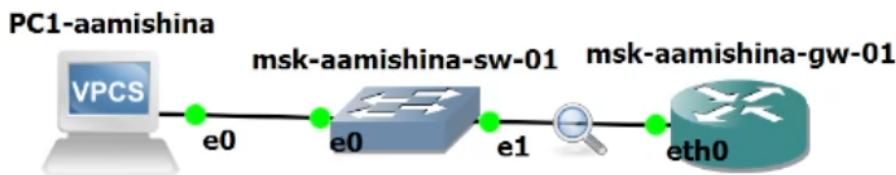


Рис. 2.18: Топология сети с маршрутизатором FRR

Настроим IP-адресацию для интерфейса узла PC1 (рис. fig. 2.19):

```
ip 192.168.1.10/24 192.168.1.1  
save  
show ip
```

```
PC1-aamishina - PuTTY

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> 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. 2.19: Настройка IP-адресации для интерфейса узла РС-1

Настроим IP-адресацию для интерфейса локальной сети маршрутизатора. Проверим конфигурацию маршрутизатора и настройки IP-адресации (рис. fig. 2.20).

```
msk-aamishina-gw-01 - PuTTY
% Unknown command: Router# configure terminal
frr#
frr#
frr#
frr# configure terminal
frr(config)# hostname msk-aamishina-gw-01
msk-aamishina-gw-01(config)# exit
msk-aamishina-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-aamishina-gw-01# configure terminal
msk-aamishina-gw-01(config)# interface eth0
msk-aamishina-gw-01(config-if)# ip address 192.168.1.1/24
msk-aamishina-gw-01(config-if)# no shutdown
msk-aamishina-gw-01(config-if)# exit
msk-aamishina-gw-01(config)# exit
msk-aamishina-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-aamishina-gw-01# show running-config
Building configuration...

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

Рис. 2.20: Настройка IP-адресации для интерфейса локальной сети маршрутизатора. Проверка конфигурации.

Проверим подключение. Узел PC1 успешно отправляет эхо-запросы на адрес маршрутизатора 192.168.1.1 (рис. fig. 2.21).

```
VPCS> ping 192.168.1.1
84 bytes from 192.168.1.1 icmp_seq=1 ttl=64 time=14.388 ms
84 bytes from 192.168.1.1 icmp_seq=2 ttl=64 time=7.265 ms
84 bytes from 192.168.1.1 icmp_seq=3 ttl=64 time=5.302 ms
84 bytes from 192.168.1.1 icmp_seq=4 ttl=64 time=8.868 ms
84 bytes from 192.168.1.1 icmp_seq=5 ttl=64 time=3.492 ms

VPCS> █
```

Рис. 2.21: Отправка эхо-запросов с узла PC1 на адрес маршрутизатора

В окне Wireshark проанализируем полученную информацию (рис. fig. 2.22). В поле канального уровня можем посмотреть mac-адреса источника и получателя. По нулевому и первому битам можно определить типа mac-адресов (получатель и источник - глобально администрируемые и одиночные, так как биты равны 0). В поле сетевого уровня указано протокол ICMP и IP-адреса источника (192.168.1.10, то есть PC-1) и получателя (192.168.1.1, то есть маршрутизатор FRR).

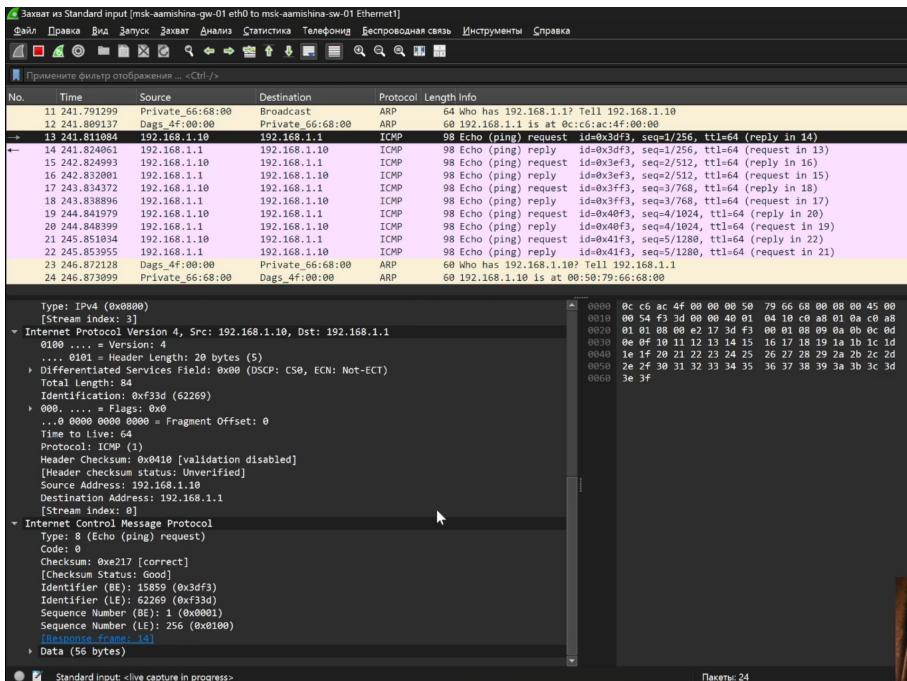


Рис. 2.22: Полученная информация в Wireshark по ICMP-сообщениям

Останавливаем захват пакетов в Wireshark. Останавливаем все устройства в проекте.

2.3 Моделирование простейшей сети на базе маршрутизатора VyOS в GNS3

В рабочей области GNS3 разместим VPCS, коммутатор Ethernet и маршрутизатор VyOS. Изменим отображаемые названия устройств. Коммутатору присвоим название msk-aamishina-sw-01, маршрутизатору — по принципу msk-aamishina-gw-01, VPCS — по принципу PC1-aamishina. Включим захват трафика на соединении между коммутатором и маршрутизатором (рис. fig. 2.23) (появится значок лупы). Запустим все устройства проекта и откроем консоль всех устройств проекта.

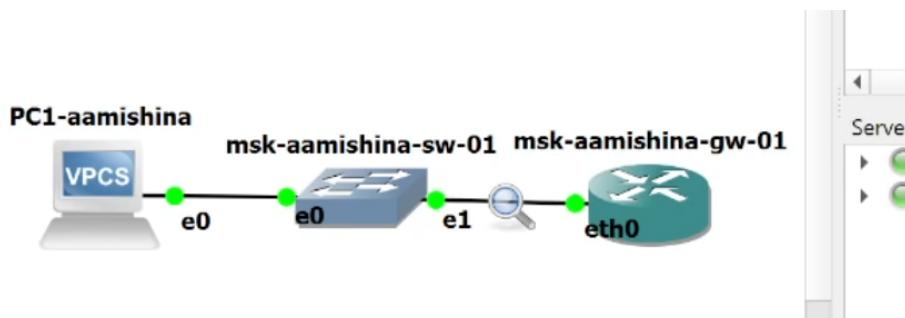
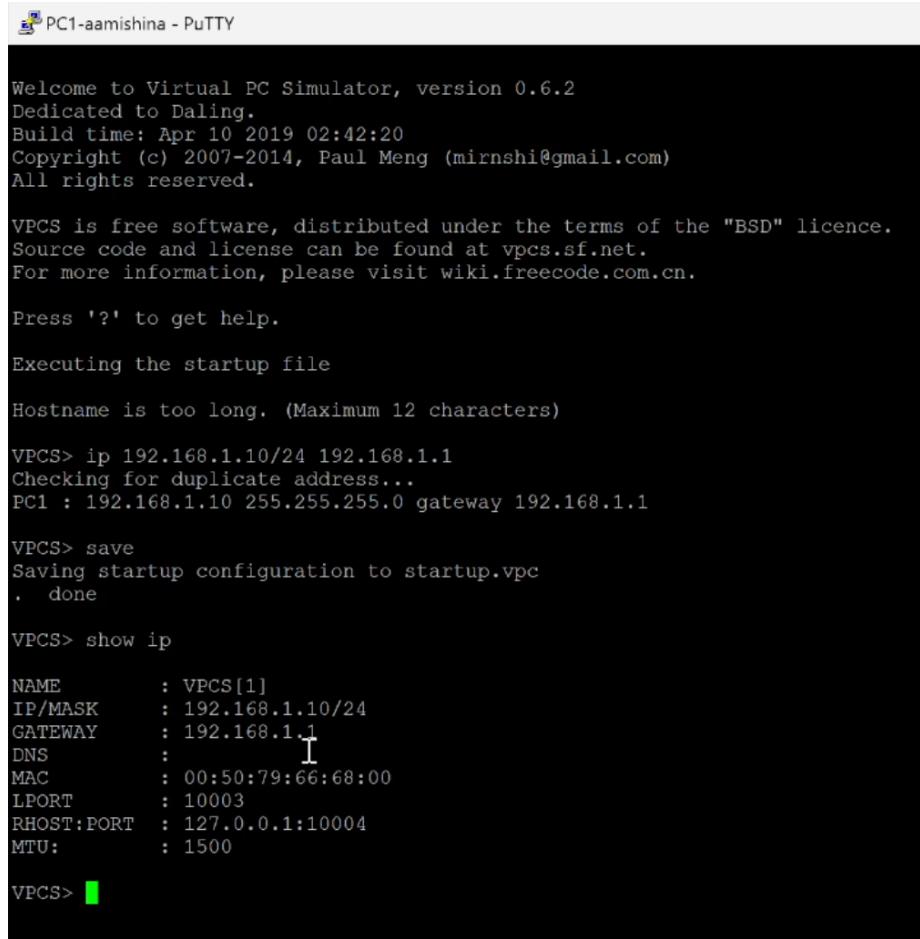


Рис. 2.23: Топология сети с маршрутизатором VyOS

Откроем окно терминала РС-1 и настроим IP-адресацию для интерфейса этого узла (рис. fig. 2.24).



```
PC1-aamishina - PuTTY

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> 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. 2.24: Настройка IP-адресации для интерфейса узла РС-1

Настроим маршрутизатор VyOS (рис. fig. 2.25): Во-первых, после загрузки нужно ввести логин vyos и пароль vyos. В рабочем режиме в командной строке отобразится символ \$. Далее надо установить систему на диск с помощью команды install image, но у меня эта система уже была установлена. На всякий случай перезапускаем маршрутизатор.

The screenshot shows a PuTTY terminal window titled "msk-aamishina-gw-01 - PuTTY". The terminal displays the kernel boot log, which includes messages from various drivers like EXT4, loop, squashfs, ehci_hcd, uhci_hcd, ohci_hcd, and usbcore, along with systemd startup logs. It also shows the system detecting kvm, architecture x86-64, and setting the hostname to <vyos>. The log ends with a message about bridge filtering via arp/ip/ip6tables and a note to update scripts to load br_netfilter if needed.

Below the log, the system prompts for a login:

```
Welcome to VyOS - vyos ttyS0

vyos login: vyos
Password:
Welcome to VyOS!
```

It then displays a banner:

```
Check out project news at https://blog.vyos.io
and feel free to report bugs at https://vyos.dev
```

And provides configuration instructions:

```
You can change this banner using "set system login banner post-login"

VyOS is a free software distribution that includes multiple components.
you can check individual component licenses under https://vyos.dev/licenses/
vyos@vyos:~$ install image
You are trying to install from an already installed image file.
image file to install or URL must be specified.
Exiting...
vyos@vyos:~$ reboot
Are you sure you want to reboot this system? [y/N]
```

Рис. 2.25: Логин, проверка установки системы на диск

Следующим шагом перейдем в режим конфигурирования с помощью команды `configure`. Изменим имя устройства командой `set system host-name msk-aamishina-gw-01`. Зададим IP-адрес на интерфейсе `eth0` командой `set interfaces ethernet eth0 address 192.168.1.1/24`. Посмотрим внесённые в конфигурацию изменения с помощью команды `compare`. Далее приме-

ним изменения в конфигурации и сохраним саму конфигурацию с помощью команд commit и save. Посмотрим информацию об интерфейсах маршрутизатора с помощью команды show interfaces. Выйдем из режима конфигурирования, используя команду exit (рис. fig. 2.26).

```
You can change this banner using "set system login banner post-login"
VyOS is a free software distribution that includes multiple components
you can check individual component licenses under /usr/share/doc/*
vyos@vyos:~$ configure
[edit]
vyos@vyos# system host-name msk-aamishina-gw-01
    Invalid command: [system]

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

Can't configure both static IPv4 and DHCP address on the same interface
[[interfaces ethernet eth0]] failed
Commit failed
[edit]
vyos@vyos# delete interfaces ethernet eth0 address dhcp
[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:19:ee:89:00:00
    }
    ethernet eth1 {
        hw-id 0c:19:ee:89:00:01
    }
    ethernet eth2 {
        hw-id 0c:19:ee:89:00:02
    }
    loopback lo {
    }
[edit]
vyos@vyos# exit
exit
vyos@vyos:~$
```

Рис. 2.26: Режим конфигурирования: имя устройства, ip-адрес на интерфейсе eth0. Просмотр изменений, применение изменений, сохранение.

Теперь проверим подключение. Узел PC1 должен успешно отправлять эхо-запросы на адрес маршрутизатора 192.168.1.1. Для проверки пингуем маршрутизатор (рис. fig. 2.27). Получили эхо-ответ (4 пакета).

```
VPCS> ping 192.168.1.1
84 bytes from 192.168.1.1 icmp_seq=1 ttl=64 time=4.476 ms
84 bytes from 192.168.1.1 icmp_seq=2 ttl=64 time=5.166 ms
84 bytes from 192.168.1.1 icmp_seq=3 ttl=64 time=4.886 ms
84 bytes from 192.168.1.1 icmp_seq=4 ttl=64 time=19.533 ms
84 bytes from 192.168.1.1 icmp_seq=5 ttl=64 time=6.896 ms
VPCS>
```

Рис. 2.27: Пингование маршрутизатора

В окне Wireshark проанализируем полученную информацию (рис. fig. 2.28). В поле канального уровня можем посмотреть mac-адреса источника и получателя. По нулевому и первому битам можем определить тип mac-адресов (получатель и источник - глобально администрируемые и одиночные, так как биты равны 0). В поле сетевого уровня указан протокол ICMP и IP-адреса источника (192.168.1.10, то есть PC-1) и получателя (192.168.1.1, то есть маршрутизатор VyOS).

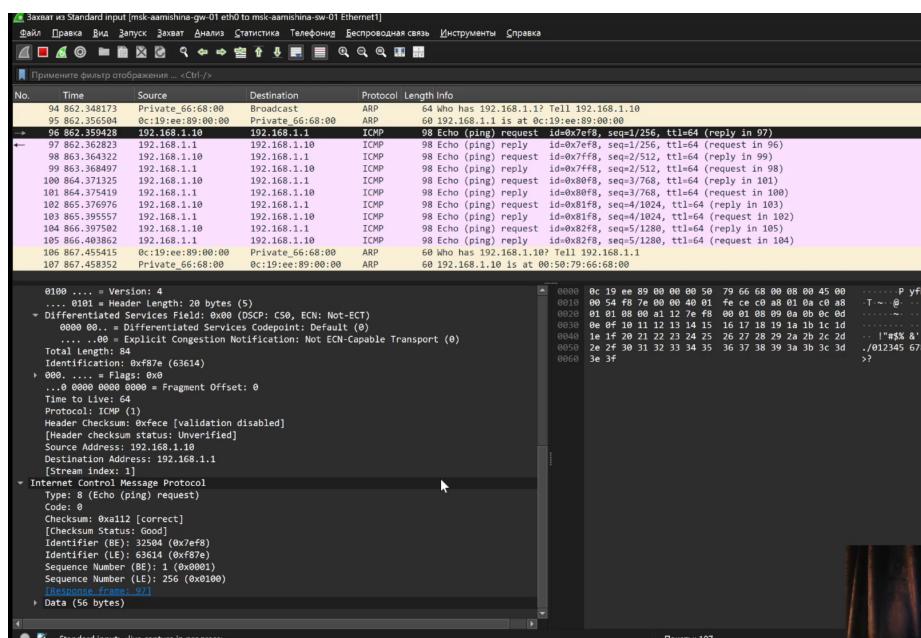


Рис. 2.28: Полученная информация в Wireshark по ICMP-сообщению

3 Выводы

В процессе выполнения лабораторной работы мы построили простейшие модели сетей на базе коммутатора и маршрутизаторов FRR и VyOS в GNS3, проанализировали трафик посредством Wireshark.