РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

дисциплина: Сетевые технологии

Студент: Сахно Никита

Группа: НФИбд-02-23

МОСКВА

2025 г.

Цель работы

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

№1

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-nvsakhno-sw-01. Соединим VPCS с коммутатором и отобразим обозначение интерфейсов соединения.

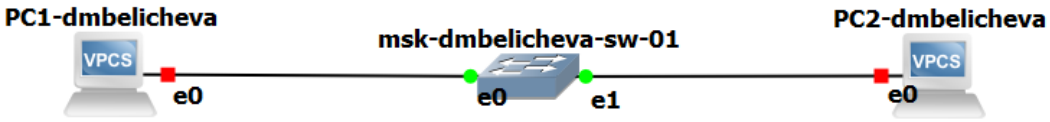

Рисунок 1. Топология простейшей сети в GNS3

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 */?*.

```
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


Checking for duplicate address...
PC1 : 192.168.1.11 255.255.255.0 gateway 192.168.1.1

VPCS> /?

?                       Print help
! COMMAND [ARG ...]     Invoke an OS COMMAND with optional ARG(s)
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>
```

Рисунок 2. Окно терминала PC1

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.

```
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
```

Рисунок 3. Задание IP-адреса для PC-1

Аналогичным образом зададим IP-адрес 192.168.1.12 для PC-2.

```
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>
```

Рисунок 4. Задание IP-адреса для PC-2

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

В терминале PC-1 введем команду ping и IP-адрес, присвоенный PC-2. Получаем эхо-ответ от PC-2 (возвращены четыре пакета).

```
VPCS> ping 192.168.1.12
84 bytes from 192.168.1.12 icmp_seq=1 ttl=64 time=1.174 ms
84 bytes from 192.168.1.12 icmp_seq=2 ttl=64 time=1.268 ms
84 bytes from 192.168.1.12 icmp_seq=3 ttl=64 time=0.981 ms
84 bytes from 192.168.1.12 icmp_seq=4 ttl=64 time=1.651 ms
84 bytes from 192.168.1.12 icmp_seq=5 ttl=64 time=0.932 ms

VPCS>
```

Рисунок 5. Пингование PC-2

То же самое делаем из терминала PC-2, пингуем соответственно IP-адрес PC-1. Получаем эхо-ответ от PC-1. Значит соединение наших PC работоспособно.

```
VPCS> ping 192.168.1.11
84 bytes from 192.168.1.11 icmp_seq=1 ttl=64 time=1.063 ms
84 bytes from 192.168.1.11 icmp_seq=2 ttl=64 time=1.139 ms
84 bytes from 192.168.1.11 icmp_seq=3 ttl=64 time=1.197 ms
84 bytes from 192.168.1.11 icmp_seq=4 ttl=64 time=1.198 ms
84 bytes from 192.168.1.11 icmp_seq=5 ttl=64 time=1.127 ms

VPCS>
```

Рисунок 6. Пингование PC-1

Остановим в проекте все узлы.

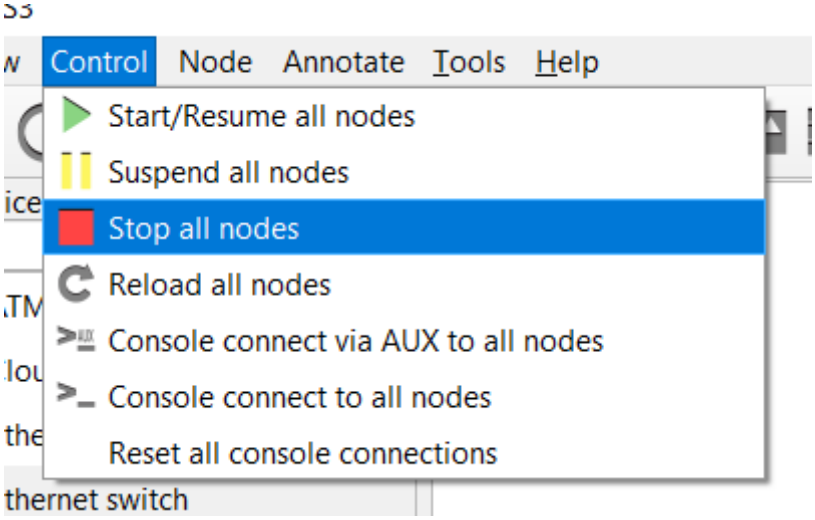

Рисунок 7. Остановка всех узлов

№2

Запустим на соединении между PC-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.

3

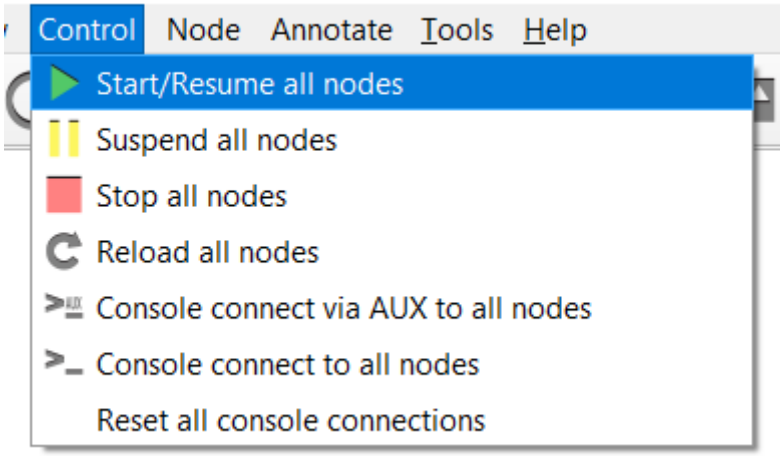

Рисунок 8. Старт всех узлов

В окне Wireshark отобразится информация по протоколу ARP.

В поле кадра физического уровня мы можем узнать длину кадра (в моем случае было 64 бита). В поле канального уровня можем посмотреть mac-адреса источника и получателя. По нулевому и первому битам можем определить тип mac-адресов (получатель – локально администрируемый и широковещательный; источник - глобально администрируемый и одиночный).

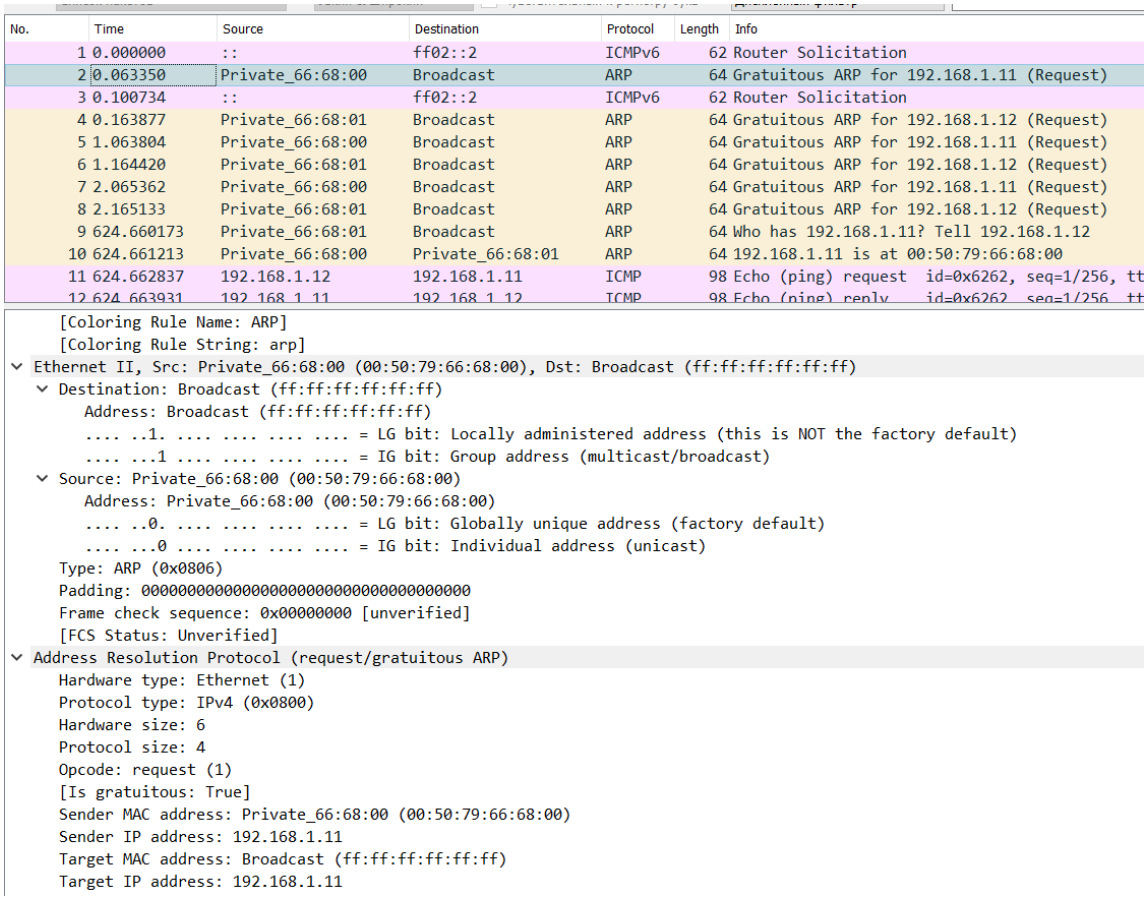

Рисунок 9. Информация по протоколу ARP

В терминале PC-2 посмотрим информацию по опциям команды ping. Затем сделаем один эхо-запрос в ICMP-моде к узлу PC-1. Для этого используем опцию -1.

```
VPCS> ping 192.168.1.11 -1
84 bytes from 192.168.1.11 icmp_seq=1 ttl=64 time=1.610 ms
84 bytes from 192.168.1.11 icmp_seq=2 ttl=64 time=1.630 ms
84 bytes from 192.168.1.11 icmp_seq=3 ttl=64 time=1.831 ms
84 bytes from 192.168.1.11 icmp_seq=4 ttl=64 time=1.621 ms
84 bytes from 192.168.1.11 icmp_seq=5 ttl=64 time=1.610 ms
```

Рисунок 10. Эхо-запрос в ICMP-моде

Далее откроем Wireshark и проанализируем эхо-запрос по протоколу ICMP. 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).

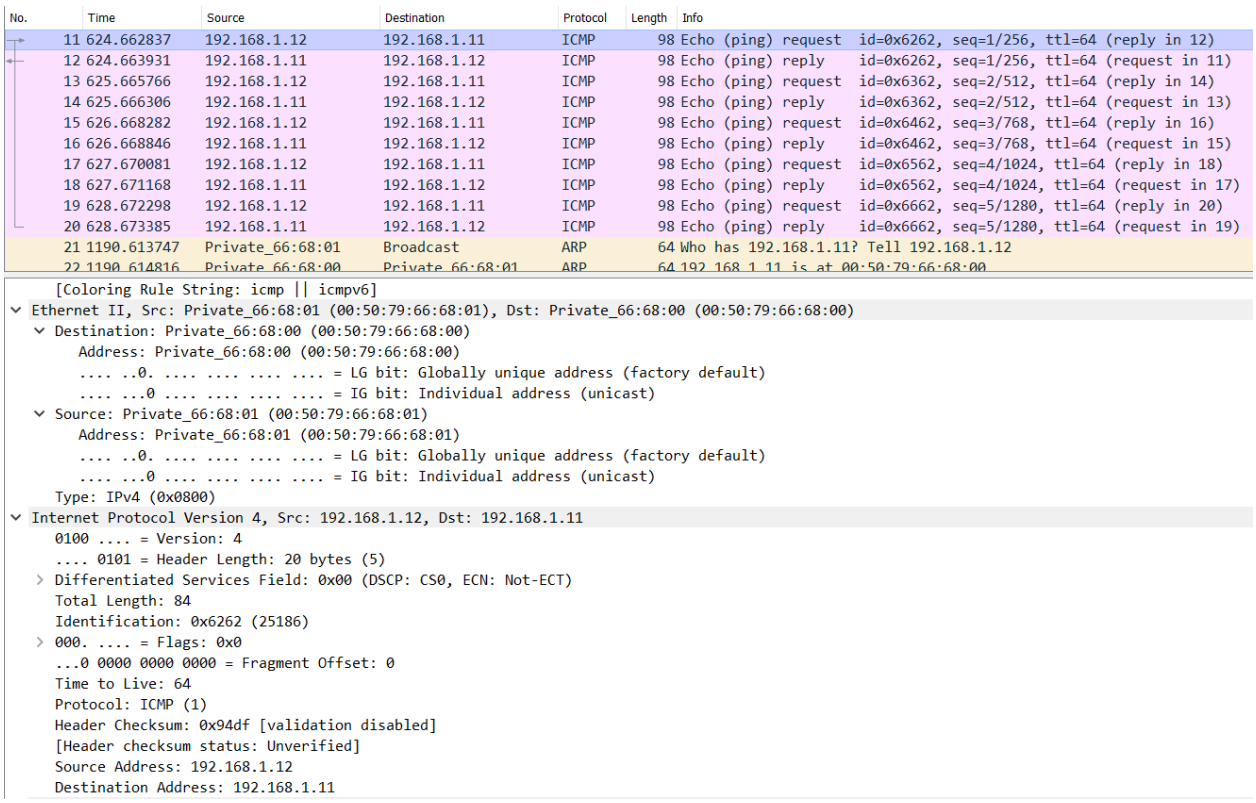

Рисунок 11. Полученная информация по эхо-запросу в ICMP-моде к узлу PC-1

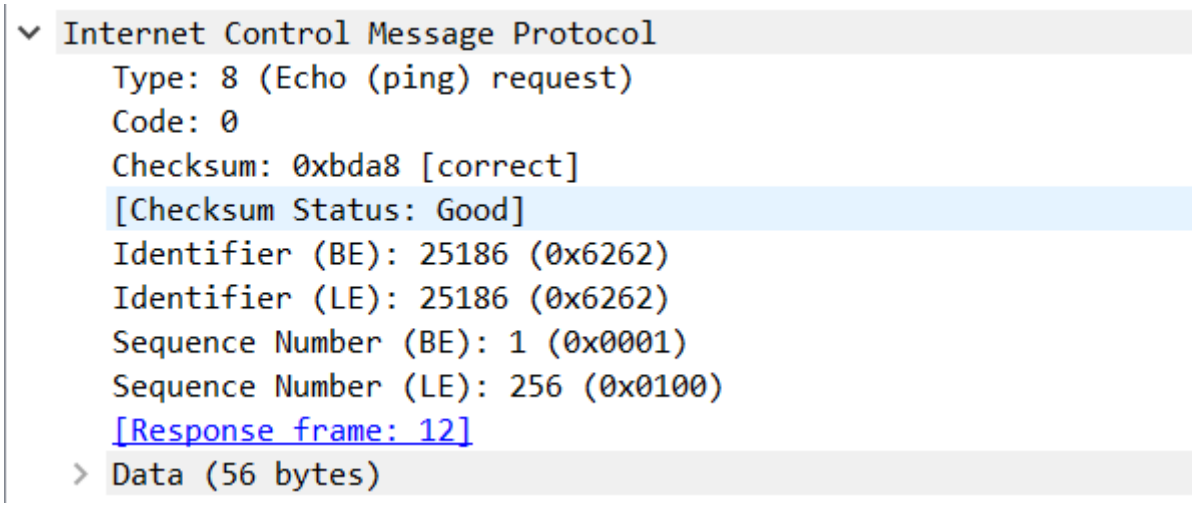

Рисунок 12 Поле уровня протокола ARP

Сделаем один эхо-запрос в UDP-моде к узлу PC-1. Для этого используем опцию -2.

```
VPCS> ping 192.168.1.11 -2
84 bytes from 192.168.1.11 udp_seq=1 ttl=64 time=1.429 ms
84 bytes from 192.168.1.11 udp_seq=2 ttl=64 time=1.537 ms
84 bytes from 192.168.1.11 udp_seq=3 ttl=64 time=1.523 ms
84 bytes from 192.168.1.11 udp_seq=4 ttl=64 time=1.614 ms
84 bytes from 192.168.1.11 udp_seq=5 ttl=64 time=1.479 ms
```

Рисунок 13. Эхо-запрос в UDP-моде

Далее откроем Wireshark и проанализируем эхо-запрос по протоколу UDP. 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-2). В поле протокола UDP указаны порты источника (17163) и получателя (7).

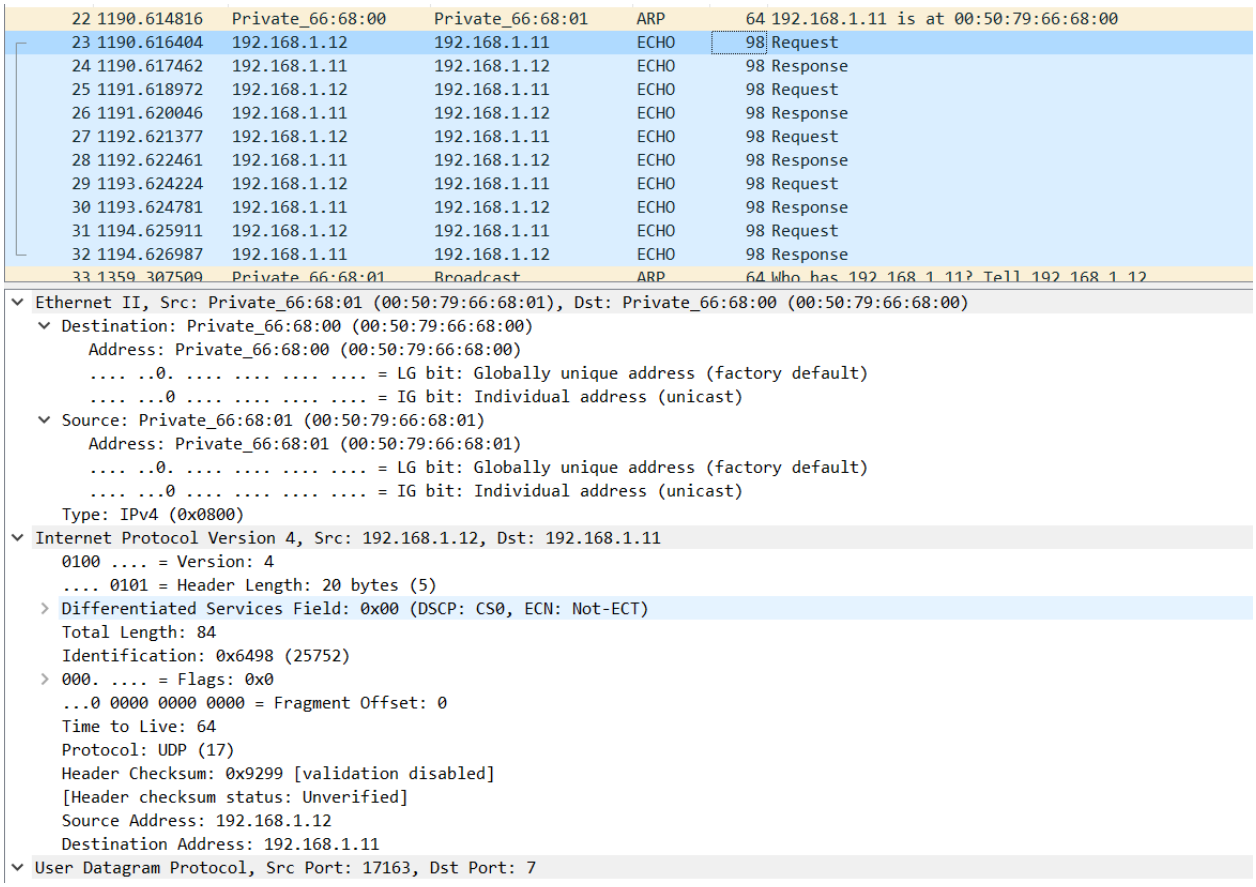

Рисунок 14. Полученная информация по эхо-запросу в UDP-моде к узлу PC-1

Сделаем один эхо-запрос в TCP-моде к узлу PC-1. Для этого используем опцию -3.

```
VPCS> ping 192.168.1.11 -3
Connect   7@192.168.1.11 seq=1 ttl=64 time=2.169 ms
SendData  7@192.168.1.11 seq=1 ttl=64 time=2.162 ms
Close     7@192.168.1.11 seq=1 ttl=64 time=3.491 ms
Connect   7@192.168.1.11 seq=2 ttl=64 time=2.198 ms
SendData  7@192.168.1.11 seq=2 ttl=64 time=2.178 ms
Close     7@192.168.1.11 seq=2 ttl=64 time=3.282 ms
Connect   7@192.168.1.11 seq=3 ttl=64 time=2.211 ms
SendData  7@192.168.1.11 seq=3 ttl=64 time=2.209 ms
Close     7@192.168.1.11 seq=3 ttl=64 time=3.296 ms
Connect   7@192.168.1.11 seq=4 ttl=64 time=2.215 ms
SendData  7@192.168.1.11 seq=4 ttl=64 time=2.192 ms
Close     7@192.168.1.11 seq=4 ttl=64 time=3.291 ms
Connect   7@192.168.1.11 seq=5 ttl=64 time=2.201 ms
SendData  7@192.168.1.11 seq=5 ttl=64 time=2.177 ms
Close     7@192.168.1.11 seq=5 ttl=64 time=3.198 ms
```

Рисунок 15. Эхо-запрос в TCP-моде

Далее откроем Wireshark и проанализируем эхо-запрос по протоколу TCP. 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 TCP и IP-адреса источника (192.168.1.12, то есть PC-2) и получателя (192.168.1.11, то есть PC-2).

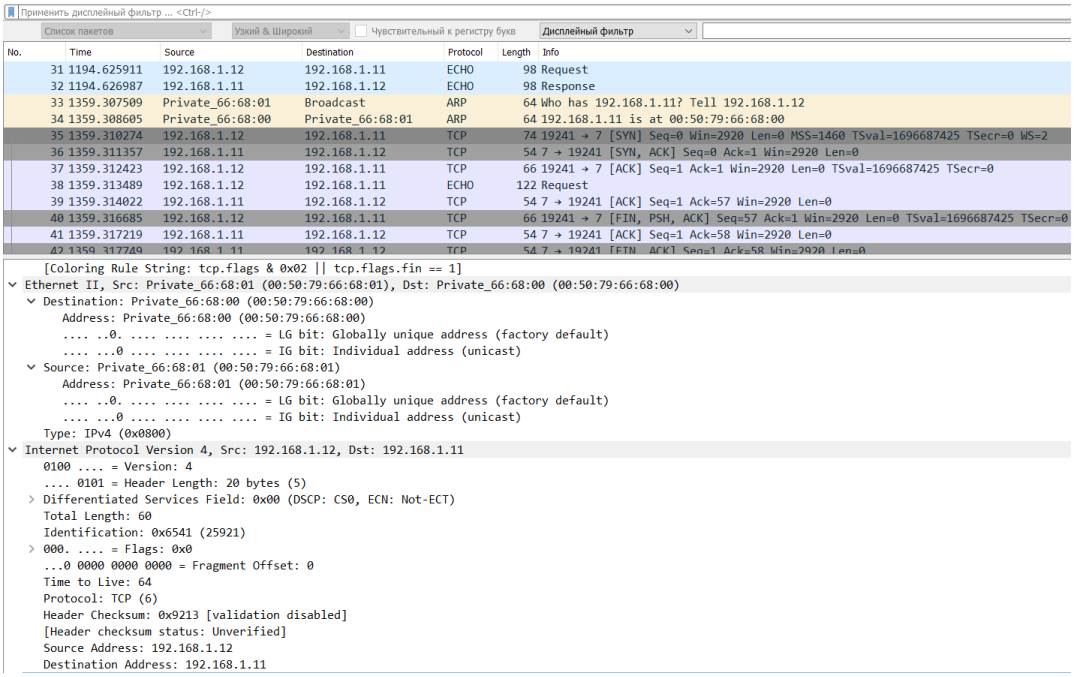

Рисунок 16. Полученная информация по эхо-запросу в TCP-моде к узлу PC-1

В поле протокола TCP можем узнать порты источника (19241) и получателя (7). А также посмотреть, как работает handshake протокола TCP. На первом шаге установлен флаг SYN, а также Порядковому номеру (Sequence Number) присвоено начальное 32-битное значение ISSa (в нашем случае 1741785218).

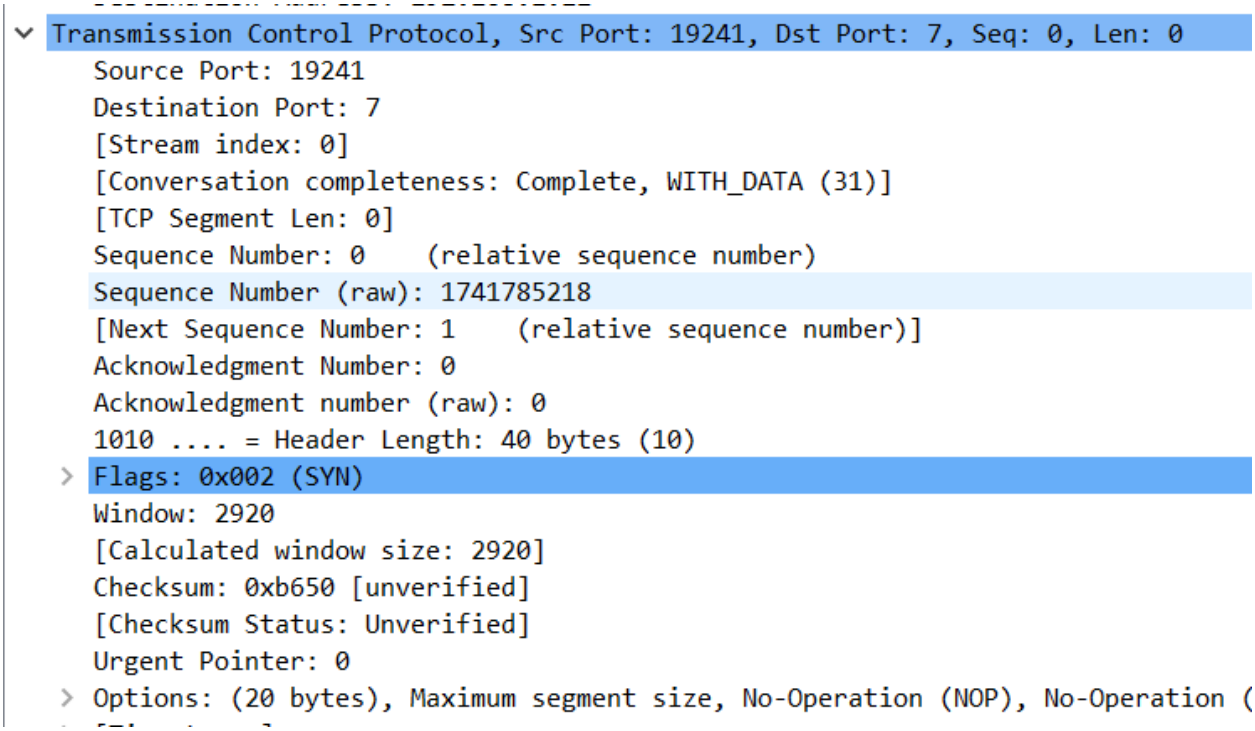

Рисунок 17. Информация по протоколу TCP

Теперь можно остановить захват трафика в Wireshark.

№3

Теперь нам нужно построить в GNS3 топологию сети, состоящей из маршрутизатора FRR, коммутатора Ethernet и оконечного устройства.

К сожалению, у меня возникают проблемы после запуска терминала маршрутизатора FRR, и нет возможности вводить команды. Видимо это из-за того, что у меня операционная система Windows. Поэтому выполнить задание №3 я не могу.

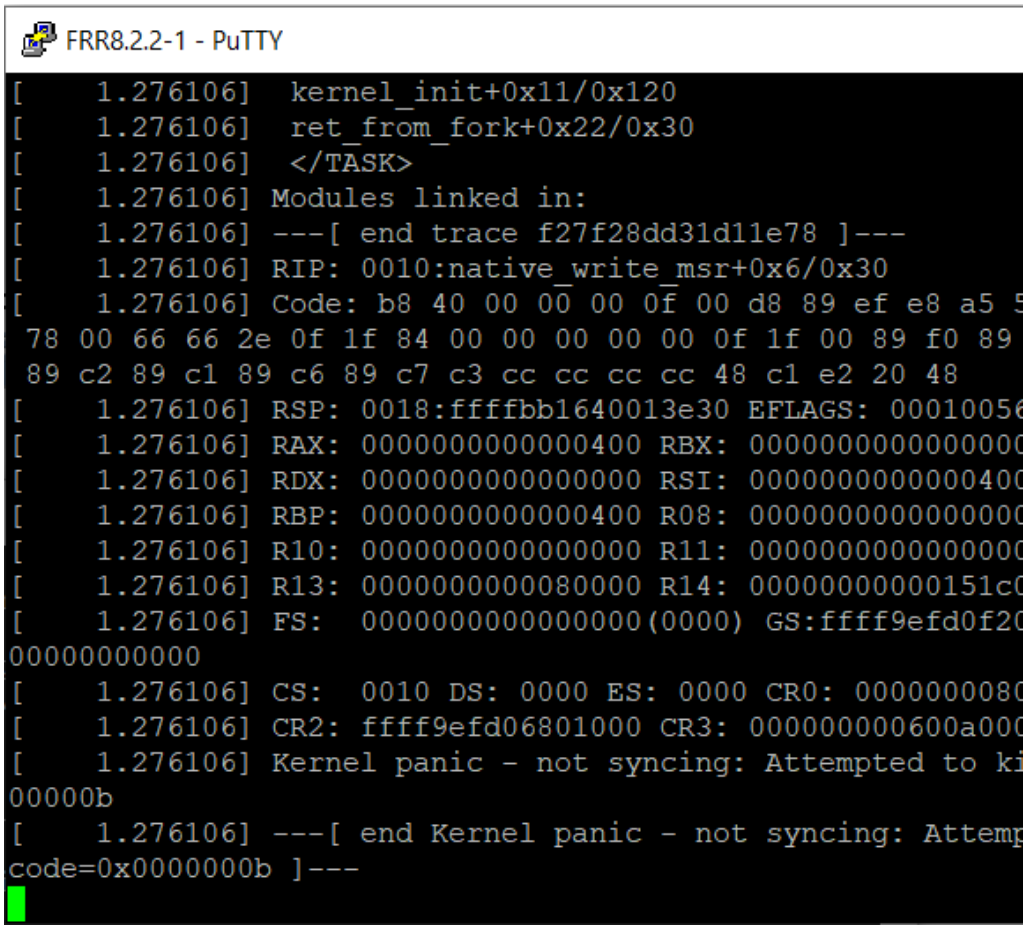

Рисунок 18. Терминал маршрутизатора FRR

№4

Запустим GNS3 VM и GNS3. Создадим новый проект.

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

Изменим отображаемые названия устройств. Коммутатору присвоим название msk-dmbelicheva-sw-01, маршрутизатору — по принципу msk-

dmbelicheva-gw-01, VPCS — по принципу PC1-dmbelicheva.

Включим захват трафика на соединении между коммутатором и маршрутизатором (появится значок лупы).

Запустим все устройства проекта и откроем консоль всех устройств проекта. Откроем окно терминала PC-1 и настроим IP-адресацию для интерфейса этого узла.

```
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
```

Рисунок 19. Настройка IP-адресации для интерфейса узла PC-1

Настроим маршрутизатор VyOS:

Во-первых, после загрузки нужно ввести логин vyos и пароль vyos.

В рабочем режиме в командной строке отобразится символ $.

```
Welcome to VyOS - vyos ttyS0

vyos login: vyos
Password:
Welcome to VyOS!

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

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

VyOS is a free software distribution that includes multiple components,
you can check individual component licenses under /usr/share/doc/*/copyright
vyos@vyos:~$ configure
```

Рисунок 20. Настройка маршрутизатора VyOS

Далее надо установить систему на диск с помощью команды *install image*, но у меня эта система уже была.

Следующим шагом перейдем в режим конфигурирования с помощью команды configure. Изменим имя устройства командой *set system host-name msk-nvsakhno-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 применим 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.

```
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:0b:4c:9a:00:00
 }
 ethernet eth1 {
     hw-id 0c:0b:4c:9a:00:01
 }
 ethernet eth2 {
     hw-id 0c:0b:4c:9a:00:02
 }
 loopback lo {
 }
[edit]
vyos@vyos# exit
exit
```

Рисунок 21. Режим конфигурирования

Теперь проверим подключение. Узел PC1 должен успешно отправлять эхо-запросы на адрес маршрутизатора 192.168.1.1. Для проверки пингуем

маршрутизатор. Получили эхо-ответ (4 пакета).

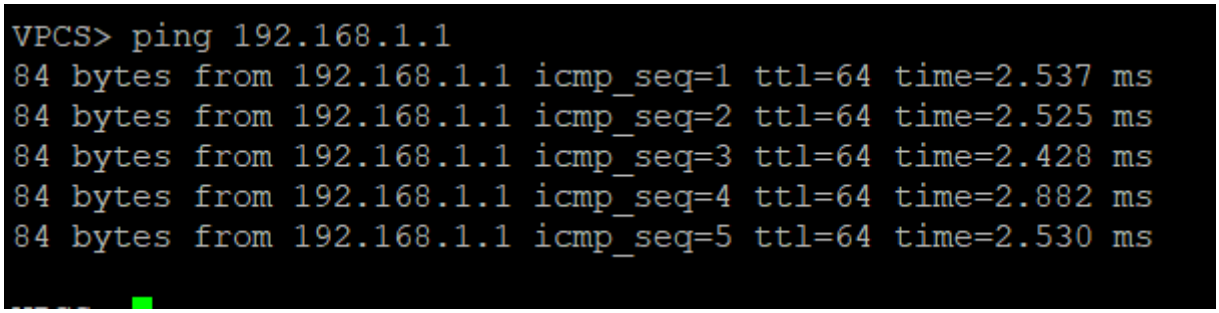

Рисунок 22. Пингование маршрутизатора

В окне Wireshark проанализируем полученную информацию. 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).

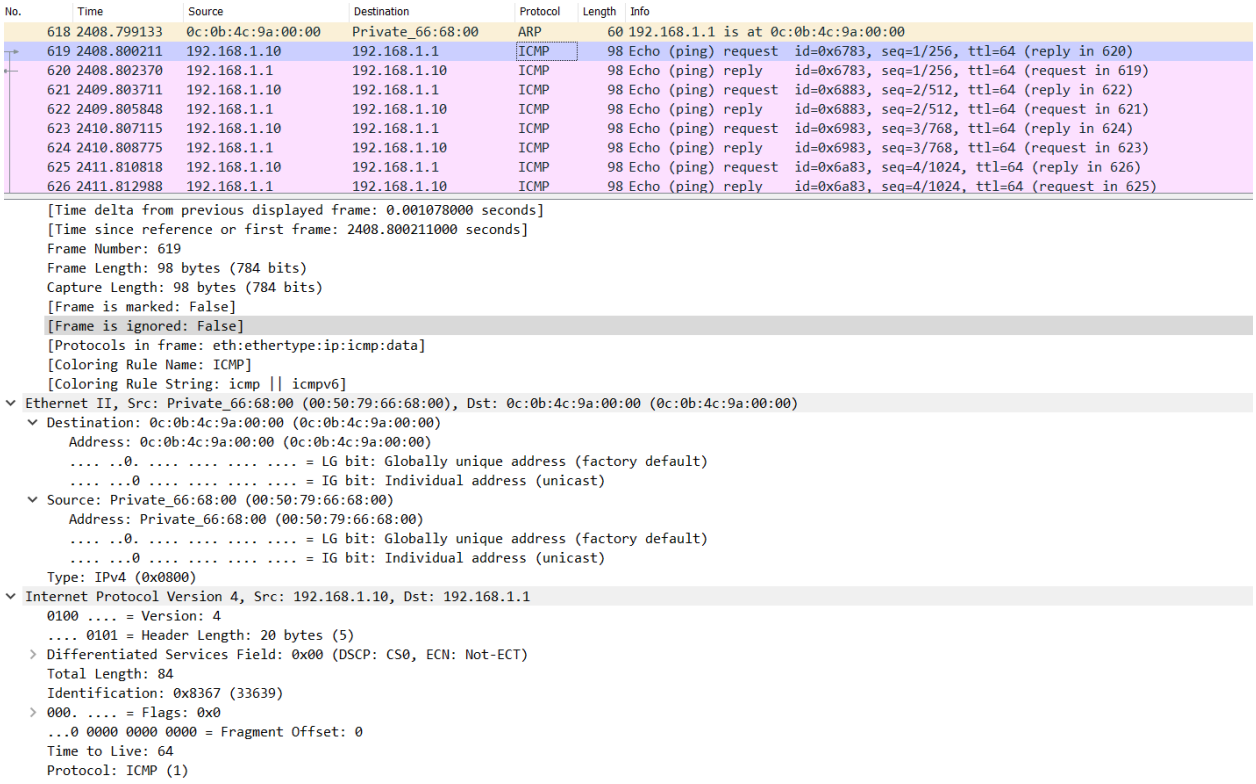

Рисунок 23. Полученная информация в Wireshark по ICMP-сообщению

Выводы:

В процессе выполнения данной лабораторной работы я построил простейшие модели сети на базе коммутатора и маршрутизатора VyOS в GNS3, проанализировал трафик посредством Wireshark.