

Отчёт по лабораторной работе №5

Дисциплина: Сетевые технологии

Ибрахим Мухсейн Алькамаль

Содержание

1 Цель работы	5
2 Анализ трафика в GNS3 посредством Wireshark	6
3 Моделирование простейшей сети на базе маршрутизатора FRR в GNS3	17
4 Моделирование простейшей сети на базе маршрутизатора VyOS в GNS3	21
5 Выводы	26

Список иллюстраций

2.1	Топология простейшей сети в GNS3	6
2.2	Задание IP-адреса для PC-1	7
2.3	Задание IP-адреса для PC-2	7
2.4	Пингование PC-2	8
2.5	Остановка всех узлов	8
2.6	Захват трафика, старт узлов	8
2.7	Информация по протоколу ARP	9
2.8	Эхо-запрос в ICMP-моде	9
2.9	Полученная информация по эхо-запросу в ICMP-моде к узлу PC-1 . .	10
2.10	Эхо-ответ	11
2.11	Эхо-запрос в UDP-моде	11
2.12	Полученная информация по эхо-запросу в UDP-моде к узлу PC-1 . .	12
2.13	Эхо-ответ	12
2.14	Эхо-запрос в TCP-моде	13
2.15	Полученная информация по эхо-запросу в TCP-моде к узлу PC-1 . .	14
2.16	Полученная информация по эхо-запросу в TCP-моде к узлу PC-1 . .	15
2.17	Полученная информация по эхо-запросу в TCP-моде к узлу PC-1 . .	16
3.1	Топология сети с маршрутизатором FRR	17
3.2	Настройка IP-адресации для интерфейса узла PC-1	18
3.3	Настройка IP-адресации для интерфейса локальной сети маршрутизатора. Проверка конфигурации.	19
3.4	Отправка эхо-запросов с узла PC1 на адрес маршрутизатора	19
3.5	Полученная информация в Wireshark по ICMP-сообщениям	20
4.1	Топология сети с маршрутизатором VyOS	21
4.2	Настройка IP-адресации для интерфейса узла PC-1	22
4.3	Вход в систему VyOS	22
4.4	Режим конфигурирования VyOS: настройка интерфейса eth0, применение и сохранение конфигурации	23
4.5	Режим конфигурирования VyOS: настройка интерфейса eth0, применение и сохранение конфигурации	24
4.6	Пингование маршрутизатора	24
4.7	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 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-alkamal-sw-01. Соединим VPCS с коммутатором и отобразим обозначение интерфейсов соединения (рис. 2.1).



Рисунок 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 команду: `ip 192.168.1.11/24 192.168.1.1`. Для сохранения конфигурации введем команду `save` (рис. 2.2).

```

Executing the startup file

PC1> /?

?
! COMMAND [ARG ...]      Print help
arp                         Invoke an OS COMMAND with optional ARG(s)
clear [ARG]                 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)
help [TEXT]                 Display TEXT in output. See also set echo ?
history                     Shortcut for: show history. List the command history
ip ARG ... [OPTION]         Configure the current VPC's IP settings. See ip ?
load [FILENAME]             Load the configuration/script from the file FILENAME
ping HOST [OPTION ...]     Ping HOST with ICMP (default) or TCP/UDP. See ping ?
quit                        Quit program
relay ARG ...                Configure packet relay between UDP ports. See relay ?
rlogin [ip] port             Telnet to port on host at ip (relative to host PC)
save [FILENAME]              Save the configuration to the file FILENAME
set ARG ...                 Set VPC name and other options. Try set ?
show [ARG ...]               Print the information of VPCs (default). See show ?
sleep [seconds] [TEXT]       Print TEXT and pause running script for seconds
trace HOST [OPTION ...]    Print the path packets take to network HOST
version                     Shortcut for: show version

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

PC1> 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

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

PC1>

```

Рисунок 2.2: Задание IP-адреса для PC-1

Аналогичным образом зададим IP-адрес 192.168.1.12 для PC-2. После этого выполним проверку доступности PC-1 с помощью команды ping. Получаем эхо-ответ от PC-1, что подтверждает корректность настройки сети (рис. 2.3).

```

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> save
Saving startup configuration to startup.vpc
. done

PC2> ping 192.168.1.11
84 bytes from 192.168.1.11 icmp_seq=1 ttl=64 time=0.330 ms
84 bytes from 192.168.1.11 icmp_seq=2 ttl=64 time=1.052 ms
84 bytes from 192.168.1.11 icmp_seq=3 ttl=64 time=0.367 ms
84 bytes from 192.168.1.11 icmp_seq=4 ttl=64 time=0.294 ms
84 bytes from 192.168.1.11 icmp_seq=5 ttl=64 time=0.297 ms

```

Рисунок 2.3: Задание IP-адреса для PC-2

Проверим работоспособность соединения между PC-1 и PC-2. В терминале PC-1 выполним команду ping 192.168.1.12. Получаем эхо-ответ от PC-2, возвращены 5 ICMP-пакетов, что свидетельствует о корректной передаче данных между узлами (рис. 2.4).

```

PC1> ping 192.168.1.12
84 bytes from 192.168.1.12 icmp_seq=1 ttl=64 time=0.305 ms
84 bytes from 192.168.1.12 icmp_seq=2 ttl=64 time=0.950 ms
84 bytes from 192.168.1.12 icmp_seq=3 ttl=64 time=0.493 ms
84 bytes from 192.168.1.12 icmp_seq=4 ttl=64 time=0.614 ms
84 bytes from 192.168.1.12 icmp_seq=5 ttl=64 time=0.756 ms

```

Рисунок 2.4: Пингование PC-2

Остановим в проекте все узлы (рис. 2.5).



Рисунок 2.5: Остановка всех узлов

Запустим анализатор трафика на соединении между PC-1 и коммутатором. Для этого щёлкнем правой кнопкой мыши по соединению и выберем пункт **Start capture**. После запуска Wireshark в проекте GNS3 на линии соединения появляется значок захвата. Затем запустим все узлы проекта (рис. 2.6).



Рисунок 2.6: Захват трафика, старт узлов

В окне Wireshark отображаются ARP-пакеты (рис. 2.7). В поле физического уровня видно, что длина кадра составляет **64 байта (512 бит)**. На канальном уровне отображаются MAC-адреса источника и назначения: адрес назначения

является **широковещательным (ff:ff:ff:ff:ff:ff)**, адрес источника — **глобально уникальным (factory default)** и индивидуальным. Тип передаваемого пакета — **gratuitous ARP**, что соответствует процессу объявления IP-адресов узлами при запуске.

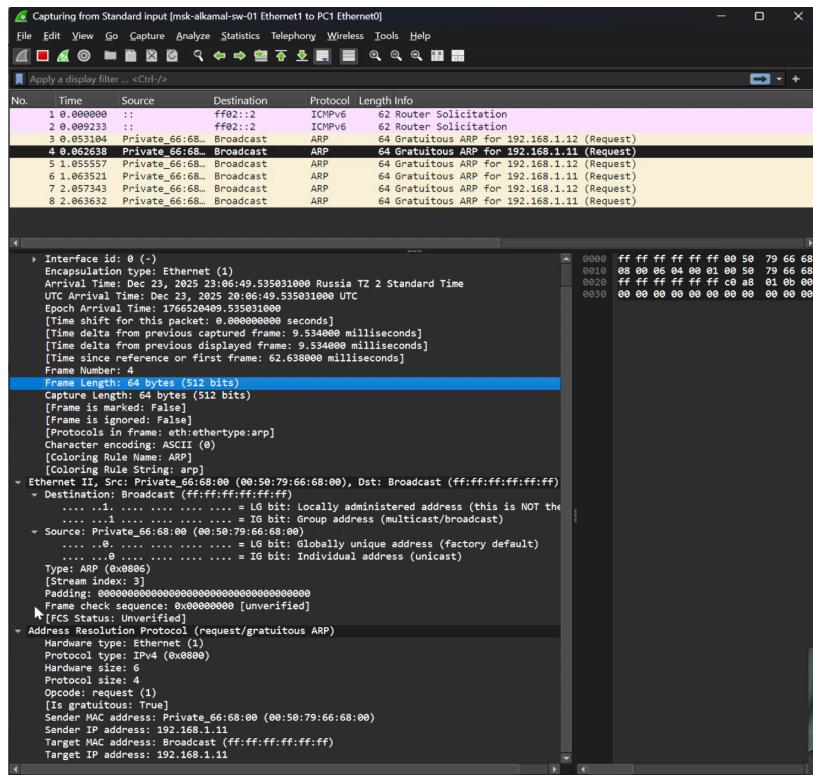


Рисунок 2.7: Информация по протоколу ARP

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

```
PC2> ping 192.168.1.11 -1
84 bytes from 192.168.1.11 icmp_seq=1 ttl=64 time=0.428 ms
84 bytes from 192.168.1.11 icmp_seq=2 ttl=64 time=0.596 ms
84 bytes from 192.168.1.11 icmp_seq=3 ttl=64 time=0.516 ms
84 bytes from 192.168.1.11 icmp_seq=4 ttl=64 time=0.373 ms
84 bytes from 192.168.1.11 icmp_seq=5 ttl=64 time=0.466 ms
```

Рисунок 2.8: Эхо-запрос в ICMP-моде

Далее откроем Wireshark и проанализируем эхо-запрос по протоколу ICMP (рис. 2.9) и (рис. 2.10). На канальном уровне видны MAC-адреса источника и получателя, которые являются **глобально уникальными и индивидуальными (unicast)**, что определяется по нулевым значениям LG и IG битов. На сетевом уровне указан протокол ICMP, а также IP-адрес источника **192.168.1.12 (PC-2)** и адрес назначения **192.168.1.11 (PC-1)**.

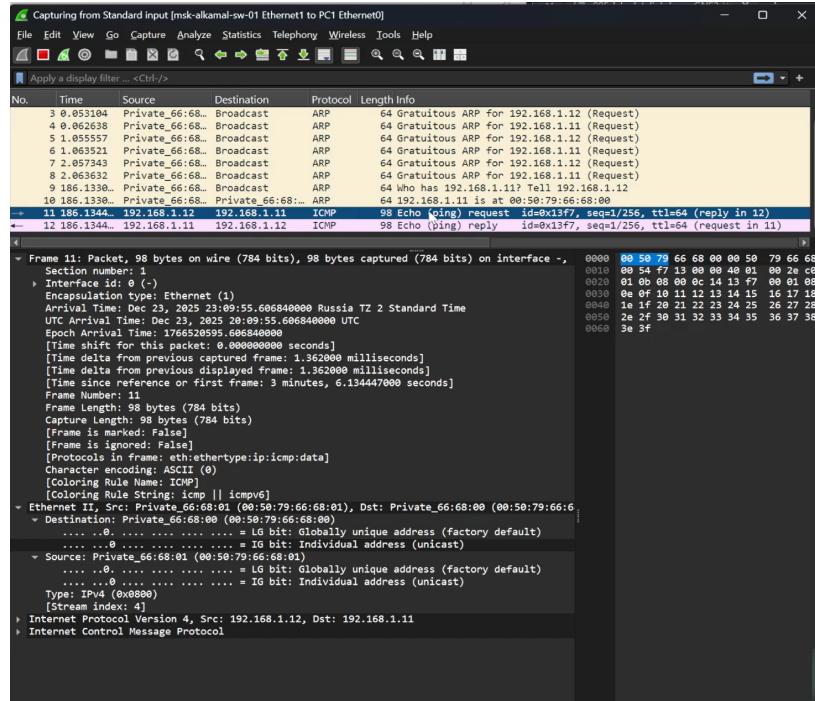


Рисунок 2.9: Полученная информация по эхо-запросу в ICMP-моде к узлу PC-1

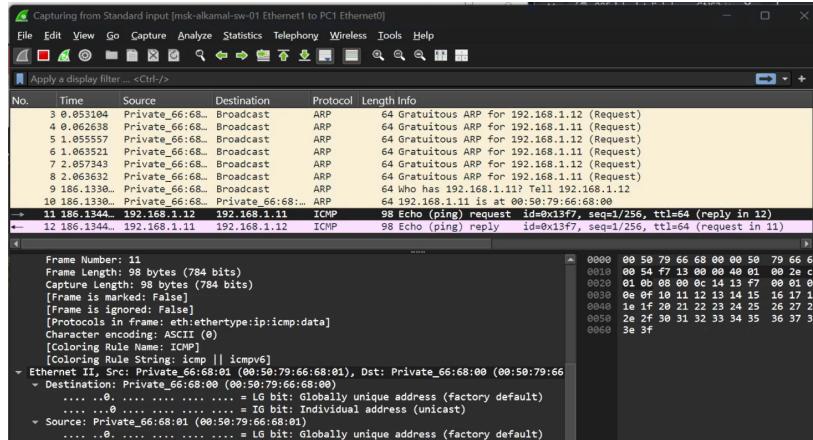


Рисунок 2.10: Эхо-ответ

Далее выполним эхо-запрос в UDP-режиме к узлу PC-1, используя опцию -2 (рис. 2.11).

```
PC2> ping 192.168.1.11 -2 -c
84 bytes from 192.168.1.11 udp seq=1 ttl=64 time=0.597 ms
84 bytes from 192.168.1.11 udp seq=2 ttl=64 time=0.308 ms
84 bytes from 192.168.1.11 udp seq=3 ttl=64 time=0.330 ms
84 bytes from 192.168.1.11 udp seq=4 ttl=64 time=0.539 ms
```

Рисунок 2.11: Эхо-запрос в UDP-моде

В Wireshark проанализируем переданные пакеты UDP (рис. 2.12) и (рис. 2.13). На канальном уровне MAC-адреса источника и получателя являются **глобально уникальными и индивидуальными**. На сетевом уровне используется протокол UDP, IP-адрес источника – **192.168.1.12 (PC-2)**, IP-адрес назначения – **192.168.1.11 (PC-1)**. На транспортном уровне указаны UDP-порты: порт источника – **19241**, порт назначения – **7 (Echo)**.

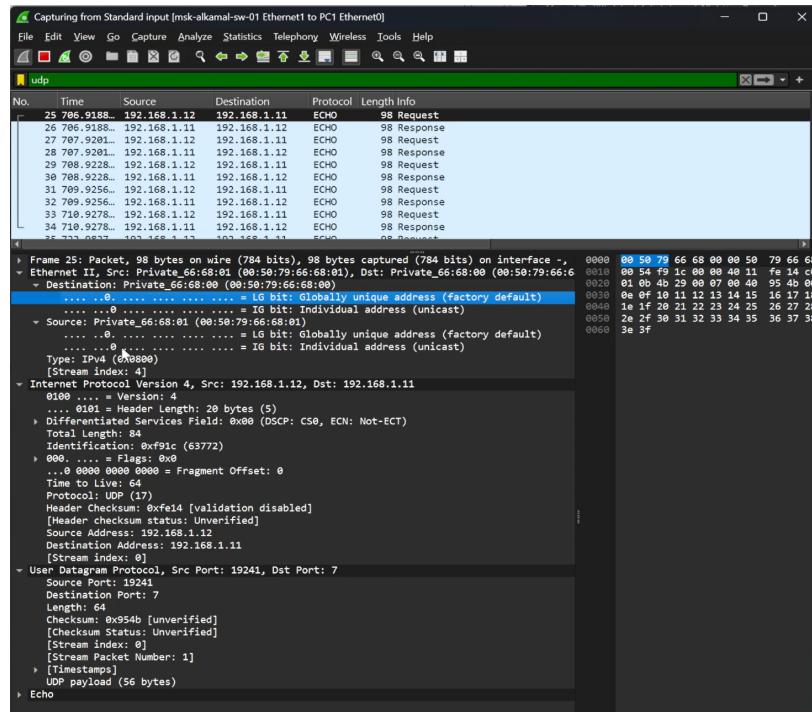


Рисунок 2.12: Полученная информация по эхо-запросу в UDP-режиме к узлу PC-1

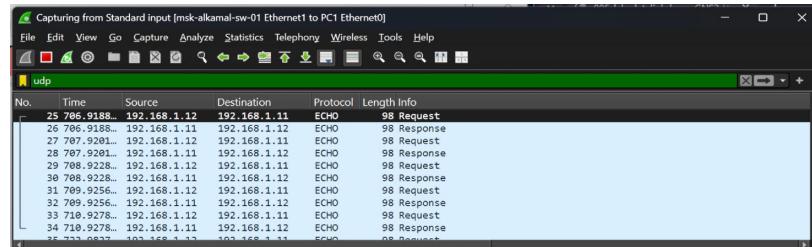


Рисунок 2.13: Эхо-ответ

Затем выполним эхо-запрос в TCP-режиме к узлу PC-1, используя опцию -3.

В результате устанавливается TCP-соединение, передаются данные и соединение корректно закрывается, что подтверждается выводом терминала PC-2 (рис. 2.14).

```
PC2> ping 192.168.1.11 -3 -c 4
Connect 7@192.168.1.11 seq=1 ttl=64 time=1.973 ms
SendData 7@192.168.1.11 seq=1 ttl=64 time=2.021 ms
Close 7@192.168.1.11 seq=1 ttl=64 time=2.955 ms
Connect 7@192.168.1.11 seq=2 ttl=64 time=1.911 ms
SendData 7@192.168.1.11 seq=2 ttl=64 time=2.191 ms
Close 7@192.168.1.11 seq=2 ttl=64 time=4.465 ms
Connect 7@192.168.1.11 seq=3 ttl=64 time=2.088 ms
SendData 7@192.168.1.11 seq=3 ttl=64 time=2.728 ms
Close 7@192.168.1.11 seq=3 ttl=64 time=3.902 ms
Connect 7@192.168.1.11 seq=4 ttl=64 time=1.994 ms
SendData 7@192.168.1.11 seq=4 ttl=64 time=1.944 ms
Close 7@192.168.1.11 seq=4 ttl=64 time=4.439 ms
PC2>
```

Рисунок 2.14: Эхо-запрос в TCP-моде

Далее откроем Wireshark и проанализируем эхо-запрос по протоколу TCP. На канальном уровне отображаются MAC-адреса источника и получателя, которые являются **глобально уникальными и индивидуальными (unicast)**, что определяется по нулевым значениям LG и IG битов. На сетевом уровне указан протокол TCP, а также IP-адрес источника **192.168.1.12 (PC-2)** и IP-адрес назначения **192.168.1.11 (PC-1)**.

На транспортном уровне TCP видны порты: порт источника – **28458**, порт назначения – **7 (Echo)**. Также можно проследить процесс установки соединения по механизму **трёхэтапного рукопожатия (three-way handshake)**. На первом этапе передаётся сегмент с установленным флагом **SYN** (рис. 2.15), при этом порядковому номеру (Sequence Number) присваивается начальное значение **ISS**, которое в данном случае определяется автоматически стеком TCP (относительное значение Seq = 0).

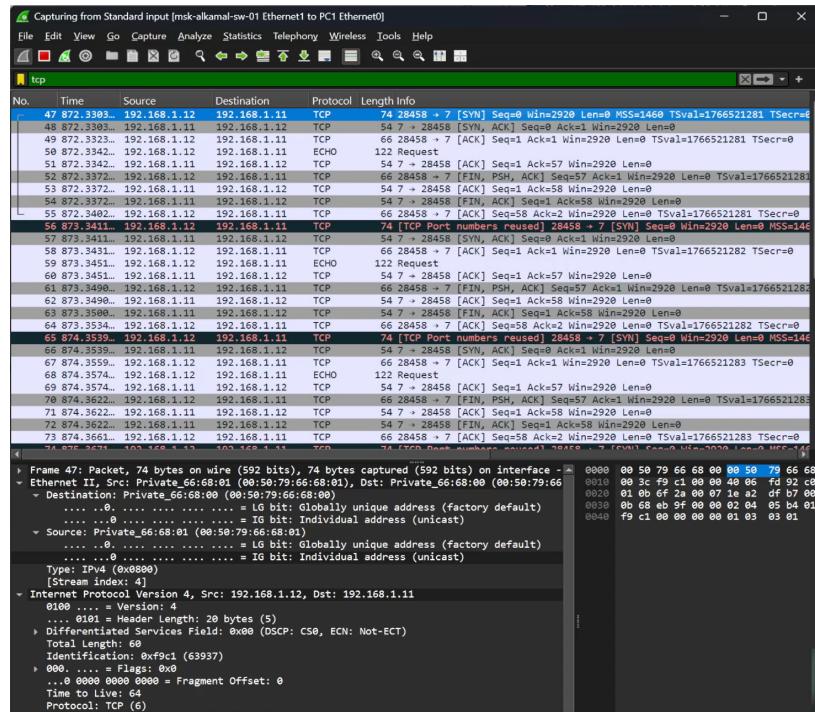


Рисунок 2.15: Полученная информация по эхо-запросу в TCP-моде к узлу PC-1

На втором этапе принимающая сторона отвечает сегментом с установленными флагами **SYN** и **ACK**, подтверждая получение запроса и предлагая свои параметры соединения (рис. 2.16).

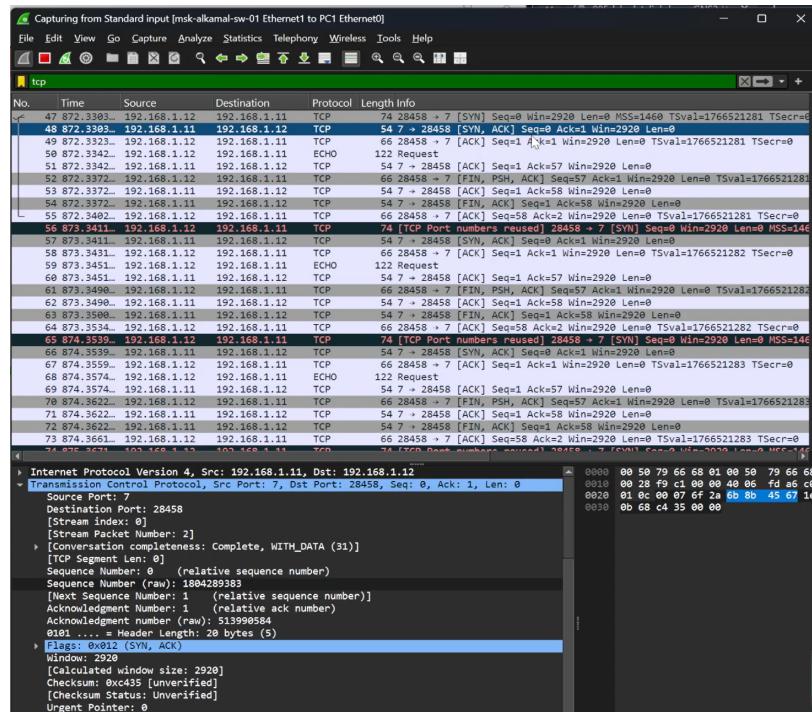


Рисунок 2.16: Полученная информация по эхо-запросу в TCP-моде к узлу PC-1

На третьем этапе инициатор соединения отправляет сегмент с установленным флагом **ACK**, тем самым подтверждая получение ответа и завершая установление TCP-соединения (рис. 2.17).

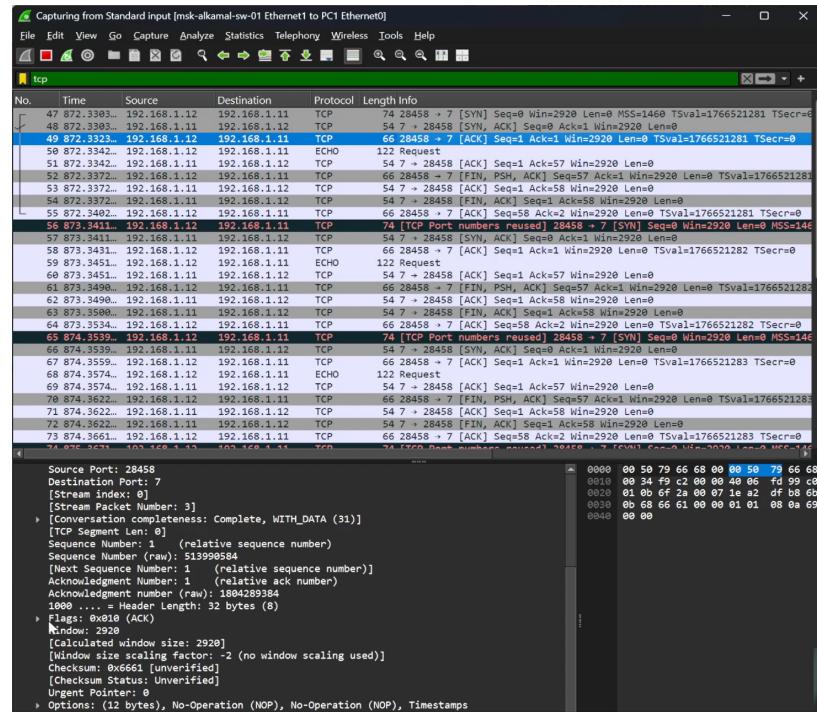


Рисунок 2.17: Полученная информация по эхо-запросу в TCP-моде к узлу PC-1

После завершения анализа можно остановить захват трафика в Wireshark.

3 Моделирование простейшей сети на базе маршрутизатора FRR в GNS3

Теперь нам необходимо построить в GNS3 топологию сети (рис. 3.1), состоящей из маршрутизатора FRR, коммутатора Ethernet и оконечного устройства. Изменим отображаемые названия устройств. Коммутатору присвоим имя `msk-alkamal-sw-01`, маршрутизатору — `msk-alkamal-gw-01`, оконечному устройству VPCS — `PC1-alkamal`. Включим захват трафика на соединении между коммутатором и маршрутизатором. После этого запустим все устройства проекта и откроем их консоли.

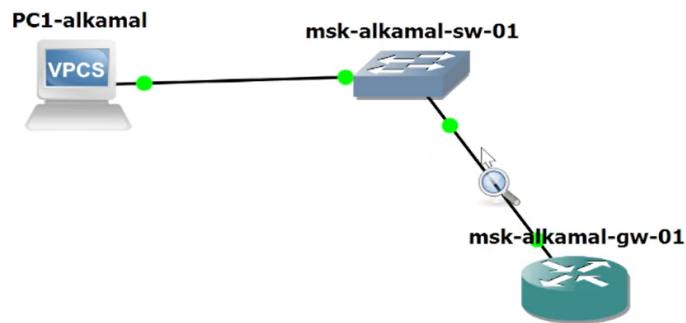
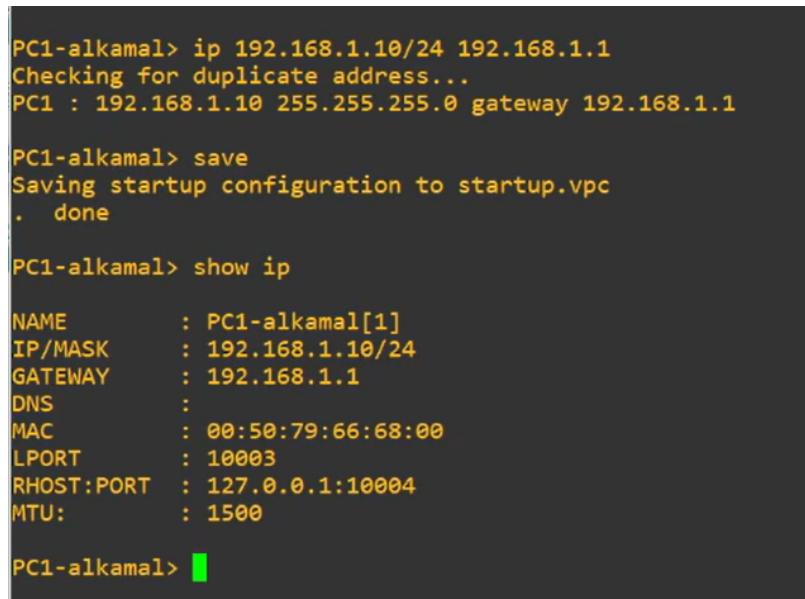


Рисунок 3.1: Топология сети с маршрутизатором FRR

Настроим IP-адресацию для интерфейса узла PC1 (рис. 3.2). В терминале VPCS выполним следующие команды:

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



```
PC1-alkamal> 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  
  
PC1-alkamal> save  
Saving startup configuration to startup.vpc  
. done  
  
PC1-alkamal> show ip  
  
NAME      : PC1-alkamal[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  
  
PC1-alkamal>
```

Рисунок 3.2: Настройка IP-адресации для интерфейса узла PC-1

Далее настроим IP-адресацию для интерфейса локальной сети маршрутизатора FRR. Проверим корректность конфигурации маршрутизатора и назначенных IP-адресов (рис. 3.3).

```

[OK]
msk-alkamal-gw-01# configure terminal
msk-alkamal-gw-01(config)# interface eth0
msk-alkamal-gw-01(config-if)# ip address 192.168.1.1/24
msk-alkamal-gw-01(config-if)# no shutdown
msk-alkamal-gw-01(config-if)# exit
msk-alkamal-gw-01(config)# exit
msk-alkamal-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-alkamal-gw-01# show running-config
Building configuration...

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

```

Рисунок 3.3: Настройка IP-адресации для интерфейса локальной сети маршрутизатора. Проверка конфигурации.

Проверим работоспособность соединения. Узел PC1 успешно отправляет ICMP эхо-запросы на IP-адрес маршрутизатора **192.168.1.1**, получая эхо-ответы, что подтверждает корректную настройку сети (рис. 3.4).

```

PC1-alkamal> ping 192.168.1.1
84 bytes from 192.168.1.1 icmp_seq=1 ttl=64 time=3.894 ms
84 bytes from 192.168.1.1 icmp_seq=2 ttl=64 time=1.498 ms
84 bytes from 192.168.1.1 icmp_seq=3 ttl=64 time=1.689 ms
84 bytes from 192.168.1.1 icmp_seq=4 ttl=64 time=1.949 ms
84 bytes from 192.168.1.1 icmp_seq=5 ttl=64 time=1.852 ms

```

Рисунок 3.4: Отправка эхо-запросов с узла PC1 на адрес маршрутизатора

В окне Wireshark проанализируем захваченный трафик (рис. 3.5). На канальном уровне отображаются MAC-адреса источника и получателя, которые являются **глобально уникальными и индивидуальными (unicast)**, что определяется по

нулевым значениям LG и IG битов. На сетевом уровне используется протокол ICMP, при этом IP-адрес источника — **192.168.1.10 (PC-1)**, IP-адрес назначения — **192.168.1.1 (маршрутизатор FRR)**.

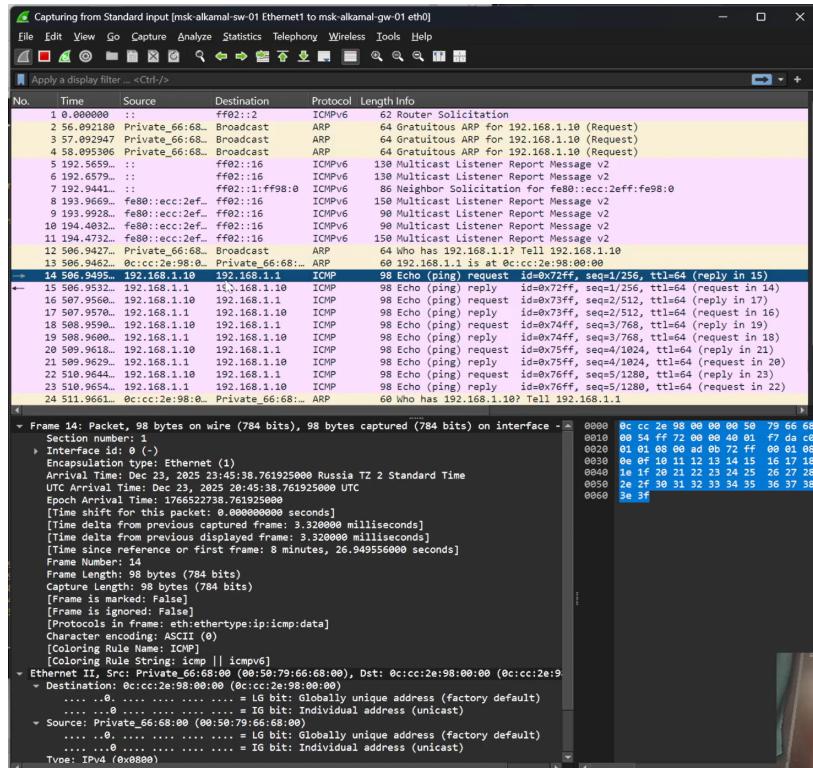


Рисунок 3.5: Полученная информация в Wireshark по ICMP-сообщениям

После завершения анализа остановим захват пакетов в Wireshark и выключим все устройства проекта.

4 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-alkamal-sw-01, маршрутизатору – по принципу msk-alkamal-gw-01, VPCS – по принципу PC1-alkamal. Включим захват трафика на соединении между коммутатором и маршрутизатором (рис. 4.1) (появится значок лупы). Запустим все устройства проекта и откроем консоль всех устройств проекта.

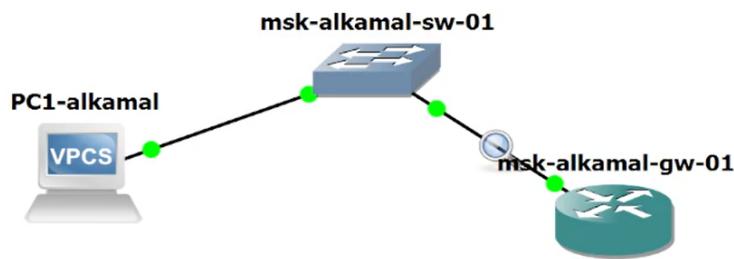
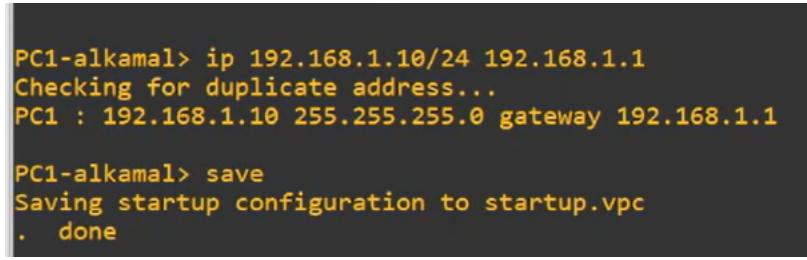


Рисунок 4.1: Топология сети с маршрутизатором VyOS

Откроем окно терминала PC-1 и настроим IP-адресацию для интерфейса этого

узла (рис. 4.2).



```
PC1-alkamal> 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

PC1-alkamal> save
Saving startup configuration to startup.vpc
. done
```

Рисунок 4.2: Настройка IP-адресации для интерфейса узла PC-1

Настроим маршрутизатор VyOS (рис. 4.3). После загрузки маршрутизатора в консоли отображается процесс инициализации системы, после чего появляется приглашение входа. Выполним вход в систему, указав логин `vyos` и пароль `vyos`. После успешной аутентификации открывается рабочий режим командной строки с приглашением `$`.

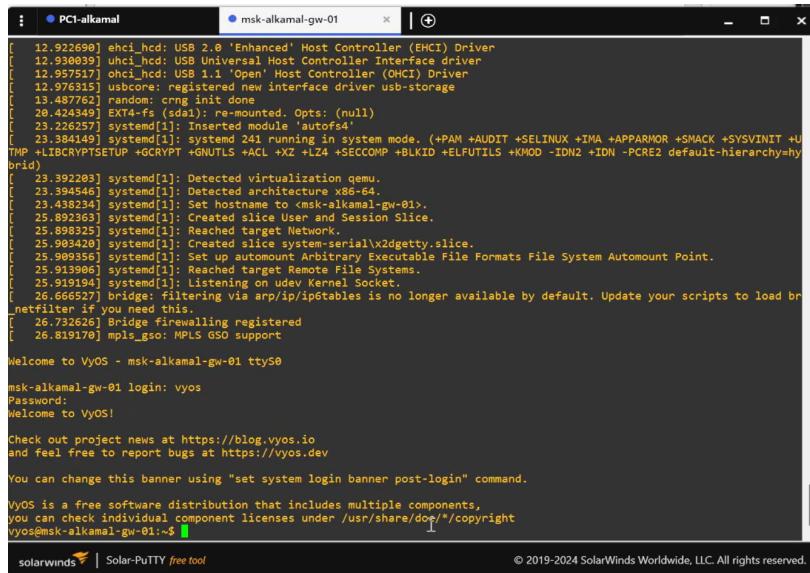


Рисунок 4.3: Вход в систему VyOS

Далее перейдём в режим конфигурирования с помощью команды `configure`. При попытке задать IP-адрес интерфейсу `eth0` вне режима конфигурирования система возвращает сообщение об ошибке, после чего настройка выполняется корректно в режиме `[edit]`. Назначим IP-адрес `192.168.1.1/24` интерфейсу

`eth0` маршрутизатора. В процессе применения конфигурации возникает сообщение о невозможности одновременного использования статического IPv4-адреса и DHCP на одном интерфейсе, поэтому предварительно удаляется DHCP-настройка.

После этого изменения успешно применяются с помощью команды `commit` и сохраняются командой `save` (рис. 4.4). Для проверки конфигурации выполним команду `show interfaces`, которая подтверждает назначение IP-адреса интерфейсу `eth0`. Завершаем настройку, оставаясь в режиме конфигурирования (рис. 4.5).

The screenshot shows a Solar-PuTTY terminal window titled "msk-alkamal-gw-01". The command history is as follows:

```
vyos@msk-alkamal-gw-01:~$ set interfaces ethernet eth0 address 192.168.1.1/24
  Invalid command: set [interfaces]
vyos@msk-alkamal-gw-01:~$ configure
[edit]
vyos@msk-alkamal-gw-01# set interfaces ethernet eth0 address 192.168.1.1/24
> `c
[edit]
vyos@msk-alkamal-gw-01# set interfaces ethernet eth0 address 192.168.1.1/24
[edit]
vyos@msk-alkamal-gw-01# compare
[edit interfaces ethernet eth0]
+address 192.168.1.1/24
[edit]
vyos@msk-alkamal-gw-01# commit
[interfaces ethernet eth0]
Can't configure both static IPv4 and DHCP address on the same interface
[[interfaces ethernet eth0]] failed
Commit failed
[edit]
vyos@msk-alkamal-gw-01#
[edit]
vyos@msk-alkamal-gw-01# delete interfaces ethernet eth0 address dhcp
[edit]
vyos@msk-alkamal-gw-01# set interfaces ethernet eth0 address 192.168.1.1/24
  Configuration path: [interfaces ethernet eth0 address 192.168.1.1/24] already exists
[edit]
vyos@msk-alkamal-gw-01# commit
[edit]
vyos@msk-alkamal-gw-01# save
Saving configuration to '/config/config.boot'...
Done
[edit]
vyos@msk-alkamal-gw-01#
```

Рисунок 4.4: Режим конфигурирования VyOS: настройка интерфейса `eth0`, применение и сохранение конфигурации

```

[edit]
vyos@msk-alkamal-gw-01# commit
[ interfaces ethernet eth0 ]
Can't configure both static IPv4 and DHCP address on the same interface

[[interfaces ethernet eth0]] failed
Commit failed
[edit]
vyos@msk-alkamal-gw-01#
[edit]
vyos@msk-alkamal-gw-01# delete interfaces ethernet eth0 address dhcp
[edit]
vyos@msk-alkamal-gw-01# set interfaces ethernet eth0 address 192.168.1.1/24

    Configuration path: [interfaces ethernet eth0 address 192.168.1.1/24] already exists

[edit]
vyos@msk-alkamal-gw-01# commit
[edit]
vyos@msk-alkamal-gw-01# save
Saving configuration to '/config/config.boot'...
Done
[edit]
vyos@msk-alkamal-gw-01# show interfaces
ethernet eth0 {
    address 192.168.1.1/24
    hw-id 0c:d4:4a:23:00:00
}
ethernet eth1 {
    hw-id 0c:d4:4a:23:00:01
}
ethernet eth2 {
    hw-id 0c:d4:4a:23:00:02
}
loopback lo {
}
[edit]
vyos@msk-alkamal-gw-01#

```

© 2019-2024 SolarWinds Worldwide, LLC. All rights reserved.

Рисунок 4.5: Режим конфигурирования VyOS: настройка интерфейса eth0, применение и сохранение конфигурации

Теперь проверим подключение. Узел **PC1** успешно отправляет ICMP эхо-запросы на IP-адрес маршрутизатора **192.168.1.1**. Для проверки выполним пинг маршрутизатора (рис. 4.6). В результате получены эхо-ответы, что подтверждается возвратом **5 ICMP-пакетов**.

```

PC1-alkamal> ping 192.168.1.1
84 bytes from 192.168.1.1 icmp_seq=1 ttl=64 time=2.060 ms
84 bytes from 192.168.1.1 icmp_seq=2 ttl=64 time=2.220 ms
84 bytes from 192.168.1.1 icmp_seq=3 ttl=64 time=1.948 ms
84 bytes from 192.168.1.1 icmp_seq=4 ttl=64 time=1.799 ms
84 bytes from 192.168.1.1 icmp_seq=5 ttl=64 time=1.374 ms

PC1-alkamal>

```

Рисунок 4.6: Пингование маршрутизатора

В окне Wireshark проанализируем захваченный трафик (рис. 4.7). На канальном уровне отображаются MAC-адреса источника и получателя, которые являются **глобально уникальными и индивидуальными (unicast)**, что определяется по нулевым значениям LG и IG битов. На сетевом уровне используется протокол **ICMP**, при этом IP-адрес источника — **192.168.1.10 (PC-1)**, IP-адрес назначения —

192.168.1.1 (маршрутизатор VyOS).

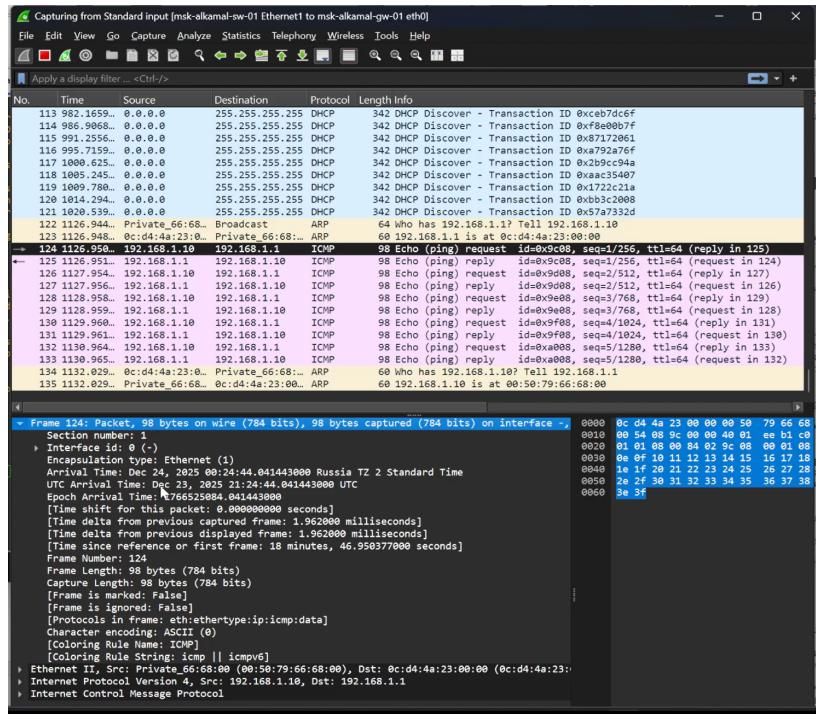


Рисунок 4.7: Полученная информация в Wireshark по ICMP-сообщению

5 Выводы

В ходе выполнения лабораторной работы были построены и настроены простейшие модели локальных сетей в среде GNS3 с использованием коммутатора Ethernet, маршрутизаторов **FRR** и **VyOS**, а также оконечных узлов VPCS. Была выполнена настройка IP-адресации сетевых интерфейсов, проверена связность узлов с помощью ICMP эхо-запросов и проанализирована корректность работы сети.

С использованием анализатора трафика **Wireshark** были исследованы протоколы **ARP**, **ICMP**, **UDP** и **TCP**, рассмотрена структура кадров и пакетов на канальном, сетевом и транспортном уровнях модели OSI, а также проанализирован процесс установления TCP-соединения. Полученные результаты подтверждают корректность выполненной конфигурации и работоспособность смоделированных сетей.