

# Содержание

0.1	Title . . . . .	3
0.2	Generic otions . . . . .	3
0.3	Pdf output format . . . . .	3
<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
3.1	Моделирование простейшей сети на базе коммутатора в GNS3 . . . . .	6
3.2	Анализ трафика в GNS3 посредством Wireshark . . . . .	8
3.3	Моделирование простейшей сети на базе маршрутизатора FRR в GNS3	13

# Список иллюстраций

3.1	Топология простейшей сети в GNS3 . . . . .	6
3.2	Просмотр синтаксиса возможных для ввода команд VPCS в GNS3 . .	6
3.3	Задание IP-адреса и сохранение конфигурации VPCS в GNS3 . . . . .	7
3.4	Задание IP-адреса и сохранение конфигурации VPCS в GNS3 . . . . .	7
3.5	Эхо-запросы . . . . .	8
3.6	Запуск анализатора трафика на соединении . . . . .	8
3.7	Анализ трафика в Wireshark . . . . .	9
3.8	Информация по опциям команды ping . . . . .	9
3.9	Эхо-запрос в ICMP-моду к узлу PC-1 . . . . .	10
3.10	Анализ трафика в Wireshark . . . . .	10
3.11	Эхо-запрос в UDP-моду к узлу PC-1 . . . . .	11
3.12	Анализ трафика в Wireshark . . . . .	11
3.13	Эхо-запрос в TCP-моду к узлу PC-1 . . . . .	12
3.14	Анализ трафика в Wireshark . . . . .	12
3.15	Топология простейшей сети с маршрутизатором в GNS3 . . . . .	13
3.16	Настройка IP-адресации . . . . .	13
3.17	Ошибка1   Невозможно использование KVM . . . . .	14
3.18	Ошибка2   Невозможно использование KVM . . . . .	14

# Список таблиц

## 0.1 Title

title: «Отчёт по лабораторной работе №5» subtitle: «Сетевые технологии» license:  
«CC BY»

## 0.2 Generic options

lang: ru-RU toc-title: «Содержание»

## 0.3 Pdf output format

toc: true # Table of contents toc-depth: 2 lof: true # List of figures lot: true # List of tables  
fontsize: 12pt linestretch: 1.5 papersize: a4 documentclass: scrreprt ## I18n polyglossia  
polyglossia-lang: name: russian polyglossia-otherlangs: name: english ## I18n babel babel-  
lang: russian babel-otherlangs: english —

# 1 Цель работы

Цель данной работы — построение простейших моделей сети на базе коммутатора и маршрутизаторов FRR и VyOS в GNS3, анализ трафика посредством Wireshark.

## 2 Задание

1. Построить в GNS3 топологию сети, состоящей из коммутатора Ethernet и двух конечных устройств (персональных компьютеров).
2. Задать конечным устройствам IP-адреса в сети 192.168.1.0/24. Проверить связь.
3. С помощью Wireshark захватить и проанализировать ARP-сообщения.
4. С помощью Wireshark захватить и проанализировать ICMP-сообщения.
5. Построить в GNS3 топологию сети, состоящей из маршрутизатора FRR, коммутатора Ethernet и конечного устройства.
6. Задать конечному устройству IP-адрес в сети 192.168.1.0/24.
7. Присвоить интерфейсу маршрутизатора адрес 192.168.1.1/24
8. Проверить связь.
9. Построить в GNS3 топологию сети, состоящей из маршрутизатора VyOS, коммутатора Ethernet и конечного устройства.
10. Задать конечному устройству IP-адрес в сети 192.168.1.0/24.
11. Присвоить интерфейсу маршрутизатора адрес 192.168.1.1/24
12. Проверить связь.

## 3 Выполнение лабораторной работы

### 3.1 Моделирование простейшей сети на базе коммутатора в GNS3

Запускаю GNS3 VM и GNS3 и создаю новый проект. В рабочей области GNS3 размещаю коммутатор Ethernet и два VPCS. Переименовываю их согласно заданию (рис. 3.1).

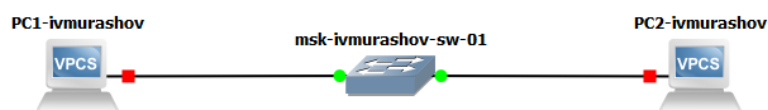


Рисунок 3.1: Топология простейшей сети в GNS3

Захожу к консоль Putty и просматриваю синтаксис возможных для ввода команд VPCS в GNS3 (рис. 3.2).

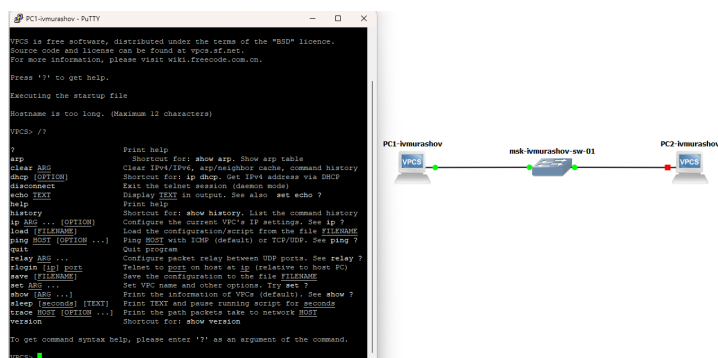
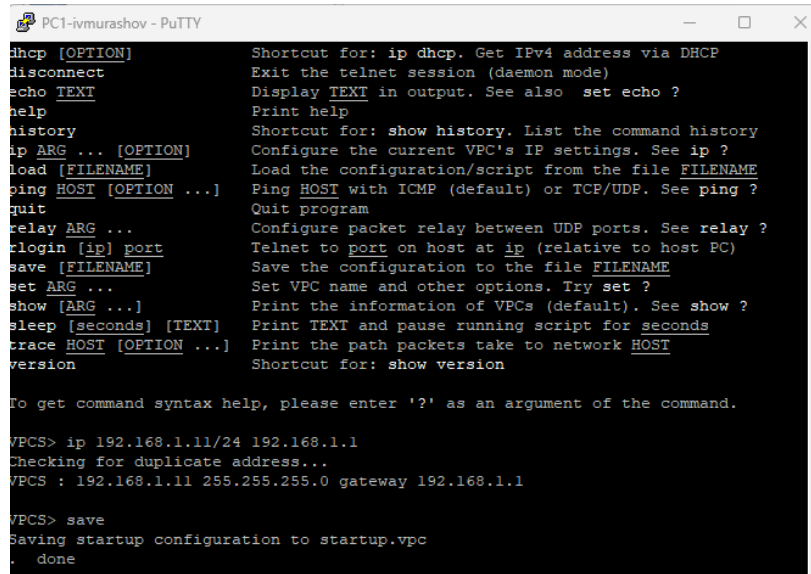


Рисунок 3.2: Просмотр синтаксиса возможных для ввода команд VPCS в GNS3

Задаю IP-адрес 192.168.1.11 в сети 192.168.1.0/24 для PC-1 (рис. 3.3).



```
PC1-ivmurashov - PuTTY
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.

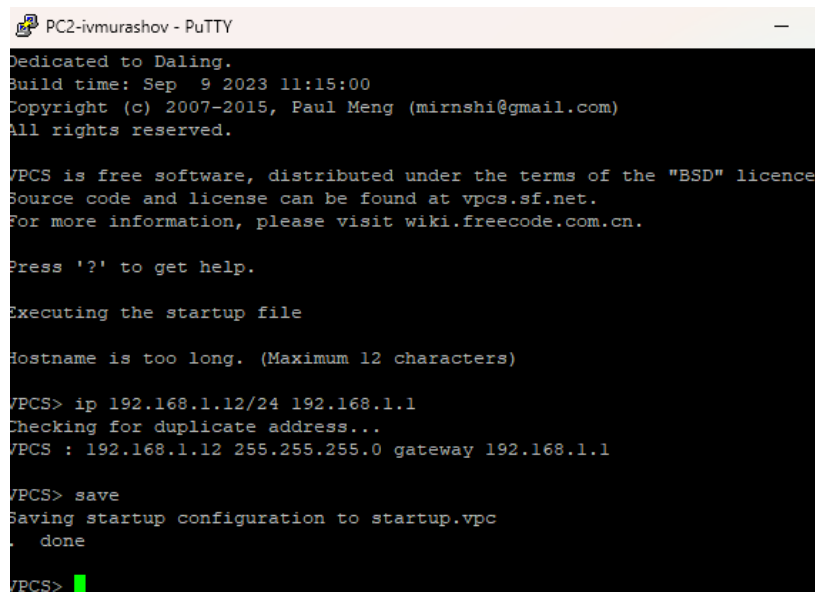
VPCS> ip 192.168.1.11/24 192.168.1.1
Checking for duplicate address...
VPCS : 192.168.1.11 255.255.255.0 gateway 192.168.1.1

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

VPCS>
```

Рисунок 3.3: Задание IP-адреса и сохранение конфигурации VPCS в GNS3

Аналогичным образом задаю IP-адрес 192.168.1.12 для PC-2 (рис. 3.4).



```
PC2-ivmurashov - PuTTY
Dedicated to Daling.
Build time: Sep  9 2023 11:15:00
Copyright (c) 2007-2015, 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...
VPCS : 192.168.1.12 255.255.255.0 gateway 192.168.1.1

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

VPCS> █
```

Рисунок 3.4: Задание IP-адреса и сохранение конфигурации VPCS в GNS3

Проверяю работоспособность соединения между PC-1 и PC-2 с помощью команды ping (рис. 3.5).

```

PC1-ivmurashov - PuTTY
show (ARG ...) Print the information of VPCS (default). See show ?
sleep (SECONDS) [TEXT] Print TEXT and pause running scripts 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 192.168.1.11/24 192.168.1.1
Checking for duplicate address...
VPCS> 192.168.1.10 255.255.255.0 gateway 192.168.1.1
VPCS> save
Saving startup configuration to startup.vpc
. done
VPCS> ping 192.168.1.12
64 bytes from 192.168.1.12 icmp_seq=1 ttl=64 time=1.057 ms
64 bytes from 192.168.1.12 icmp_seq=2 ttl=64 time=0.461 ms
64 bytes from 192.168.1.12 icmp_seq=3 ttl=64 time=0.380 ms
64 bytes from 192.168.1.12 icmp_seq=4 ttl=64 time=0.449 ms
64 bytes from 192.168.1.12 icmp_seq=5 ttl=64 time=0.523 ms
VPCS>

PC2-ivmurashov - PuTTY
Please '?' to get help.
Rescuing 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...
VPCS> 192.168.1.10 255.255.255.0 gateway 192.168.1.1
VPCS> save
Saving startup configuration to startup.vpc
. done
VPCS> ping 192.168.1.11
64 bytes from 192.168.1.11 icmp_seq=1 ttl=64 time=0.430 ms
64 bytes from 192.168.1.11 icmp_seq=2 ttl=64 time=0.515 ms
64 bytes from 192.168.1.11 icmp_seq=3 ttl=64 time=0.393 ms
64 bytes from 192.168.1.11 icmp_seq=4 ttl=64 time=0.406 ms
64 bytes from 192.168.1.11 icmp_seq=5 ttl=64 time=0.524 ms
VPCS>

```

Рисунок 3.5: Эхо-запросы

## 3.2 Анализ трафика в GNS3 посредством Wireshark

Запускаю на соединении между PC-1 и коммутатором анализатор трафика (рис. 3.6).



Рисунок 3.6: Запуск анализатора трафика на соединении

В проекте GNS3 стартую все узлы, в окне Wireshark отображается информация по протоколу ARP. В поле физического уровня отображается длина кадра (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) (рис. 3.7).



3	0.036950	Private_66:68:00	Broadcast	ARP	64 Gratuitous ARP for 192.168.1.12 (Request)
4	0.040090	Private_66:68:01	Broadcast	ARP	64 Gratuitous ARP for 192.168.1.11 (Request)
5	1.037087	Private_66:68:00	Broadcast	ARP	64 Gratuitous ARP for 192.168.1.12 (Request)
6	1.041275	Private_66:68:01	Broadcast	ARP	64 Gratuitous ARP for 192.168.1.11 (Request)
7	2.037587	Private_66:68:00	Broadcast	ARP	64 Gratuitous ARP for 192.168.1.12 (Request)
8	2.042285	Private_66:68:01	Broadcast	ARP	64 Gratuitous ARP for 192.168.1.11 (Request)

> Frame 3: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface -, id 0

▼ Ethernet II, Src: Private\_66:68:00 (00:50:79:66:68:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

Destination: Broadcast (ff:ff:ff:ff:ff:ff)

Address: Broadcast (ff:ff:ff:ff:ff:ff)

....1. .... = LG bit: Locally administered address (this is NOT the factory default)

....1. .... = IG bit: Group address (multicast/broadcast)

▼ Source: Private\_66:68:00 (00:50:79:66:68:00)

Address: Private\_66:68:00 (00:50:79:66:68:00)

....0. .... = LG bit: Globally unique address (factory default)

....0. .... = IG bit: Individual address (unicast)

Type: ARP (0x0806)

Padding: 00000000000000000000000000000000

Frame check sequence: 0x00000000 [unverified]

[FCS Status: Unverified]

▼ Address Resolution Protocol (request/gratuitous ARP)

Hardware type: Ethernet (1)

Protocol type: IPv4 (0x0800)

Hardware size: 6

Protocol size: 4

Opcode: request (1)

[Is gratuitous: True]

Sender MAC address: Private\_66:68:00 (00:50:79:66:68:00)

Sender IP address: 192.168.1.12

Target MAC address: Broadcast (ff:ff:ff:ff:ff:ff)

Target IP address: 192.168.1.12

Рисунок 3.7: Анализ трафика в Wireshark

В терминале PC-2 просматриваю информацию по опциям команды ping, введя ping ? (рис. 3.8).

```
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|P|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.
```

Рисунок 3.8: Информация по опциям команды ping

Затем делаю эхо-запрос в ICMP-моду к узлу PC-1 (рис. 3.9).

```
PC2-ivmurashov - PuTTY

VPCS> ping -l 192.168.1.11
Cannot resolve -l

VPCS> ping 192.168.1.11 -l

84 bytes from 192.168.1.11 icmp_seq=1 ttl=64 time=0.950 ms
84 bytes from 192.168.1.11 icmp_seq=2 ttl=64 time=0.748 ms
84 bytes from 192.168.1.11 icmp_seq=3 ttl=64 time=2.182 ms
84 bytes from 192.168.1.11 icmp_seq=4 ttl=64 time=0.582 ms
84 bytes from 192.168.1.11 icmp_seq=5 ttl=64 time=0.331 ms
```

Рисунок 3.9: Эхо-запрос в ICMP-моду к узлу PC-1

В окне Wireshark видим, что в поле сетевого уровня отображается протокол ICMP и IP-адреса отправителя и получателя (рис. 3.10).

11	148.964985	192.168.1.12	192.168.1.11	ICMP	98 Echo (ping) request	id=0x3e30, seq=1/256, ttl=64 (reply in 12)
12	148.965635	192.168.1.11	192.168.1.12	ICMP	98 Echo (ping) reply	id=0x3e30, seq=1/256, ttl=64 (request in 11)
13	149.967536	192.168.1.12	192.168.1.11	ICMP	98 Echo (ping) request	id=0x3f30, seq=2/512, ttl=64 (reply in 14)
14	149.967815	192.168.1.11	192.168.1.12	ICMP	98 Echo (ping) reply	id=0x3f30, seq=2/512, ttl=64 (request in 13)
15	150.978684	192.168.1.12	192.168.1.11	ICMP	98 Echo (ping) request	id=0x4030, seq=3/768, ttl=64 (reply in 16)
16	150.980482	192.168.1.11	192.168.1.12	ICMP	98 Echo (ping) reply	id=0x4030, seq=3/768, ttl=64 (request in 15)
17	151.982808	192.168.1.12	192.168.1.11	ICMP	98 Echo (ping) request	id=0x4130, seq=4/1024, ttl=64 (reply in 18)
18	151.983071	192.168.1.11	192.168.1.12	ICMP	98 Echo (ping) reply	id=0x4130, seq=4/1024, ttl=64 (request in 17)
19	152.985723	192.168.1.12	192.168.1.11	ICMP	98 Echo (ping) request	id=0x4230, seq=5/1280, ttl=64 (reply in 20)
20	152.985925	192.168.1.11	192.168.1.12	ICMP	98 Echo (ping) reply	id=0x4230, seq=5/1280, ttl=64 (request in 19)

Frame 11: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface -, id 0

Ethernet II, Src: Private\_66:68:00 (00:50:79:66:68:00), Dst: Private\_66:68:01 (00:50:79:66:68:01)

Destination: Private\_66:68:01 (00:50:79:66:68:01)

Address: Private\_66:68:01 (00:50:79:66:68:01)

.....0..... = LG bit: Globally unique address (factory default)

.....0..... = IG bit: Individual address (unicast)

Sources: Private\_66:68:00 (00:50:79:66:68:00)

Address: Private\_66:68:00 (00:50:79:66:68:00)

.....0..... = LG bit: Globally unique address (factory default)

.....0..... = IG bit: Individual address (unicast)

Type: IPv4 (0x0000)

Internet Protocol Version 4, Src: 192.168.1.12, Dst: 192.168.1.11

0100 .... = Version: 4

....0101 = Header Length: 20 bytes (5)

> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 84

Identification: 0x303e (12350)

> 000. .... = Flags: 0x0

...0 0000 0000 0000 = Fragment Offset: 0

Time to Live: 64

Protocol: ICMP (1)

Header Checksum: 0xc703 [validation disabled]

[Header checksum status: Unverified]

Source Address: 192.168.1.12

Destination Address: 192.168.1.11

> Internet Control Message Protocol

Рисунок 3.10: Анализ трафика в Wireshark

Затем делаю эхо-запрос в UDP-моду к узлу PC-1 (рис. 3.11).

10

```

PC2-ivmurashov - PuTTY
84 bytes from 192.168.1.11 icmp_seq=3 ttl=64 time=2.182 ms
84 bytes from 192.168.1.11 icmp_seq=4 ttl=64 time=0.582 ms
84 bytes from 192.168.1.11 icmp_seq=5 ttl=64 time=0.331 ms

VPCS> ping 192.168.1.11 -2

84 bytes from 192.168.1.11 udp_seq=1 ttl=64 time=1.458 ms
84 bytes from 192.168.1.11 udp_seq=2 ttl=64 time=0.653 ms
84 bytes from 192.168.1.11 udp_seq=3 ttl=64 time=0.695 ms
84 bytes from 192.168.1.11 udp_seq=4 ttl=64 time=0.772 ms
84 bytes from 192.168.1.11 udp_seq=5 ttl=64 time=0.552 ms

```

Рисунок 3.11: Эхо-запрос в UDP-моду к узлу PC-1

В окне Wireshark видим, что в поле сетевого уровня отображается протокол UDP и IP-адреса отправителя и получателя. В поле канального уровня по нулевому и первому битам можем определить тип мас-адресов: получатель и источник - глобально администрируемые и одиночные, так как биты равны 0 (рис. 3.12).

21	247.257558	192.168.1.12	192.168.1.11	ECHO	98 Request
22	247.258568	192.168.1.11	192.168.1.12	ECHO	98 Response
23	248.260121	192.168.1.12	192.168.1.11	ECHO	98 Request
24	248.260550	192.168.1.11	192.168.1.12	ECHO	98 Response
25	249.263788	192.168.1.12	192.168.1.11	ECHO	98 Request
26	249.264151	192.168.1.11	192.168.1.12	ECHO	98 Response
27	250.267635	192.168.1.12	192.168.1.11	ECHO	98 Request
28	250.268124	192.168.1.11	192.168.1.12	ECHO	98 Response
29	251.270077	192.168.1.12	192.168.1.11	ECHO	98 Request
30	251.270343	192.168.1.11	192.168.1.12	ECHO	98 Response

```

> Frame 21: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface -, id 0
< Ethernet II, Src: Private_66:68:00 (00:50:79:66:68:00), Dst: Private_66:68:01 (00:50:79:66:68:01)
  < Destination: Private_66:68:01 (00:50:79:66:68:01)
    Address: Private_66:68:01 (00:50:79:66:68:01)
    ... ..0. .... = LG bit: Globally unique address (factory default)
    ... ..0. .... = IG bit: Individual address (unicast)
  < Source: Private_66:68:00 (00:50:79:66:68:00)
    Address: Private_66:68:00 (00:50:79:66:68:00)
    ... ..0. .... = LG bit: Globally unique address (factory default)
    ... ..0. .... = IG bit: Individual address (unicast)
  Type: IPv4 (0x0800)
< Internet Protocol Version 4, Src: 192.168.1.12, Dst: 192.168.1.11
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 84
    Identification: 0x30a1 (12449)
  > 000. .... = Flags: 0x0
    ... 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: UDP (17)
    Header Checksum: 0xc690 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.1.12
    Destination Address: 192.168.1.11
  > User Datagram Protocol, Src Port: 3267, Dst Port: 7
  > Echo

```

Рисунок 3.12: Анализ трафика в Wireshark

Затем делаю эхо-запрос в TCP-моду к узлу PC-1 (рис. 3.13).

```
PC2-ivmurashov - PuTTY

VPCS> ping 192.168.1.11 -3

Connect 7@192.168.1.11 seq=1 ttl=64 time=1.633 ms
SendData 7@192.168.1.11 seq=1 ttl=64 time=2.796 ms
Close 7@192.168.1.11 seq=1 ttl=64 time=3.868 ms
Connect 7@192.168.1.11 seq=2 ttl=64 time=2.483 ms
SendData 7@192.168.1.11 seq=2 ttl=64 time=1.849 ms
Close 7@192.168.1.11 seq=2 ttl=64 time=6.822 ms
Connect 7@192.168.1.11 seq=3 ttl=64 time=1.722 ms
SendData 7@192.168.1.11 seq=3 ttl=64 time=6.263 ms
Close 7@192.168.1.11 seq=3 ttl=64 time=3.410 ms
Connect 7@192.168.1.11 seq=4 ttl=64 time=2.389 ms
SendData 7@192.168.1.11 seq=4 ttl=64 time=1.323 ms
Close 7@192.168.1.11 seq=4 ttl=64 time=5.711 ms
Connect 7@192.168.1.11 seq=5 ttl=64 time=1.952 ms
SendData 7@192.168.1.11 seq=5 ttl=64 time=2.030 ms
Close 7@192.168.1.11 seq=5 ttl=64 time=5.474 ms

VPCS>
```

Рисунок 3.13: Эхо-запрос в TCP-моду к узлу PC-1

В окне Wireshark видим, что в поле сетевого уровня отображается протокол TCP и IP-адреса отправителя и получателя. В поле канального уровня по нулевому и первому битам можем определить тип мас-адресов: получатель и источник - глобально администрируемые и одиночные, так как биты равны 0 (рис. 3.14).

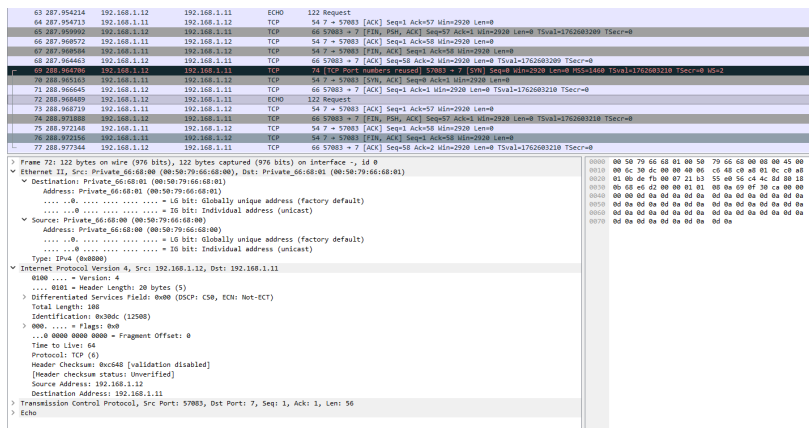


Рисунок 3.14: Анализ трафика в Wireshark

### 3.3 Моделирование простейшей сети на базе маршрутизатора FRR в GNS3

В рабочей области GNS3 размещаю VPCS, коммутатор Ethernet и маршрутизатор FRR и изменяю отображаемые названия устройств (рис. 3.15).

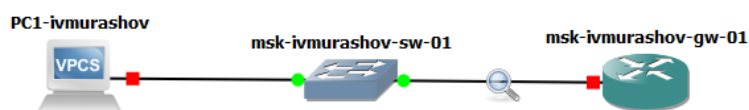


Рисунок 3.15: Топология простейшей сети с маршрутизатором в GNS3

В консоли PC-1 настраиваю IP-адресацию (рис. 3.16).

```
PC1-ivmurashov - PuTTY
Hostname is too long. (Maximum 12 characters)

VPCS> ip 192.168.1.10/24 192.168.1.1
Checking for duplicate address...
VPCS : 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      : 20004
RHOST:PORT  : 127.0.0.1:20005
MTU         : 1500

VPCS>
```

Рисунок 3.16: Настройка IP-адресации

Но при попытке запуска получаю сообщение об ошибке и недоступности KVM виртуализации (рис. 3.17, рис. 3.18)

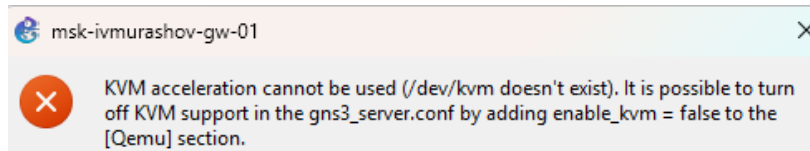


Рисунок 3.17: Ошибка1 | Невозможно использование KVM



Рисунок 3.18: Ошибка2 | Невозможно использование KVM

Как бы я не пытался исправить эту ошибку, ничего не вышло (даже переустанавливал GNS3). В связи с чем, к сожалению, не имею возможности довести до конца выполнение данной лабораторной работы.