

# Лабораторная работа 5

---

Генералов Даниил, НПИбд-01-21, 1032202280

2023

<sup>1</sup>RUDN University, Moscow, Russian Federation

## Задача

---

1. Построить в GNS3 топологию сети, состоящей из коммутатора Ethernet и двух конечных устройств (персональных компьютеров).

## Задача

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.

## Задача

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

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

## Задача

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.
- 
1. С помощью Wireshark захватить и проанализировать ARP-сообщения.
  2. С помощью Wireshark захватить и проанализировать ICMP-сообщения.
- 
1. Построить в GNS3 топологию сети, состоящей из маршрутизатора FRR, коммутатора Ethernet и оконечного устройства.

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



## Задача

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

## Задача

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

## Задача

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.

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

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.

1. Построить в GNS3 топологию сети, состоящей из маршрутизатора

## Задача

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.

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

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.

1. Построить в GNS3 топологию сети, состоящей из маршрутизатора

## Задача

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.

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

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.

1. Построить в GNS3 топологию сети, состоящей из маршрутизатора

## Задача

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.

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

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.

1. Построить в GNS3 топологию сети, состоящей из маршрутизатора

## Выполнение

---

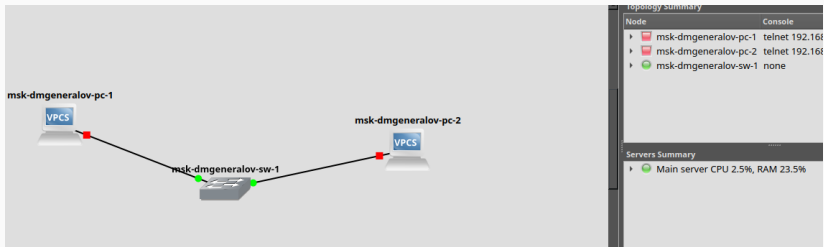


Рис. 1: gns



ATM switch  
Cloud  
Ethernet hub  
Ethernet switch  
Frame Relay switch  
FRR 8.2.2  
NAT

msk-dmgeneralov-pc-1

msk-dmgeneralov-pc-2

msk-dmgeneralov-sw-1

msk-dmgeneralov-pc-1

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

VPCS> ip 192.168.1.2/24  
Checking for duplicate address...  
VPCS : 192.168.1.2 255.255.255.0

VPCS> show

NAME	IP/MASK	GATEWAY	MAC	LPORT	RHOST:PORT
VPCS1	192.168.1.2/24	0.0.0.0	00:50:79:66:08:00	20004	127.0.0.1:20004
5		fe80::250:79ff:fe66:0800/64			

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

VPCS> ping 192.168.1.3

```
84 bytes from 192.168.1.3 icmp_seq=1 ttl=64 time=0.533 ms
84 bytes from 192.168.1.3 icmp_seq=2 ttl=64 time=0.546 ms
84 bytes from 192.168.1.3 icmp_seq=3 ttl=64 time=0.750 ms
```

msk-dmgeneralov-pc-2

Hostname is too long. (Maximum 12 characters)

VPCS> ip 192.168.1.3/24  
Checking for duplicate address...  
VPCS : 192.168.1.3 255.255.255.0

VPCS> show

NAME	IP/MASK	GATEWAY	MAC	LPORT	RHOST:PORT
VPCS1	192.168.1.3/24	0.0.0.0	00:50:79:66:68:01	20006	127.0.0.1:20006
0007		fe80::250:79ff:fe66:6801/64			

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

VPCS> ping 192.168.1.2

```
84 bytes from 192.168.1.2 icmp_seq=1 ttl=64 time=0.467 ms
84 bytes from 192.168.1.2 icmp_seq=2 ttl=64 time=0.305 ms
84 bytes from 192.168.1.2 icmp_seq=3 ttl=64 time=0.541 ms
```

Servers Summary

- msk-dmgeneralov-pc-2 telnet 192.168.122.76:5003
- msk-dmgeneralov-sw-1 none
- Main server CPU 5.8%, RAM 23.5%

Рис. 2: gns

# Ping

The image shows a Wireshark network traffic capture. The top pane displays a network diagram with three virtual machines (VPCS) connected in a triangle topology. The middle pane shows the command prompt of a VPCS instance running a series of ping commands to 192.168.1.2. The bottom pane shows the captured network traffic, including ICMP Echo (ping) requests and responses, as well as TCP connections for the ping utility.

**Command Prompt Output:**

```
VPCS> ping 192.168.1.2 -c 1 -1
84 bytes from 192.168.1.2 icmp_seq=1 ttl=64 time=0.572 ms

VPCS> ping 192.168.1.2 -c 1 -2
84 bytes from 192.168.1.2 udp_seq=1 ttl=64 time=0.526 ms

VPCS> ping 192.168.1.2 -c 1 -3
Connect 70192.168.1.2 seq=1 ttl=64 time=1.453 ms
SendData 70192.168.1.2 seq=1 ttl=64 time=1.132 ms
Close 70192.168.1.2 seq=1 ttl=64 time=2.242 ms

VPCS>
```

**Packet List Table:**

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.1	255.255.255.255	IGMPv3	62	Router Solicitation
2	0.021057	192.168.1.1	255.255.255.255	IGMPv3	62	Router Solicitation
3	0.051051	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.2
4	0.074266	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.2
5	1.051458	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.2
6	1.077390	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.2
7	2.052202	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.2
8	2.079107	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.2
9	102.015150	Private_66:68:00	Broadcast	ARP	64	Who has 192.168.1.2? Tell 192.168.1.2
10	102.015176	Private_66:68:00	Private_66:68:00	ARP	64	192.168.1.2 is at 00:50:7b:66:66:66
11	102.014430	192.168.1.3	192.168.1.2	ICMP	98	Echo (ping) request id=0x21e9
12	102.014705	192.168.1.2	192.168.1.3	ICMP	98	Echo (ping) reply id=0x21e9
13	106.094874	192.168.1.3	192.168.1.2	ICMP	98	Request
14	106.095118	192.168.1.2	192.168.1.3	ICMP	98	Response
15	109.478756	192.168.1.3	192.168.1.2	TCP	74	20350 -> 7 [EST] Seq=6150222
16	109.478903	192.168.1.3	192.168.1.2	TCP	54	7 -> 20350 [RST, ACK] Seq=6150222
17	109.480224	192.168.1.3	192.168.1.2	TCP	64	20350 -> 7 [ACK] Seq=6150222
18	109.480962	192.168.1.3	192.168.1.2	ICMP	122	Request
19	109.481103	192.168.1.2	192.168.1.3	ICMP	54	7 -> 20350 [ACK] Seq=6150222
20	109.482432	192.168.1.3	192.168.1.2	TCP	64	20350 -> 7 [FIN, RST, ACK] Seq=6150222
21	109.482555	192.168.1.3	192.168.1.2	TCP	54	7 -> 20350 [ACK] Seq=6150222
22	109.482611	192.168.1.3	192.168.1.2	TCP	54	7 -> 20350 [FIN, ACK] Seq=6150222
23	109.483137	192.168.1.3	192.168.1.2	TCP	64	20350 -> 7 [ACK] Seq=6150222

**Packet Details:**

Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface 0  
Ethernet II, Src: Private\_66:68:00 (00:50:7b:66:68:00), Dst: IP66:68:00:00:00:00  
Internet Protocol Version 6, Src: 192.168.1.2, Dst: 255.255.255.255  
Internet Control Message Protocol v6

Рис. 3: ping

lab2  
lab3  
lab4

187 Сначала каждый из компьютеров отправляет IPv6 Router Solicit  
188 где он ищет роутер для этой сети.  
189 После этого они делают ARP-запросы, направленные от себя

untitled - GNS3

frf# configure terminal  
frf(config)# hostname msk-dmgeneralov-gw-1  
msk-dmgeneralov-gw-1(config)# exit  
msk-dmgeneralov-gw-1# write mem  
Note: this version of vtysh never writes vtysh.conf  
Building Configuration...  
Integrated configuration saved to /etc/frf/frf.conf  
[OK]  
msk-dmgeneralov-gw-1# conf t  
msk-dmgeneralov-gw-1(config)# int eth0  
msk-dmgeneralov-gw-1(config-if)# ip addr 192.168.1.1/24  
msk-dmgeneralov-gw-1(config-if)# no shutdown  
msk-dmgeneralov-gw-1(config-if)# exit  
msk-dmgeneralov-gw-1(config)# exit  
msk-dmgeneralov-gw-1# write mem  
Note: this version of vtysh never writes vtysh.conf  
Building Configuration...  
Integrated configuration saved to /etc/frf/frf.conf  
[OK]  
msk-dmgeneralov-gw-1# show running-config  
Building configuration...

frf version 8.2.2  
frf defaults traditional  
hostname frf  
hostname msk-dmgeneralov-gw-1  
service integrated-vtysh-config

interface eth0  
ip address 192.168.1.1/24  
exit  
end

msk-dmgeneralov-gw-1# show interface brief

Interface	Status	VRF	Addresses
eth0	up	default	192.168.1.1/24
eth1	down	default	

Trying 192.168.122.76...  
Connected to 192.168.122.76.  
Escape character is '^['.

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	: 00:50:79:66:68:01

Рис. 4: frf

The screenshot displays a network simulation environment with three main components:

- Top Panel (CLI):** Shows a terminal window with the following commands and output:
 

```
lab1> report.md
lab2>
lab3>
lab4>
lab5> report> report.md
lab5> После этого мы выполним три разных типа FRR
lab5> Прежде чем это сделать, второй компьютер с
lab5> Первый компьютер на это отвечает со своим
lab5> Теперь второй компьютер отправляет ICMP Echo
```
- Middle Panel (Network Diagram):** Shows a network topology with three nodes:
  - PC1-dmgeneralov** (VPCS)
  - msk-dmgeneralov-sw-1** (VPCS)
  - msk-dmgeneralov-sw-2** (VPCS)
- Bottom Left Panel (CLI):** Shows the configuration for PC1-dmgeneralov:
 

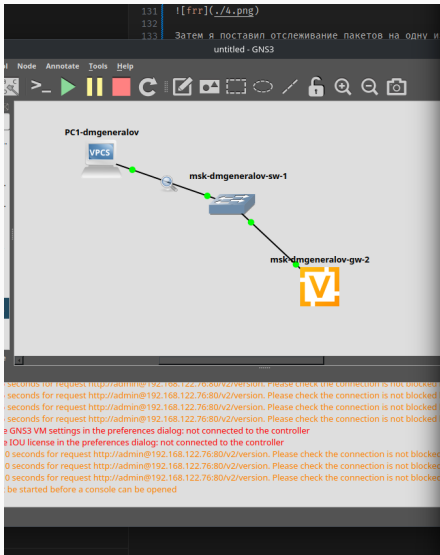
```
GATEWAY : 192.168.1.1
DNS :
MAC : 08:58:79:66:68:01
I.P. PORT : 20806
RHOST:PORT : 127.0.0.1:20807
MTU : 1500

VPCS>
VPCS>
VPCS> ping 192.168.1.1

84 bytes from 192.168.1.1 icmp_seq=1 ttl=64 time=4.339 ms
84 bytes from 192.168.1.1 icmp_seq=2 ttl=64 time=2.669 ms
84 bytes from 192.168.1.1 icmp_seq=3 ttl=64 time=2.571 ms
84 bytes from 192.168.1.1 icmp_seq=4 ttl=64 time=1.841 ms
84 bytes from 192.168.1.1 icmp_seq=5 ttl=64 time=1.152 ms
```
- Bottom Right Panel (Packet Capture):** Shows a packet capture window titled "Capturing from - [msk-dmgeneralov-sw-1] Ethernet0 to PC1-dmgeneralov Ethernet0". The capture shows a series of ICMP Echo (ping) requests and replies. The first frame is highlighted, showing the details of the first ping request:
 

```
Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface Ethernet11, Src: Private_66:68:01 (08:58:79:66:68:01), Dst: Broadcast
Address Resolution Protocol (request)
```

Рис. 5: frr



vyos is a free software distribution that includes multiple components. you can check individual component licenses under /usr/share/doc/vyos/

```
vyos@vyos:~$ install image
You are trying to install from an already installed system. An IP
image file to install or URL must be specified.
```

```
vyos@vyos:~$ configure
[edit]
vyos@vyos# set system host-name msk-dmgeneralov-gw-2
[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-dmgeneralov-gw-2
[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:21:cf:40:00:00
  }
  ethernet eth1 {
    hw-id 0c:21:cf:40:00:01
  }
  ethernet eth2 {
    hw-id 0c:21:cf:40:00:02
  }
  loopback lo {
  }
[edit]
vyos@vyos# exit
exit
vyos@vyos:~$
```

The screenshot displays the VyOS network simulator interface. On the left, a topology diagram shows a VPCS node connected to a switch labeled 'msk-dmgeneralov-sw-1', which is connected to another switch labeled 'msk-dmgeneralov-sw-2'. The VPCS node is also connected to a 'PCI-dmgeneralov' interface. Below the diagram, a terminal window shows the following output:

```

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...
VPCS : 192.168.1.10 255.255.255.0 gateway 192.168.1.1

VPCS> ping 192.168.1.1

84 bytes from 192.168.1.1 icmp_seq=1 ttl=64 time=4.611 ms
64 bytes from 192.168.1.1 icmp_seq=2 ttl=64 time=3.149 ms
84 bytes from 192.168.1.1 icmp_seq=3 ttl=64 time=2.917 ms
84 bytes from 192.168.1.1 icmp_seq=4 ttl=64 time=2.633 ms
84 bytes from 192.168.1.1 icmp_seq=5 ttl=64 time=1.610 ms

VPCS>

```

On the right, a packet capture window is open, showing a list of captured packets. The table below represents the data shown in the packet capture window:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Private_46:68:01	Broadcast	ARP		64 Who has 192.168.1.1? Tell 192.168.1.10
2	0.004000	00:12:cf:40:00:00	Private_46:68:01	ARP		60 192.168.1.1 is at 00:12:cf:40:00:00
3	0.004512	192.168.1.10	192.168.1.1	ICMP		98 Echo (ping) request 10=0x42f
4	0.007678	192.168.1.1	192.168.1.10	ICMP		98 Echo (ping) reply 10=0x42f
5	1.011696	192.168.1.10	192.168.1.1	ICMP		98 Echo (ping) request 10=0x43f
6	1.014253	192.168.1.1	192.168.1.10	ICMP		98 Echo (ping) reply 10=0x43f
7	2.015757	192.168.1.10	192.168.1.1	ICMP		98 Echo (ping) request 10=0x44f
8	2.017968	192.168.1.1	192.168.1.10	ICMP		98 Echo (ping) reply 10=0x44f
9	3.019442	192.168.1.10	192.168.1.1	ICMP		98 Echo (ping) request 10=0x45f
10	3.021639	192.168.1.1	192.168.1.10	ICMP		98 Echo (ping) reply 10=0x45f
11	4.023145	192.168.1.10	192.168.1.1	ICMP		98 Echo (ping) request 10=0x46f
12	4.024462	192.168.1.1	192.168.1.10	ICMP		98 Echo (ping) reply 10=0x46f
13	5.004025	00:12:cf:40:00:00	Private_46:68:01	ARP		60 Who has 192.168.1.1? Tell 192.168.1.10
14	5.005076	Private_46:68:01	00:12:cf:40:00:00	ARP		60 192.168.1.10 is at 00:12:cf:40:00:00

Рис. 7: vyos

Я получил опыт работы с GNS3 для создания сетей, настройки роутеров и компьютеров, и анализа пакетов с помощью Wireshark.