

Лабораторная работа №5

Сетевые технологии

Иванов Сергей Владимирович, НПИбд-01-23

3 ноября 2025

Российский университет дружбы народов, Москва, Россия

Построение простейших моделей сети на базе коммутатора и маршрутизаторов FRR и VyOS в GNS3, анализ трафика посредством Wireshark.

Моделирование простейшей сети на базе коммутатора в GNS3

Запустим GNS3 VM и GNS3. (рис. 1)

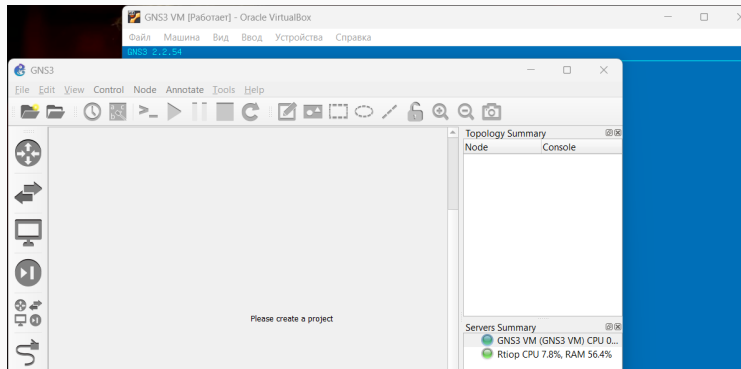


Рис. 1: Запуск GNS3

Моделирование простейшей сети на базе коммутатора в GNS3

Создадим новый проект. (рис. 2)

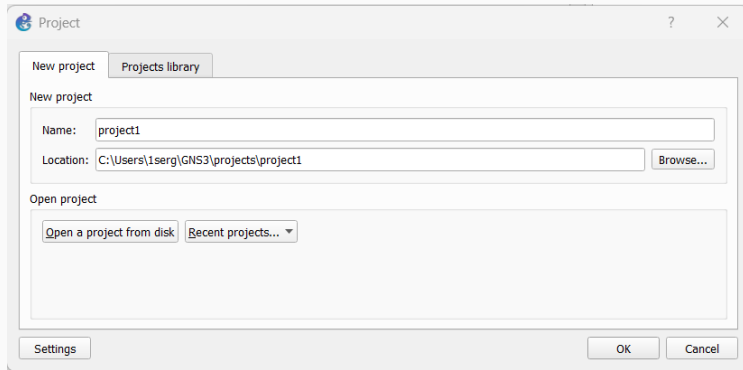


Рис. 2: Создание проекта

Моделирование простейшей сети на базе коммутатора в GNS3

В рабочей области разместим коммутатор Ethernet и два VPCS. Изменим название устройств. (рис. 3)

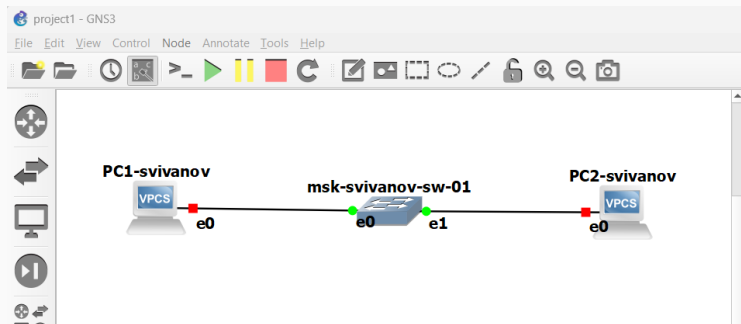
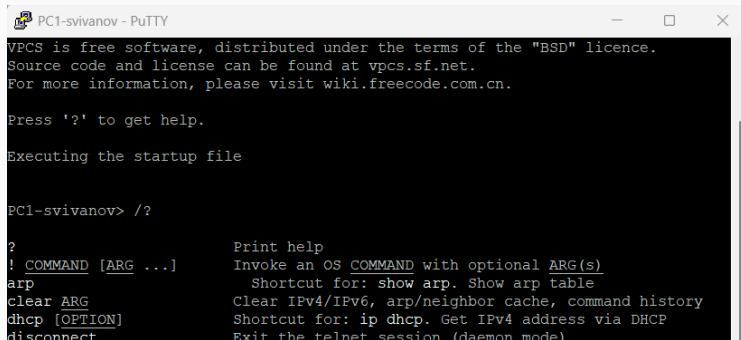


Рис. 3: Размещение компонентов

Моделирование простейшей сети на базе коммутатора в GNS3

Зададим IP-адреса VPCS. (рис. 4)



```
PC1-svivanov - PuTTY
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

PC1-svivanov> /?

?                                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)
```

Рис. 4: Просмотр возможных команд

Моделирование простейшей сети на базе коммутатора в GNS3

Зададим IP-адрес 192.168.1.11 для PC-1.

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

PC1-svivanov> 
```

Рис. 5: Задание ip-адреса

Моделирование простейшей сети на базе коммутатора в GNS3

Аналогичным образом зададим IP-адрес 192.168.1.12 для PC-2. (рис. 6)

```
PC2-svivanov> 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-svivanov> save
Saving startup configuration to startup.vpc
. done

PC2-svivanov> █
```

Рис. 6: Задание ip-адреса

Проверим работоспособность соединения между PC-1 и PC-2 (рис. 7)

```
PC1-svivanov> ping 192.168.1.12
84 bytes from 192.168.1.12 icmp_seq=1 ttl=64 time=0.267 ms
84 bytes from 192.168.1.12 icmp_seq=2 ttl=64 time=0.277 ms
84 bytes from 192.168.1.12 icmp_seq=3 ttl=64 time=0.409 ms
84 bytes from 192.168.1.12 icmp_seq=4 ttl=64 time=0.302 ms
84 bytes from 192.168.1.12 icmp_seq=5 ttl=64 time=0.501 ms
PC1-svivanov> █
```

Рис. 7: Проверка соединения

Анализ трафика в GNS3 посредством Wireshark

Запустим на соединении между PC-1 и коммутатором анализатор трафика.
(рис. 8)

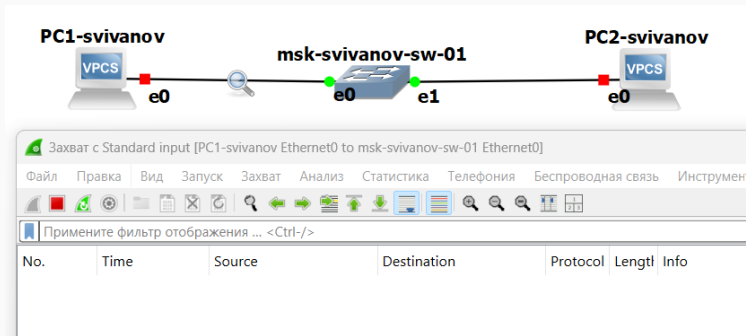
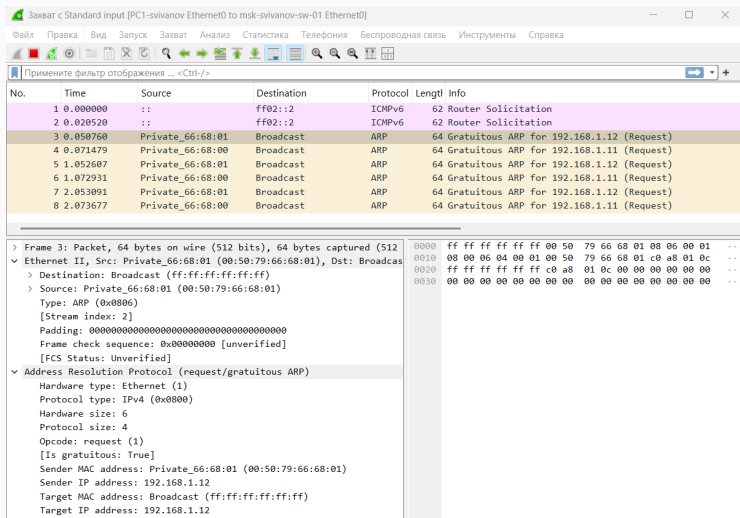


Рис. 8: Запуск анализатора трафика

В проекте GNS3 стартуем все узлы. В окне Wireshark отобразится информация по протоколу ARP. (рис. 9)

Анализ трафика в GNS3 посредством Wireshark



Захват с Standard input [PC1-svivanov Ethernet0 to msk-svivanov-sw-01 Ethernet0]

Файл Правка Вид Запуск Захват Анализ Статистика Телефония Беспроводная связь Инструменты Справка

Примените фильтр отображения ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	::	ff02::2	ICMPv6	62	Router Solicitation
2	0.020520	::	ff02::2	ICMPv6	62	Router Solicitation
3	0.050760	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.12 (Request)
4	0.071479	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.11 (Request)
5	1.052607	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.12 (Request)
6	1.072931	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.11 (Request)
7	2.053091	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.12 (Request)
8	2.073677	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.11 (Request)

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

▼ Ethernet II, Src: Private_66:68:01 (00:50:79:66:68:01), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

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

> Source: Private_66:68:01 (00:50:79:66:68:01)

Type: ARP (0x0806)

[Stream index: 2]

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_66:68:01 (00:50:79:66:68:01)

Sender IP address: 192.168.1.12

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

Target IP address: 192.168.1.12

0000 ff ff ff ff ff ff 00 50 79 66 68 01 08 06 00 01 ...

0010 08 00 06 04 00 01 00 50 79 66 68 01 c0 a8 01 0c ...

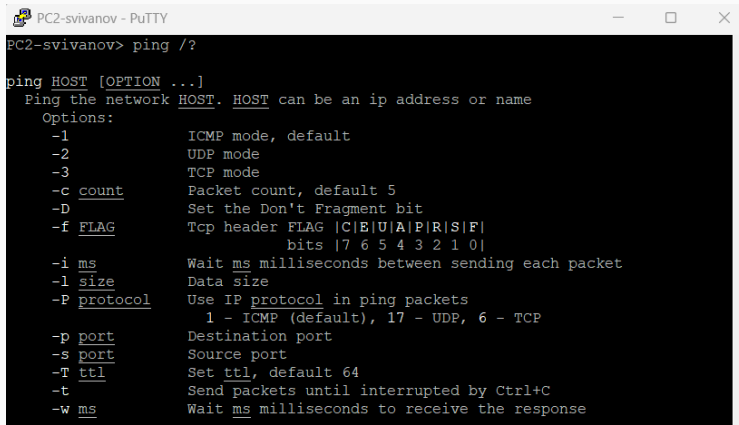
0020 ff ff ff ff ff ff c0 a8 01 0c 00 00 00 00 00 00 ...

0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...

Рис. 9: Протокол ARP

Анализ трафика в GNS3 посредством Wireshark

В терминале PC-2 посмотрим информацию по опциям команды ping (рис. 10)



```
PC2-svivanov> ping /?

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
```

Рис. 10: Просмотр опций команды ping

Сделаем один эхо-запрос в ICMP-моду к узлу PC-1. (рис. 11)

```
PC2-svivanov> ping 192.168.1.11 1 -c 1  
84 bytes from 192.168.1.11 icmp_seq=1 ttl=64 time=0.843 ms  
PC2-svivanov> █
```

Рис. 11: Эхо запрос в ICMP моде

Анализ трафика в GNS3 посредством Wireshark

Захват с Standard input [PC1-svivanov Ethernet0 to msk-svivanov-sw-01 Ethernet0]

Файл Правка Вид Запуск Захват Анализ Статистика Телефония Беспроводная связь Инструменты Справка

Примените фильтр отображения ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
8	2.073677	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.11 (Request)
9	344.494460	Private_66:68:01	Broadcast	ARP	64	Who has 192.168.1.1? Tell 192.168.1.12
10	345.494591	Private_66:68:01	Broadcast	ARP	64	Who has 192.168.1.1? Tell 192.168.1.12
11	346.495625	Private_66:68:01	Broadcast	ARP	64	Who has 192.168.1.1? Tell 192.168.1.12
12	366.835768	Private_66:68:01	Broadcast	ARP	64	Who has 192.168.1.11? Tell 192.168.1.12
13	366.835768	Private_66:68:00	Private_66:68:01	ARP	64	192.168.1.11 is at 00:50:79:66:68:00
→ 14	366.837763	192.168.1.12	192.168.1.11	ICMP	98	Echo (ping) request id=0x8621, seq=1/256, ttl=64 (r
← 15	366.837763	192.168.1.11	192.168.1.12	ICMP	98	Echo (ping) reply id=0x8621, seq=1/256, ttl=64 (r

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: 0x2186 (8582)
- > 000. = Flags: 0x0
- ...0 0000 0000 0000 = Fragment Offset: 0
- Time to Live: 64
- Protocol: ICMP (1)
- Header Checksum: 0xd5bb [validation disabled]
- [Header checksum status: Unverified]
- Source Address: 192.168.1.12
- Destination Address: 192.168.1.11
- [Stream index: 0]

Internet Control Message Protocol

- Type: Echo (ping) request (8)
- Code: 0
- Checksum: 0x99e9 [correct]
- [Checksum Status: Good]
- Identifier (BE): 34337 (0x8621)
- Identifier (LE): 8582 (0x2186)
- Sequence Number (BE): 1 (0x0001)
- Sequence Number (LE): 256 (0x0100)
- [Response frame: 15]

0000 00 50 79 66 68 00 00 50 79 66 68 01 08 00 45 00 -Py
0010 00 54 21 86 00 00 40 01 d5 bb c0 a8 01 0c c0 a8 -TI
0020 01 0b 08 00 99 e9 86 21 00 01 08 09 0a 0b 0c 0d ...
0030 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d ...
0040 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d ...
0050 2e 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d ./0
0060 3e 3f >>

Сделаем один эхо-запрос в UDP-режиме к узлу PC-1. (рис. 13)

```
PC2-svivanov> ping 192.168.1.11 -2 -c 1  
84 bytes from 192.168.1.11 udp_seq=1 ttl=64 time=0.582 ms  
PC2-svivanov> 
```

Рис. 13: Эхо запрос в UDP режиме

Анализ трафика в GNS3 посредством Wireshark

The image shows the Wireshark network protocol analyzer interface. The title bar reads "Захват с Standard input [PC1-svivanov Ethernet0 to msk-svivanov-sw-01 Ethernet0]". The menu bar includes "Файл", "Правка", "Вид", "Запуск", "Захват", "Анализ", "Статистика", "Телефония", "Беспроводная связь", "Инструменты", and "Справка". The toolbar contains various icons for file operations, capture, analysis, and display. The status bar at the top indicates "Примените фильтр отображения ... <Ctrl-/>".

The main display area shows a list of captured packets with the following columns: No., Time, Source, Destination, Protocol, Length, and Info. The packets are as follows:

No.	Time	Source	Destination	Protocol	Length	Info
14	366.837763	192.168.1.12	192.168.1.11	ICMP	98	Echo (ping) request id=0x8621, seq=1/256, ttl=64 (r
15	366.837763	192.168.1.11	192.168.1.12	ICMP	98	Echo (ping) reply id=0x8621, seq=1/256, ttl=64 (r
16	592.148025	Private_66:68:01	Broadcast	ARP	64	Who has 192.168.1.11? Tell 192.168.1.12
17	592.148025	Private_66:68:00	Private_66:68:01	ARP	64	192.168.1.11 is at 00:50:79:66:68:00
18	592.149044	192.168.1.12	192.168.1.11	ICMP	98	Echo (ping) request id=0x6722, seq=1/256, ttl=64 (r
19	592.149044	192.168.1.11	192.168.1.12	ICMP	98	Echo (ping) reply id=0x6722, seq=1/256, ttl=64 (r
20	703.542858	192.168.1.12	192.168.1.11	ECHO	98	Request
21	703.542858	192.168.1.11	192.168.1.12	ECHO	98	Response

The packet details pane on the left shows the selected packet (No. 20) with the following structure:

- 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: 0x22d6 (8918)
 - > 000. = Flags: 0x0
 - ...0 0000 0000 0000 = Fragment Offset: 0
 - Time to Live: 64
 - Protocol: UDP (17)
 - Header Checksum: 0xd45b [validation disabled]
 - [Header checksum status: Unverified]
 - Source Address: 192.168.1.12
 - Destination Address: 192.168.1.11
 - [Stream index: 0]
- User Datagram Protocol, Src Port: 14503, Dst Port: 7
 - Source Port: 14503
 - Destination Port: 7
 - Length: 64
 - Checksum: 0xa7cd [unverified]
 - [Checksum Status: Unverified]
 - [Stream index: 0]
 - [Stream Packet Number: 1]
 - > [Timestamps]
 - UDP payload (56 bytes)

The packet bytes pane on the right shows the raw data of the selected packet in hexadecimal and ASCII:

```
0000 00 50 79 66 68 00 00 50 79 66 68 01 08 00 45 00 -Py
0010 00 54 22 d6 00 00 40 11 d4 5b c0 a8 01 0c c0 a8 -T"
0020 01 0b 38 a7 00 07 00 40 a7 cd 00 50 79 66 68 01 --8
0030 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d --.
0040 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d --.
0050 2e 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d ./0
0060 3e 3f                                     >?
```

Сделаем один эхо-запрос в TCP-моде к узлу PC-1. (рис. 15)

```
PC2-svivanov> ping 192.168.1.11 -3 -c 1
Connect    7@192.168.1.11 seq=1 ttl=64 time=1.027 ms
SendData   7@192.168.1.11 seq=1 ttl=64 time=1.943 ms
Close      7@192.168.1.11 seq=1 ttl=64 time=2.984 ms

PC2-svivanov> █
```

Рис. 15: Эхо запрос в TCP моде

Анализ трафика в GNS3 посредством Wireshark

23 854.233194 Private_66:68:00 Private_66:68:01 ARP 64 192.168.1.11 is at 00:50:79:66:68:00

24 854.234198 192.168.1.12 192.168.1.11 TCP 74 28458 → 7 [SYN] Seq=0 Win=2920 Len=0 MSS=1460 TSval=

25 854.234198 192.168.1.11 192.168.1.12 TCP 54 7 → 28458 [SYN, ACK] Seq=0 Ack=1 Win=2920 Len=0

26 854.235218 192.168.1.12 192.168.1.11 TCP 66 28458 → 7 [ACK] Seq=1 Ack=1 Win=2920 Len=0 TSval=176

27 854.235218 192.168.1.12 192.168.1.11 ECHO 122 Request

28 854.235218 192.168.1.11 192.168.1.12 TCP 54 7 → 28458 [ACK] Seq=1 Ack=57 Win=2920 Len=0

29 854.237200 192.168.1.12 192.168.1.11 TCP 66 28458 → 7 [FIN, PSH, ACK] Seq=57 Ack=1 Win=2920 Len=

30 854.237200 192.168.1.11 192.168.1.12 TCP 54 7 → 28458 [ACK] Seq=1 Ack=58 Win=2920 Len=0

31 854.237200 192.168.1.11 192.168.1.12 TCP 54 7 → 28458 [FIN, ACK] Seq=1 Ack=58 Win=2920 Len=0

32 854.240185 192.168.1.12 192.168.1.11 TCP 66 28458 → 7 [ACK] Seq=58 Ack=2 Win=2920 Len=0 TSval=17

Destination Port: 7
[Stream index: 0]
[Stream Packet Number: 1]
> [Conversation completeness: Complete, WITH_DATA (31)]
[TCP Segment Len: 0]
Sequence Number: 0 (relative sequence number)
Sequence Number (raw): 1871250148
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 0
Acknowledgment number (raw): 0
1010 = Header Length: 40 bytes (10)
> Flags: 0x002 (SYN)
Window: 2920
[Calculated window size: 2920]
Checksum: 0x4e24 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
> Options: (20 bytes), Maximum segment size, No-Operation (NOP), N
> [Timestamps]
[Client Contiguous Streams: 1]
[Server Contiguous Streams: 1]

0000 00 50 79 66 68 00 00 50 79 66 68 01 08 00 45 00 -Py
0010 00 3c 23 6d 00 00 40 06 d3 e7 c0 a8 01 0c c0 a8 -<#
0020 01 0b 6f 2a 00 07 6f 89 02 e4 00 00 00 00 a0 02 --o
0030 0b 68 4e 24 00 00 02 04 05 b4 01 01 08 0a 69 07 -hN
0040 23 6d 00 00 00 00 01 03 03 01 #m-

This shows the raw value of the acknowledgment number (tcp.ack_raw), 4 byte(s) | Пакеты: 32 | Профиль: Default

Рис. 16: Эхо запрос в TCP режиме

Моделирование простейшей сети на базе маршрутизатора FRR в GNS3

Создадим новый проект. В рабочей области разместим VPCS, коммутатор Ethernet и маршрутизатор FRR. (рис. 17)

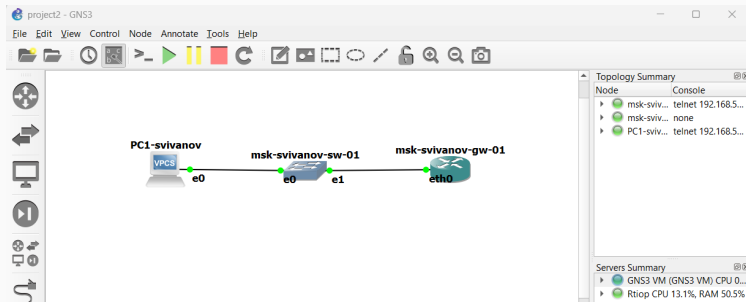


Рис. 17: Построение сети

Моделирование простейшей сети на базе маршрутизатора FRR в GNS3

Включим захват трафика на соединении между коммутатором и маршрутизатором. (рис. 18)

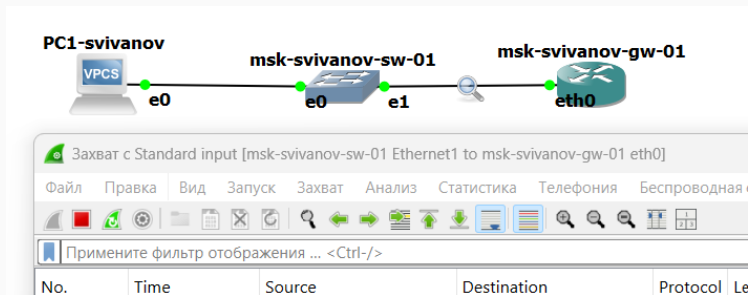


Рис. 18: Включение захвата трафика

Моделирование простейшей сети на базе маршрутизатора FRR в GNS3

Настроим IP-адресацию для интерфейса узла PC1: (рис. 19)

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

PC1-svivanov> show ip

NAME           : PC1-svivanov[1]
IP/MASK        : 192.168.1.10/24
GATEWAY        : 192.168.1.1
DNS            :
MAC            : 00:50:79:66:68:00
```

Рис. 19: Настройка IP адресации PC1

Моделирование простейшей сети на базе маршрутизатора FRR в GNS3

Настроим IP-адресацию для интерфейса локальной сети маршрутизатора:
(рис. 20)

```
frr# configure terminal
frr(config)# hostname msk-user-gw-01
msk-user-gw-01(config)# exit
msk-user-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-user-gw-01# configure terminal
msk-user-gw-01(config)# interface eth0
msk-user-gw-01(config-if)# ip address 192.168.1.1/24
msk-user-gw-01(config-if)# no shutdown
msk-user-gw-01(config-if)# exit
msk-user-gw-01(config)# exit
msk-user-gw-01# memory
% Unknown command: memory
msk-user-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-user-gw-01# █
```

Моделирование простейшей сети на базе маршрутизатора FRR в GNS3

Проверим конфигурацию маршрутизатора и настройки IP-адресации: (рис. 21)

```
msk-user-gw-01# show running-config
Building configuration...

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

Interface	Status	VRF	Addresses
-----	-----	---	-----
eth0	up	default	192.168.1.1/24
eth1	down	default	
eth2	down	default	

Проверим подключение. (рис. 22)

```
PC1-svivanov> ping 192.168.1.1
84 bytes from 192.168.1.1 icmp_seq=1 ttl=64 time=7.227 ms
84 bytes from 192.168.1.1 icmp_seq=2 ttl=64 time=2.750 ms
84 bytes from 192.168.1.1 icmp_seq=3 ttl=64 time=2.128 ms
84 bytes from 192.168.1.1 icmp_seq=4 ttl=64 time=1.714 ms
84 bytes from 192.168.1.1 icmp_seq=5 ttl=64 time=1.162 ms
PC1-svivanov> █
```

Рис. 22: Проверка подключения

Моделирование простейшей сети на базе маршрутизатора FRR в GNS3

Захват с Standard input [msk-svivanov-sw-01 Ethernet1 to msk-svivanov-gw-01 eth0]

Файл Правка Вид Запуск Захват Анализ Статистика Телефония Беспроводная связь Инструменты Справка

Примените фильтр отображения ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
→ 1	0.000000	192.168.1.10	192.168.1.1	ICMP	98	Echo (ping) request id=0xd428, seq=1/256, ttl=64 (re
← 2	0.008600	192.168.1.1	192.168.1.10	ICMP	98	Echo (ping) reply id=0xd428, seq=1/256, ttl=64 (rec
3	1.010384	192.168.1.10	192.168.1.1	ICMP	98	Echo (ping) request id=0xd528, seq=2/512, ttl=64 (ref
4	1.012342	192.168.1.1	192.168.1.10	ICMP	98	Echo (ping) reply id=0xd528, seq=2/512, ttl=64 (rec
5	2.013788	192.168.1.10	192.168.1.1	ICMP	98	Echo (ping) request id=0xd628, seq=3/768, ttl=64 (ref
6	2.015789	192.168.1.1	192.168.1.10	ICMP	98	Echo (ping) reply id=0xd628, seq=3/768, ttl=64 (rec
7	3.019162	192.168.1.10	192.168.1.1	ICMP	98	Echo (ping) request id=0xd728, seq=4/1024, ttl=64 (re
8	3.020163	192.168.1.1	192.168.1.10	ICMP	98	Echo (ping) reply id=0xd728, seq=4/1024, ttl=64 (re
9	4.022324	192.168.1.10	192.168.1.1	ICMP	98	Echo (ping) request id=0xd828, seq=5/1280, ttl=64 (re
10	4.024340	192.168.1.1	192.168.1.10	ICMP	98	Echo (ping) reply id=0xd828, seq=5/1280, ttl=64 (re
11	5.039288	0c:c3:95:2a:00:00	Private_66:68:00	ARP	60	Who has 192.168.1.10? Tell 192.168.1.1
12	5.039288	Private_66:68:00	0c:c3:95:2a:00:00	ARP	60	192.168.1.10 is at 00:50:79:66:68:00

> Frame 1: Packet, 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0

> Ethernet II, Src: Private_66:68:00 (00:50:79:66:68:00), Dst: 0c:c3:95:2a:00:00

> Internet Protocol Version 4, Src: 192.168.1.10, Dst: 192.168.1.1

▼ Internet Control Message Protocol

- Type: Echo (ping) request (8)
- Code: 0
- Checksum: 0x4be2 [correct]
- [Checksum Status: Good]
- Identifier (BE): 54312 (0xd428)
- Identifier (LE): 10452 (0x28d4)
- Sequence Number (BE): 1 (0x0001)
- Sequence Number (LE): 256 (0x0100)
- [Response frame: 2]
- > Data (56 bytes)

0000 0c c3 95 2a 00 00 00 50 79 66 68 00 00 00 45 00 ...
0010 00 54 28 d4 00 00 40 01 ce 79 c0 a8 01 0a c0 a8 -T(
0020 01 01 08 00 4b e2 d4 28 00 01 08 09 0a 0b 0c 0d ...
0030 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d ...
0040 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d ...
0050 2e 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d ./0
0060 3e 3f >?

Рис. 23: Анализ ICMP запросов

Моделирование простейшей сети на базе маршрутизатора VyOS в GNS3

Создадим новый проект. В рабочей области GNS3 разместим VPCS, коммутатор Ethernet и маршрутизатор VyOS. (рис. 24)

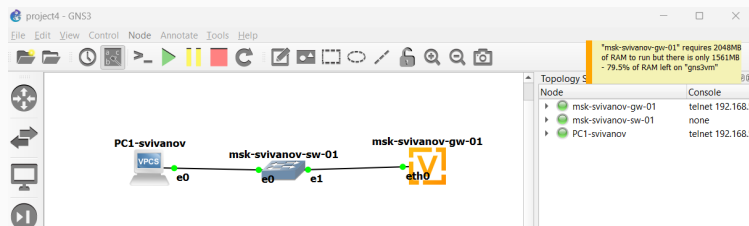


Рис. 24: Построение сети с VyOS

Моделирование простейшей сети на базе маршрутизатора VyOS в GNS3

Включим захват трафика на соединении между коммутатором и маршрутизатором. (рис. 25)

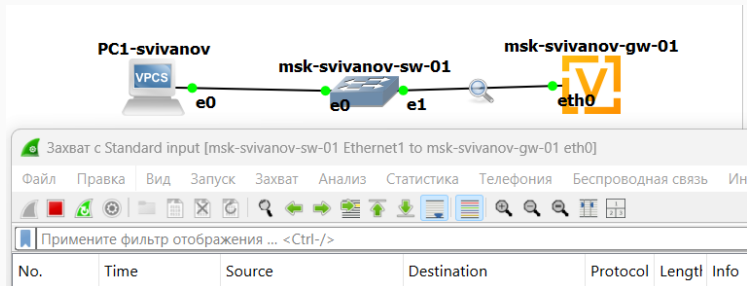


Рис. 25: Включение захвата трафика

Моделирование простейшей сети на базе маршрутизатора VyOS в GNS3

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

PC1-svivanov> show ip

NAME          : PC1-svivanov[1]
IP/MASK       : 192.168.1.10/24
GATEWAY       : 192.168.1.1
```

Рис. 26: Настройка IP адресации PC1

Моделирование простейшей сети на базе маршрутизатора VyOS в GNS3

Настроим маршрутизатор VyOS. (рис. 27)

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

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

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

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

Рис. 27: Загрузка VyOS

Установим систему на диск (рис. 28)

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

Рис. 28: Установка системы VyOS

Изменим имя устройства (рис. 29)

```
vyos@vyos:~$ configure
[edit]
vyos@vyos# system host-name msk-user-gw-01

Invalid command: [system]

[edit]
vyos@vyos# set system host-name msk-user-gw-01
[edit]
vyos@vyos# █
```

Рис. 29: Изменение имени устройства

Моделирование простейшей сети на базе маршрутизатора VyOS в GNS3

Зададим IP-адрес на интерфейсе eth0. Посмотрим внесённые в конфигурацию изменения (рис. 30)

```
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-user-gw-01
[edit]
```

Рис. 30: Задаем IP адрес

Применим изменения в конфигурации и сохраним саму конфигурацию: (рис. 31)

```
vyos@vyos# commit
[edit]
vyos@vyos# save
Saving configuration to '/config/config.boot'...
Done
[edit]
vyos@vyos#
```

Рис. 31: Сохранение изменений

Моделирование простейшей сети на базе маршрутизатора VyOS в GNS3

Посмотрим информацию об интерфейсах маршрутизатора (рис. 32)

```
vyos@vyos# show interfaces
  ethernet eth0 {
    address 192.168.1.1/24
    hw-id 0c:44:58:03:00:00
  }
  ethernet eth1 {
    hw-id 0c:44:58:03:00:01
  }
  ethernet eth2 {
    hw-id 0c:44:58:03:00:02
  }
```

Рис. 32: Просмотр информации об интерфейсах

Проверим подключение. (рис. 33)

```
PC1-svivanov> ping 192.168.1.1
84 bytes from 192.168.1.1 icmp_seq=1 ttl=64 time=1.256 ms
84 bytes from 192.168.1.1 icmp_seq=2 ttl=64 time=0.934 ms
84 bytes from 192.168.1.1 icmp_seq=3 ttl=64 time=1.326 ms
84 bytes from 192.168.1.1 icmp_seq=4 ttl=64 time=1.151 ms
84 bytes from 192.168.1.1 icmp_seq=5 ttl=64 time=1.381 ms
PC1-svivanov> █
```

Рис. 33: Проверка подключения

Моделирование простейшей сети на базе маршрутизатора VyOS в GNS3

14	329.999063	Private_66:68:00	Broadcast	ARP	64	Who has 192.168.1.1? Tell 192.168.1.10
15	330.001558	0c:44:58:03:00:00	Private_66:68:00	ARP	42	192.168.1.1 is at 0c:44:58:03:00:00
→	16	330.004063	192.168.1.10	192.168.1.1	ICMP	98 Echo (ping) request id=0xa95f, seq=1/256, ttl=64 (r
←	17	330.005051	192.168.1.1	192.168.1.10	ICMP	98 Echo (ping) reply id=0xa95f, seq=1/256, ttl=64 (r
	18	331.006329	192.168.1.10	192.168.1.1	ICMP	98 Echo (ping) request id=0xaa5f, seq=2/512, ttl=64 (r
	19	331.006329	192.168.1.1	192.168.1.10	ICMP	98 Echo (ping) reply id=0xaa5f, seq=2/512, ttl=64 (r
	20	332.010155	192.168.1.10	192.168.1.1	ICMP	98 Echo (ping) request id=0xab5f, seq=3/768, ttl=64 (r
	21	332.011177	192.168.1.1	192.168.1.10	ICMP	98 Echo (ping) reply id=0xab5f, seq=3/768, ttl=64 (r
	22	333.012596	192.168.1.10	192.168.1.1	ICMP	98 Echo (ping) request id=0xac5f, seq=4/1024, ttl=64 (r
	23	333.013603	192.168.1.1	192.168.1.10	ICMP	98 Echo (ping) reply id=0xac5f, seq=4/1024, ttl=64 (r
	24	334.016262	192.168.1.10	192.168.1.1	ICMP	98 Echo (ping) request id=0xad5f, seq=5/1280, ttl=64 (r
	25	334.017268	192.168.1.1	192.168.1.10	ICMP	98 Echo (ping) reply id=0xad5f, seq=5/1280, ttl=64 (r
	26	335.180493	0c:44:58:03:00:00	Private_66:68:00	ARP	42 Who has 192.168.1.10? Tell 192.168.1.1
	27	335.180493	Private_66:68:00	0c:44:58:03:00:00	ARP	42 192.168.1.10 is at 00:50:79:66:68:00

> Frame 17: Packet, 98 bytes on wire (784 bits), 98 bytes captured (784	0000	00 50 79 66 68 00 0c 44 58 03 00 00 08 00 45 00	-Py
> Ethernet II, Src: 0c:44:58:03:00:00 (0c:44:58:03:00:00), Dst: Private	0010	00 54 c3 75 00 00 40 01 33 d8 c0 a8 01 01 c0 a8	-T
> Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.10	0020	01 0a 00 00 7e ab a9 5f 00 01 08 09 0a 0b 0c 0d	...
▼ Internet Control Message Protocol	0030	0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d	...
Type: Echo (ping) reply (0)	0040	1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d	...
Code: 0	0050	2e 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d	...
Checksum: 0x7eab [correct]	0060	3e 3f	>>
[Checksum Status: Good]			
Identifier (BE): 43359 (0xa95f)			
Identifier (LE): 24489 (0x5fa9)			
Sequence Number (BE): 1 (0x0001)			
Sequence Number (LE): 256 (0x0100)			
[Request frame: 16]			
[Response time: 0,988 ms]			
> Data (56 bytes)			

Рис. 34: Анализ эхо запросов

Вывод

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