

Отчёт по лабораторной работе 5

Простые сети в GNS3. Анализ трафика

Метвалли Ахмед Фарг Набеев

Содержание

1	Цель работы	5
2	Выполнение	6
2.1	Моделирование простейшей сети на базе коммутатора в GNS3 . . .	6
2.1.1	Построение топологии и настройка IP-адресов	6
2.1.2	Анализ трафика в Wireshark	8
2.2	Моделирование простейшей сети на базе маршрутизатора FRR в GNS3	12
2.2.1	Построение топологии и настройка устройств	12
2.2.2	Проверка связи и анализ трафика	15
2.3	Моделирование простейшей сети на базе маршрутизатора VyOS в GNS3	17
2.3.1	Построение топологии и настройка устройств	17
2.3.2	Проверка связи и анализ трафика	20
3	Заключение	22

Список иллюстраций

2.1	Топология сети в GNS3	6
2.2	Просмотр списка команд VPCS	7
2.3	Настройка IP и проверка связи между ПК	7
2.4	Захват ARP-пакетов в Wireshark	8
2.5	ICMP-эхо-запрос и ответ	9
2.6	UDP-эхо-запрос в Wireshark	10
2.7	TCP-эхо-запрос и установка соединения	11
2.8	Проверка различных режимов ping в VPCS	12
2.9	Топология сети с маршрутизатором FRR	13
2.10	Настройка IP-адреса для PC1	13
2.11	Настройка маршрутизатора FRR	14
2.12	Проверка конфигурации FRR	15
2.13	Проверка связи между PC1 и маршрутизатором	16
2.14	Захват ICMP и ARP пакетов в Wireshark	17
2.15	Топология сети с маршрутизатором VyOS	18
2.16	Настройка маршрутизатора VyOS	19
2.17	Просмотр интерфейсов VyOS	20
2.18	Проверка связи между ПК и маршрутизатором	20
2.19	Анализ ICMP и ARP пакетов в Wireshark	21

Список таблиц

1 Цель работы

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

2 Выполнение

2.1 Моделирование простейшей сети на базе коммутатора в GNS3

2.1.1 Построение топологии и настройка IP-адресов

1. В приложении **GNS3** создан новый проект и размещены устройства: коммутатор **msk-ahmedfarg-sw-01** и два виртуальных ПК — **PC1-ahmedfarg** и **PC2-ahmedfarg**.

Устройства соединены кабелями Ethernet, формируя простую локальную сеть.

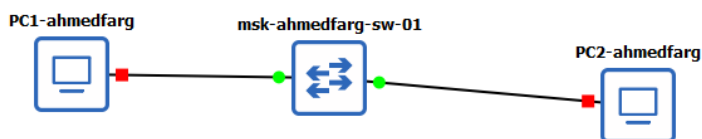


Рис. 2.1: Топология сети в GNS3

2. На каждом ПК была открыта консоль **VPCS**. Для просмотра доступных команд введена команда **?**, отобразившая список поддерживаемых операций.

```
VPCS> ?
?          Print help
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>
```

Рис. 2.2: Просмотр списка команд VPCS

3. Выполнена настройка IP-адресов:

- для **PC1-ahmedfarg**: ip 192.168.1.11/24 192.168.1.1
- для **PC2-ahmedfarg**: ip 192.168.1.12/24 192.168.1.1

После настройки параметры были сохранены командой save. Проверка связи с помощью ping показала успешный обмен ICMP-пакетами между устройствами.

```
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> ping 192.168.1.12
84 bytes from 192.168.1.12 icmp_seq=1 ttl=64 time=2.121 ms
84 bytes from 192.168.1.12 icmp_seq=2 ttl=64 time=1.659 ms
84 bytes from 192.168.1.12 icmp_seq=3 ttl=64 time=2.032 ms
84 bytes from 192.168.1.12 icmp_seq=4 ttl=64 time=2.750 ms
84 bytes from 192.168.1.12 icmp_seq=5 ttl=64 time=2.290 ms

VPCS> ping 192.168.1.11
84 bytes from 192.168.1.11 icmp_seq=1 ttl=64 time=1.261 ms
84 bytes from 192.168.1.11 icmp_seq=2 ttl=64 time=1.672 ms
84 bytes from 192.168.1.11 icmp_seq=3 ttl=64 time=1.589 ms
84 bytes from 192.168.1.11 icmp_seq=4 ttl=64 time=2.006 ms
84 bytes from 192.168.1.11 icmp_seq=5 ttl=64 time=0.909 ms

VPCS>
```

Рис. 2.3: Настройка IP и проверка связи между ПК

2.1.2 Анализ трафика в Wireshark

4. Для анализа сетевого обмена запущен захват трафика на линке между **PC1** и коммутатором. В окне **Wireshark** зафиксированы ARP-пакеты, определяющие соответствие IP- и MAC-адресов.

На скриншоте видно, что оба устройства отправляют *Gratuitous ARP*-запросы с целью обновления ARP-таблиц в сети.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	::	ff02::2	ICMPv6	62	Router Solicitation
2	0.000917	::	ff02::2	ICMPv6	62	Router Solicitation
3	0.050606	Private 66:68:01	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.12 (Request)
4	0.051254	Private 66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.11 (Request)
5	1.051596	Private 66:68:01	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.12 (Request)
6	1.051703	Private 66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.11 (Request)
7	2.052226	Private 66:68:01	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.12 (Request)
8	2.052371	Private 66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.11 (Request)

Frame 4: 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)

Source: Private 66:68:00 (00:50:79:66:68:00)

Type: ARP (0x0806)

[Stream index: 3]

Padding: 00000000000000000000000000000000

Frame check sequence: 0x00000000 [unverified]

[FCS Status: Unverified]

Address Resolution Protocol (request/gratuitous ARP)

Hardware type: Ethernet (1)

Protocol type: IPv4 (0x0000)

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: 192.168.1.11

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

Target IP address: 192.168.1.11

Рис. 2.4: Захват ARP-пакетов в Wireshark

5. Далее был выполнен обмен ICMP-пакетами (эхо-запрос и эхо-ответ) при выполнении команды `ping` с **PC2** на **PC1**.

На захвате видно тип пакета *Echo (ping) request/reply* и соответствующие IP-адреса источника и назначения.

1 0.000000	::	ff02::2	ICMPv6	62 Router Solicitation
2 0.000917	::	ff02::2	ICMPv6	62 Router Solicitation
3 0.050606	Private_66:68:01	Broadcast	ARP	64 Gratuitous ARP for 192.168.1.12 (Request)
4 0.051254	Private_66:68:00	Broadcast	ARP	64 Gratuitous ARP for 192.168.1.11 (Request)
5 1.051596	Private_66:68:01	Broadcast	ARP	64 Gratuitous ARP for 192.168.1.12 (Request)
6 1.051703	Private_66:68:00	Broadcast	ARP	64 Gratuitous ARP for 192.168.1.11 (Request)
7 2.052226	Private_66:68:01	Broadcast	ARP	64 Gratuitous ARP for 192.168.1.12 (Request)
8 2.052371	Private_66:68:00	Broadcast	ARP	64 Gratuitous ARP for 192.168.1.11 (Request)
9 72.429825	Private_66:68:01	Broadcast	ARP	64 Who has 192.168.1.11? Tell 192.168.1.12
10 72.431223	Private_66:68:00	Private_66:68:01	ARP	64 192.168.1.11 is at 00:50:79:66:68:00
11 72.432292	192.168.1.12	192.168.1.11	ICMP	98 Echo (ping) request id=0xcd29, seq=1/256, ttl=64 (reply in 12)
12 72.434067	192.168.1.11	192.168.1.12	ICMP	98 Echo (ping) reply id=0xcd29, seq=1/256, ttl=64 (request in 11)

> Frame 11: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface -, id 0 > Ethernet II, Src: Private_66:68:01 (00:50:79:66:68:01), Dst: Private_66:68:00 (00:50:79:66:68:00) > 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: 0x29cd (10701) 000. = Flags: 0x0 ...0 0000 0000 0000 = Fragment Offset: 0 Time to Live: 64 Protocol: ICMP (1) Header Checksum: 0xcd74 [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: 8 (Echo (ping) request) Code: 0 Checksum: 0x52e1 [correct] [Checksum Status: Good] Identifier (BE): 52521 (0xcd29) Identifier (LE): 10701 (0x29cd) Sequence Number (BE): 1 (0x0001) Sequence Number (LE): 256 (0x0100) [Response frame 12] > Data (56 bytes)	0000 00 50 7 0010 00 54 2 0020 01 0b 0 0030 0e 0f 1 0040 1e 1f 2 0050 2e 2f 3 0060 3e 3f
---	--

Рис. 2.5: ICMP-эхо-запрос и ответ

6. При выполнении ring с флагом -2 был отправлен UDP-эхо-запрос.

Анализ показал использование протокола **UDP (port 7)** для передачи данных, при этом в поле *Payload* отображается содержимое передаваемого сообщения.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	::	ff02::2	ICMPv6	62	Router Solicitation
2	0.000917	::	ff02::2	ICMPv6	62	Router Solicitation
3	0.050606	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.12 (Request)
4	0.051254	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.11 (Request)
5	1.051596	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.12 (Request)
6	1.051703	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.11 (Request)
7	2.052226	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.12 (Request)
8	2.052371	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192.168.1.11 (Request)
9	72.429825	Private_66:68:01	Broadcast	ARP	64	Who has 192.168.1.11? Tell 192.168.1.12
10	72.431223	Private_66:68:00	Private_66:68:01	ARP	64	192.168.1.11 is at 00:50:79:66:68:00
11	72.432292	192.168.1.12	192.168.1.11	ICMP	98	Echo (ping) request id=0xcd29, seq=1/256, ttl=64 (reply in 12)
12	72.434067	192.168.1.11	192.168.1.12	ICMP	98	Echo (ping) reply id=0xcd29, seq=1/256, ttl=64 (request in 11)
13	152.549098	192.168.1.12	192.168.1.11	ECHO	98	Request
14	152.549765	192.168.1.11	192.168.1.12	ECHO	98	Response

<p>Frame 13: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface -, id 0</p> <p>Ethernet II, Src: Private_66:68:01 (00:50:79:66:68:01), Dst: Private_66:68:00 (00:50:79:66:68:00)</p> <p>Internet Protocol Version 4, Src: 192.168.1.12, Dst: 192.168.1.11</p> <p>0100 = Version: 4</p> <p>.... 0101 = Header Length: 20 bytes (5)</p> <p>Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)</p> <p>Total Length: 84</p> <p>Identification: 0x2a1e (10782)</p> <p>0000 = Flags: 0x0</p> <p>...0 0000 0000 0000 = Fragment Offset: 0</p> <p>Time to Live: 64</p> <p>Protocol: UDP (17)</p> <p>Header Checksum: 0xcd13 [validation disabled]</p> <p>[Header checksum status: Unverified]</p> <p>Source Address: 192.168.1.12</p> <p>Destination Address: 192.168.1.11</p> <p>[Stream Index: 0]</p> <p>User Datagram Protocol, Src Port: 37007, Dst Port: 7</p> <p>Source Port: 37007</p> <p>Destination Port: 7</p> <p>Length: 64</p> <p>Checksum: 0x4fe5 [unverified]</p> <p>[Checksum Status: Unverified]</p> <p>[Stream Index: 0]</p> <p>[Stream Packet Number: 1]</p> <p>[Timestamps]</p> <p>UDP payload (56 bytes)</p> <p>Echo</p> <p>Echo data: 0050796668010e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f</p>	<p>0000 00 50 79</p> <p>0010 00 54 2a</p> <p>0020 01 0b 9d</p> <p>0030 0e 0f 1d</p> <p>0040 1e 1f 20</p> <p>0050 2e 2f 30</p> <p>0060 3e 3f</p>
--	---

Рис. 2.6: UDP-эхо-запрос в Wireshark

7. При использовании опции -3 команда ring выполнялась в режиме **TCP**.
- В захваченных пакетах Wireshark зафиксированы типичные фазы установления TCP-соединения: *SYN*, *SYN-ACK* и *ACK*, подтверждающие корректную работу транспортного уровня.

10	72.431223	Private_66:68:00	Private_66:68:01	ARP	64	192.168.1.11 is at 00:50:79:66:68:00
11	72.432292	192.168.1.12	192.168.1.11	ICMP	98	Echo (ping) request id=0xcd29, seq=1/256, ttl=64 (reply in 12)
12	72.434867	192.168.1.11	192.168.1.12	ICMP	98	Echo (ping) reply id=0xcd29, seq=1/256, ttl=64 (request in 11)
13	152.549098	192.168.1.12	192.168.1.11	ECHO	98	Request
14	152.549765	192.168.1.11	192.168.1.12	ECHO	98	Response
15	182.469486	192.168.1.12	192.168.1.11	TCP	74	6940 → 7 [SYN] Seq=0 Win=2920 Len=0 MSS=1460 TSval=1761290812 TSecr=0 WS=2
16	182.470842	192.168.1.11	192.168.1.12	TCP	54	7 → 6940 [SYN, ACK] Seq=0 Ack=1 Win=2920 Len=0
17	182.472851	192.168.1.12	192.168.1.11	TCP	66	6940 → 7 [ACK] Seq=1 Ack=1 Win=2920 Len=0 TSval=1761290812 TSecr=0
18	182.475464	192.168.1.12	192.168.1.11	ECHO	122	Request
19	182.477111	192.168.1.11	192.168.1.12	TCP	54	7 → 6940 [ACK] Seq=1 Ack=57 Win=2920 Len=0
20	182.481696	192.168.1.12	192.168.1.11	TCP	66	6940 → 7 [FIN, PSH, ACK] Seq=57 Ack=1 Win=2920 Len=0 TSval=1761290812 TSecr=0
21	182.482181	192.168.1.11	192.168.1.12	TCP	54	7 → 6940 [ACK] Seq=1 Ack=58 Win=2920 Len=0
22	182.482380	192.168.1.11	192.168.1.12	TCP	54	7 → 6940 [FIN, ACK] Seq=1 Ack=58 Win=2920 Len=0
23	182.484548	192.168.1.12	192.168.1.11	TCP	66	6940 → 7 [ACK] Seq=58 Ack=2 Win=2920 Len=0 TSval=1761290812 TSecr=0


```

Frame 16: 54 bytes on wire (432 bits), 54 bytes captured (432 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)
  Internet Protocol Version 4, Src: 192.168.1.11, Dst: 192.168.1.12
    Transmission Control Protocol, Src Port: 7, Dst Port: 6940, Seq: 0, Ack: 1, Len: 0
      Source Port: 7
      Destination Port: 6940
      [Stream index: 0]
      [Stream Packet Number: 2]
      [Conversation completeness: Complete, WITH_DATA (31)]
      [TCP Segment Len: 0]
      Sequence Number: 0 (relative sequence number)
      Sequence Number (raw): 1242142575
      [Next Sequence Number: 1 (relative sequence number)]
      Acknowledgment Number: 1 (relative ack number)
      Acknowledgment number (raw): 637225478
      0101 .... = Header Length: 20 bytes (5)
      [Flags: 0x012 (SYN, ACK)]
      Window: 2920
      [Calculated window size: 2920]
      Checksum: 0x815c [unverified]
      [Checksum Status: Unverified]
      Urgent Pointer: 0
      [Timestamps]
      [SEQ/ACK analysis]

```

Рис. 2.7: TCP-эхо-запрос и установка соединения

- В терминале VPCS была изучена справка по команде `ping`, где представлены возможные режимы работы: ICMP (-1), UDP (-2), TCP (-3).
Все три режима были протестированы, что подтвердило корректное взаимодействие на уровнях **ICMP**, **UDP** и **TCP**.

```
VPCS> 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

Notes: 1. Using names requires DNS to be set.
       2. Use Ctrl+C to stop the command.

VPCS> ping 192.168.1.11 -c 1
64 bytes from 192.168.1.11 icmp_seq=1 ttl=64 time=3.048 ms

VPCS> ping 192.168.1.11 -c 1 -2
64 bytes from 192.168.1.11 udp_seq=1 ttl=64 time=1.612 ms

VPCS> ping 192.168.1.11 -c 1 -3
Connect  7@192.168.1.11 seq=1 ttl=64 time=3.769 ms
SendData 7@192.168.1.11 seq=1 ttl=64 time=3.933 ms
Close    7@192.168.1.11 seq=1 ttl=64 time=2.934 ms

VPCS>
```

Рис. 2.8: Проверка различных режимов ping в VPCS

2.2 Моделирование простейшей сети на базе маршрутизатора FRR в GNS3

2.2.1 Построение топологии и настройка устройств

1. В приложении **GNS3** создан новый проект. В рабочей области размещены устройства: коммутатор **msk-ahmedfarg-sw-01**, маршрутизатор **msk-ahmedfarg-gw-01** и конечное устройство **PC1-ahmedfarg**.

Все элементы соединены между собой кабелями Ethernet, формируя простейшую сеть.

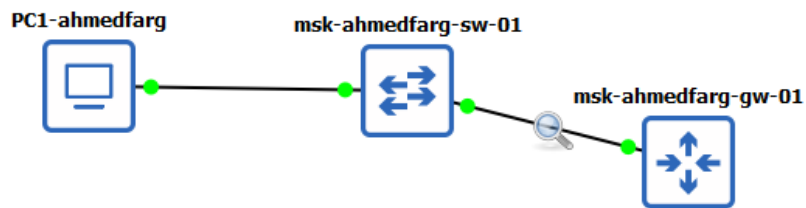


Рис. 2.9: Топология сети с маршрутизатором FRR

- В консоли **VPCS** для узла **PC1-ahmedfarg** выполнена настройка IP-адреса:

```

ip 192.168.1.10/24 192.168.1.1
save
show ip

```

В результате устройство получило адрес 192.168.1.10/24 с шлюзом 192.168.1.1.

```

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

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      : 10004
RHOST:PORT : 127.0.0.1:10005
MTU        : 1500

VPCS>

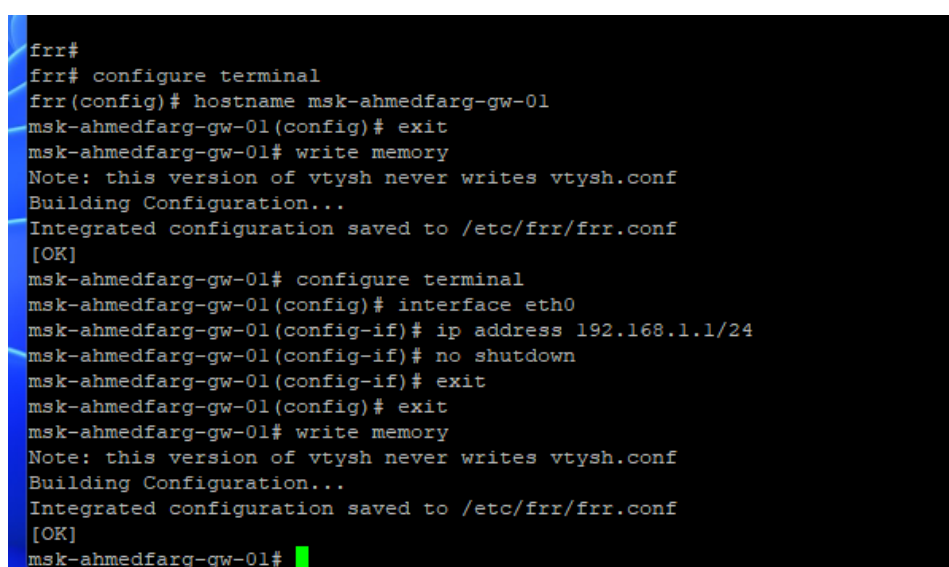
```

Рис. 2.10: Настройка IP-адреса для PC1

- На маршрутизаторе **FRR** выполнена базовая настройка. В режиме конфигурации был изменён hostname и задан IP-адрес интерфейса eth0:

```
configure terminal
hostname msk-ahmedfarg-gw-01
interface eth0
ip address 192.168.1.1/24
no shutdown
exit
write memory
```

Конфигурация была сохранена и успешно применена.



```
frr#
frr# configure terminal
frr(config)# hostname msk-ahmedfarg-gw-01
msk-ahmedfarg-gw-01(config)# exit
msk-ahmedfarg-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-ahmedfarg-gw-01# configure terminal
msk-ahmedfarg-gw-01(config)# interface eth0
msk-ahmedfarg-gw-01(config-if)# ip address 192.168.1.1/24
msk-ahmedfarg-gw-01(config-if)# no shutdown
msk-ahmedfarg-gw-01(config-if)# exit
msk-ahmedfarg-gw-01(config)# exit
msk-ahmedfarg-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-ahmedfarg-gw-01#
```

Рис. 2.11: Настройка маршрутизатора FRR

4. Для проверки настроек маршрутизатора использованы команды:

```
show running-config
show interface brief
```

В выводе показано, что интерфейс **eth0** активен и имеет адрес 192.168.1.1/24.

```

msk-ahmedfarg-gw-01# show running-config
Building configuration...

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

```

Рис. 2.12: Проверка конфигурации FRR

2.2.2 Проверка связи и анализ трафика

- На **PC1-ahmedfarg** выполнена проверка связи с маршрутизатором. Команда `ping 192.168.1.1` показала успешные ответы, что подтверждает корректность настройки IP-сети.

```
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      : 10004
RHOST:PORT  : 127.0.0.1:10005
MTU         : 1500

VPCS>
VPCS> ping 192.168.1.1

84 bytes from 192.168.1.1 icmp_seq=1 ttl=64 time=5.054 ms
84 bytes from 192.168.1.1 icmp_seq=2 ttl=64 time=3.336 ms
84 bytes from 192.168.1.1 icmp_seq=3 ttl=64 time=3.501 ms
84 bytes from 192.168.1.1 icmp_seq=4 ttl=64 time=4.085 ms
84 bytes from 192.168.1.1 icmp_seq=5 ttl=64 time=3.708 ms

VPCS>
```

Рис. 2.13: Проверка связи между PC1 и маршрутизатором

6. В **Wireshark** был запущен захват пакетов на линии между коммутатором и маршрутизатором.

Анализ показал обмен ICMP-пакетами *Echo Request* и *Echo Reply*, а также предварительные ARP-запросы, использовавшиеся для определения MAC-адресов.

Это подтверждает успешное взаимодействие между конечным устройством и маршрутизатором.

No.	Time	Source	Destination	Protocol	Length	Info
15	247.702524	Private_66:68:00	Broadcast	ARP	64	Who has 192.168.1.1? Tell 192.168.1.10
16	247.706995	0c:be:b1:30:00:00	Private_66:68:00	ARP	60	192.168.1.1 is at 0c:be:b1:30:00:00
17	247.708986	192.168.1.10	192.168.1.1	ICMP	98	Echo (ping) request id=0xa42b, seq=1/256, ttl=64 (reply in 18)
18	247.712756	192.168.1.1	192.168.1.10	ICMP	98	Echo (ping) reply id=0xa42b, seq=1/256, ttl=64 (request in 17)
19	248.714566	192.168.1.10	192.168.1.1	ICMP	98	Echo (ping) request id=0xa52b, seq=2/512, ttl=64 (reply in 20)
20	248.717031	192.168.1.1	192.168.1.10	ICMP	98	Echo (ping) reply id=0xa52b, seq=2/512, ttl=64 (request in 19)
21	249.719583	192.168.1.10	192.168.1.1	ICMP	98	Echo (ping) request id=0xa62b, seq=3/768, ttl=64 (reply in 22)
22	249.721816	192.168.1.1	192.168.1.10	ICMP	98	Echo (ping) reply id=0xa62b, seq=3/768, ttl=64 (request in 21)
23	250.724441	192.168.1.10	192.168.1.1	ICMP	98	Echo (ping) request id=0xa72b, seq=4/1024, ttl=64 (reply in 24)
24	250.727281	192.168.1.1	192.168.1.10	ICMP	98	Echo (ping) reply id=0xa72b, seq=4/1024, ttl=64 (request in 23)
25	251.728606	192.168.1.10	192.168.1.1	ICMP	98	Echo (ping) request id=0xa82b, seq=5/1280, ttl=64 (reply in 26)
26	251.731391	192.168.1.1	192.168.1.10	ICMP	98	Echo (ping) reply id=0xa82b, seq=5/1280, ttl=64 (request in 25)
27	252.773800	0c:be:b1:30:00:00	Private_66:68:00	ARP	60	Who has 192.168.1.10? Tell 192.168.1.1
28	252.774428	Private_66:68:00	0c:be:b1:30:00:00	ARP	60	192.168.1.10 is at 00:50:79:66:68:00


```

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: 0c:be:b1:30:00:00 (0c:be:b1:30:00:00)
Internet Protocol Version 4, Src: 192.168.1.10, Dst: 192.168.1.1
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 64
    Identification: 0x2ba6 (11174)
    000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: ICMP (1)
    Header Checksum: 0xcba7 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.1.10
    Destination Address: 192.168.1.1
    [Stream index: 0]
Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0
  Checksum: 0x79dd [correct]
  [Checksum Status: Good]
  Identifier (BE): 42539 (0xa62b)
  Identifier (LE): 11174 (0x2ba6)
  Sequence Number (BE): 3 (0x0003)
  Sequence Number (LE): 768 (0x0300)
  [Response frame: 22]
  Data (56 bytes)

```

Рис. 2.14: Захват ICMP и ARP пакетов в Wireshark

2.3 Моделирование простейшей сети на базе маршрутизатора VyOS в GNS3

2.3.1 Построение топологии и настройка устройств

1. В приложении **GNS3** создан новый проект. В рабочей области размещены устройства: коммутатор **msk-ahmedfarg-sw-01**, маршрутизатор **msk-ahmedfarg-gw-01** на базе **VyOS** и конечное устройство **PC1-ahmedfarg**.

Все элементы соединены между собой кабелями Ethernet, формируя простейшую сеть.

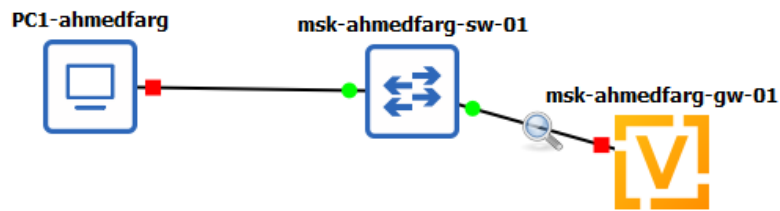


Рис. 2.15: Топология сети с маршрутизатором VyOS

2. На ПК **PC1-ahmedfarg** через консоль **VPCS** была выполнена настройка IP-адреса:

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

В результате узлу был присвоен адрес 192.168.1.10/24, а шлюзом по умолчанию назначен 192.168.1.1.

3. На маршрутизаторе **VyOS** после входа в систему выполнен переход в режим конфигурации командой `configure`.

Затем заданы имя устройства и IP-адрес интерфейса eth0:

```
set system host-name msk-ahmedfarg-gw-01
set interfaces ethernet eth0 address 192.168.1.1/24
compare
commit
save
```

Изменения применены и сохранены.

```
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:~$ 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:~$ configure
[edit]
vyos@vyos# system

    Invalid command: [system]

[edit]
vyos@vyos# set system host-name msk-ahmedfarg-gw-01
[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-ahmedfarg-gw-01
[edit]
th
vyos@vyos#
```

Рис. 2.16: Настройка маршрутизатора VyOS

4. Для проверки корректности конфигурации маршрутизатора была выполнена команда `show interfaces`.

В выводе отображается активный интерфейс **eth0** с IP-адресом `192.168.1.1/24` и соответствующим MAC-адресом.

```
vyos@vyos#  
[edit]  
vyos@vyos# show interfaces  
  ethernet eth0 {  
    address 192.168.1.1/24  
    hw-id 0c:25:cc:60:00:00  
  }  
  ethernet eth1 {  
    hw-id 0c:25:cc:60:00:01  
  }  
  ethernet eth2 {  
    hw-id 0c:25:cc:60:00:02  
  }  
  loopback lo {  
  }  
[edit]  
vyos@vyos#
```

Рис. 2.17: Просмотр интерфейсов VyOS

2.3.2 Проверка связи и анализ трафика

5. Проверка связи между устройствами выполнена с помощью команды `ping` 192.168.1.1 на **PC1-ahmedfarg**.

Устройство успешно получает ответы от маршрутизатора, что подтверждает правильность конфигурации сети.

```
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  
64 bytes from 192.168.1.1 icmp_seq=1 ttl=64 time=1.505 ms  
64 bytes from 192.168.1.1 icmp_seq=2 ttl=64 time=1.008 ms  
64 bytes from 192.168.1.1 icmp_seq=3 ttl=64 time=1.074 ms  
64 bytes from 192.168.1.1 icmp_seq=4 ttl=64 time=1.315 ms  
64 bytes from 192.168.1.1 icmp_seq=5 ttl=64 time=0.871 ms  
VPCS>
```

Рис. 2.18: Проверка связи между ПК и маршрутизатором

6. В **Wireshark** запущен захват пакетов на соединении между коммутатором и маршрутизатором.

Анализ показал корректный обмен **ARP** и **ICMP Echo Request/Reply** пакетами между устройствами.

Это демонстрирует, что сетевой стек функционирует правильно и маршрутизатор отвечает на запросы.

No.	Time	Source	Destination	Protocol	Length	Info
24	207.171678	Private_66:68:00	Broadcast	ARP	64	Who has 192.168.1.1? Tell 192.168.1.10
25	207.173006	0c:25:cc:60:00:00	Private_66:68:00	ARP	60	192.168.1.1 is at 0c:25:cc:60:00:00
26	207.174567	192.168.1.10	192.168.1.1	ICMP	98	Echo (ping) request id=0xfdd2, seq=1/256, ttl=64 (reply in 27)
27	207.175768	192.168.1.1	192.168.1.10	ICMP	98	Echo (ping) reply id=0xfdd2, seq=1/256, ttl=64 (request in 26)
28	208.177295	192.168.1.10	192.168.1.1	ICMP	98	Echo (ping) request id=0xfe2d, seq=2/512, ttl=64 (reply in 29)
29	208.177977	192.168.1.1	192.168.1.10	ICMP	98	Echo (ping) reply id=0xfe2d, seq=2/512, ttl=64 (request in 28)
30	209.179371	192.168.1.10	192.168.1.1	ICMP	98	Echo (ping) request id=0xff2d, seq=3/768, ttl=64 (reply in 31)
31	209.179982	192.168.1.1	192.168.1.10	ICMP	98	Echo (ping) reply id=0xff2d, seq=3/768, ttl=64 (request in 30)
32	210.180731	192.168.1.10	192.168.1.1	ICMP	98	Echo (ping) request id=0x002e, seq=4/1024, ttl=64 (reply in 33)
33	210.181719	192.168.1.1	192.168.1.10	ICMP	98	Echo (ping) reply id=0x002e, seq=4/1024, ttl=64 (request in 32)
34	211.183440	192.168.1.10	192.168.1.1	ICMP	98	Echo (ping) request id=0x012e, seq=5/1280, ttl=64 (reply in 35)
35	211.184039	192.168.1.1	192.168.1.10	ICMP	98	Echo (ping) reply id=0x012e, seq=5/1280, ttl=64 (request in 34)
36	212.625079	0c:25:cc:60:00:00	Private_66:68:00	ARP	60	Who has 192.168.1.10? Tell 192.168.1.1
37	212.626166	Private_66:68:00	0c:25:cc:60:00:00	ARP	60	192.168.1.10 is at 00:50:79:66:68:00

<p>Frame 35: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface -, id 0</p> <p>Ethernet II, Src: 0c:25:cc:60:00:00 (0c:25:cc:60:00:00), Dst: Private_66:68:00 (00:50:79:66:68:00)</p> <p>Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.10</p> <p>0100 = Version: 4</p> <p>.... 0101 = Header Length: 20 bytes (5)</p> <p>Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)</p> <p>Total Length: 84</p> <p>Identification: 0x175d (5981)</p> <p>0000 = Flags: 0x00</p> <p>...0 0000 0000 0000 = Fragment Offset: 0</p> <p>Time to Live: 64</p> <p>Protocol: ICMP (1)</p> <p>Header Checksum: 0xdff0 [validation disabled]</p> <p>[Header checksum status: Unverified]</p> <p>Source Address: 192.168.1.1</p> <p>Destination Address: 192.168.1.10</p> <p>[Stream index: 1]</p> <p>Internet Control Message Protocol</p> <p>Type: 0 (Echo (ping) reply)</p> <p>Code: 0</p> <p>Checksum: 0x26d9 [correct]</p> <p>[Checksum Status: Good]</p> <p>Identifier (BE): 302 (0x012e)</p> <p>Identifier (LE): 11777 (0x2e01)</p> <p>Sequence Number (BE): 5 (0x0005)</p> <p>Sequence Number (LE): 1280 (0x0500)</p> <p>[Request frame: 34]</p> <p>[Response time: 0.599 ms]</p> <p>Data (56 bytes)</p>	<p>0000 00</p> <p>0010 00</p> <p>0020 01</p> <p>0030 0e</p> <p>0040 1e</p> <p>0050 2e</p> <p>0060 3e</p>
---	--

Рис. 2.19: Анализ ICMP и ARP пакетов в Wireshark

3 Заключение

В ходе работы была выполнена настройка и запуск виртуальной машины **GNS3**, а также добавлены и сконфигурированы образы маршрутизаторов **FRR** и **VyOS**. Построены и протестированы простейшие сетевые топологии с использованием коммутатора Ethernet и виртуальных ПК.

Настроена IP-адресация, выполнена проверка связности между узлами и проведён анализ сетевого трафика с помощью **Wireshark**.