

# **Отчёт по лабораторной работе №8**

**Адресация IPv4 и IPv6. Настройка маршрутизации**

Авдадаев Джамал Геланиевич

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение задания</b>	<b>6</b>
2.1	Построение туннеля IPv6–IPv4 . . . . .	6
2.1.1	Назначение IPv6-адресов конечным устройствам . . . . .	7
2.1.2	Установка и первичная настройка маршрутизаторов VyOS .	9
2.1.3	Проверка получения маршрутизатора по IPv6 на конечных устройствах . . . . .	12
2.1.4	Проверка связности и анализ трафика . . . . .	14
2.1.5	Настройка динамической маршрутизации IPv4 (RIP) . . . . .	15
2.1.6	Проверка маршрутизации IPv4 . . . . .	17
2.1.7	Анализ трафика протокола RIP и ICMP . . . . .	18
2.1.8	Настройка туннеля . . . . .	19
2.1.9	Настройка статической маршрутизации IPv6 . . . . .	21
2.1.10	Проверка доступности конечных устройств по IPv6 . . . . .	22
2.1.11	Анализ прохождения пакетов по туннелю IPv6–IPv4 . . . . .	24
<b>3</b>	<b>Заключение</b>	<b>26</b>

# Список иллюстраций

2.1	Топология сети IPv6–IPv4 . . . . .	7
2.2	Настройка IPv6 на PC1 . . . . .	8
2.3	Настройка IPv6 на PC2 . . . . .	9
2.4	Конфигурация msk-dgavdadaev-gw-01 . . . . .	10
2.5	Конфигурация msk-dgavdadaev-gw-02 . . . . .	11
2.6	Конфигурация msk-dgavdadaev-gw-03 . . . . .	12
2.7	Проверка IPv6 на PC1 . . . . .	13
2.8	Проверка IPv6 на PC2 . . . . .	13
2.9	Проверка ping с gw-01 . . . . .	14
2.10	Анализ ICMP и ARP трафика . . . . .	15
2.11	Настройка RIP на msk-dgavdadaev-gw-01 . . . . .	16
2.12	Настройка RIP на msk-dgavdadaev-gw-02 . . . . .	16
2.13	Настройка RIP на msk-dgavdadaev-gw-03 . . . . .	17
2.14	Проверка ping после настройки RIP . . . . .	18
2.15	Анализ RIP и ICMP трафика . . . . .	19
2.16	Настройка туннеля на msk-dgavdadaev-gw-01 . . . . .	20
2.17	Настройка туннеля на msk-dgavdadaev-gw-02 . . . . .	21
2.18	Статический IPv6-маршрут на msk-dgavdadaev-gw-01 . . . . .	22
2.19	Статический IPv6-маршрут на msk-dgavdadaev-gw-02 . . . . .	22
2.20	Ping и traceroute с PC1 . . . . .	23
2.21	Ping и traceroute с PC2 . . . . .	24
2.22	Общий вид трафика IPv6 через IPv4 . . . . .	25
2.23	Детализация инкапсуляции IPv6 в IPv4 . . . . .	25

## **Список таблиц**

# 1 Цель работы

Изучение принципов маршрутизации в IPv4- и IPv6-сетях и принципов настройки сетевого оборудования.

## 2 Выполнение задания

### 2.1 Построение туннеля IPv6–IPv4

В ходе выполнения лабораторной работы в среде моделирования **GNS3** была построена сеть согласно заданной топологии, включающая три маршрутизатора **VyOS**, два Ethernet-коммутатора и два конечных устройства типа **VPCS**.

Устройства были переименованы в соответствии с требованиями задания с указанием имени учётной записи студента:

- маршрутизаторы: **msk-dgavdadaev-gw-01**, **msk-dgavdadaev-gw-02**, **msk-dgavdadaev-gw-03**;
- коммутаторы: **msk-dgavdadaev-sw-01**, **msk-dgavdadaev-sw-02**;
- конечные устройства: **PC1-dgavdadaev**, **PC2-dgavdadaev**.

Между маршрутизаторами **msk-dgavdadaev-gw-01** и **msk-dgavdadaev-gw-03** был подключён анализатор трафика для последующего изучения обмена данными на уровне протоколов IPv4 и ICMP.

Сформированная топология сети представлена на рисунке ниже.

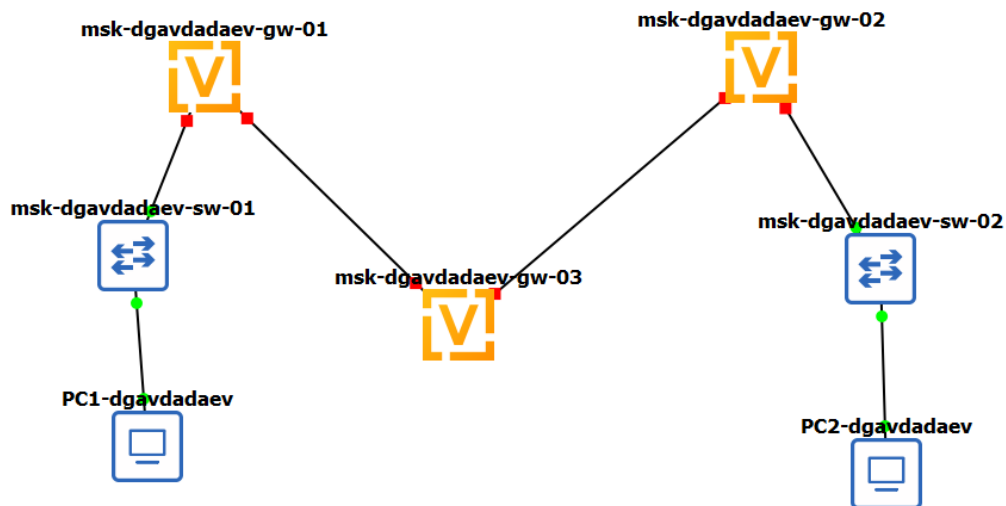
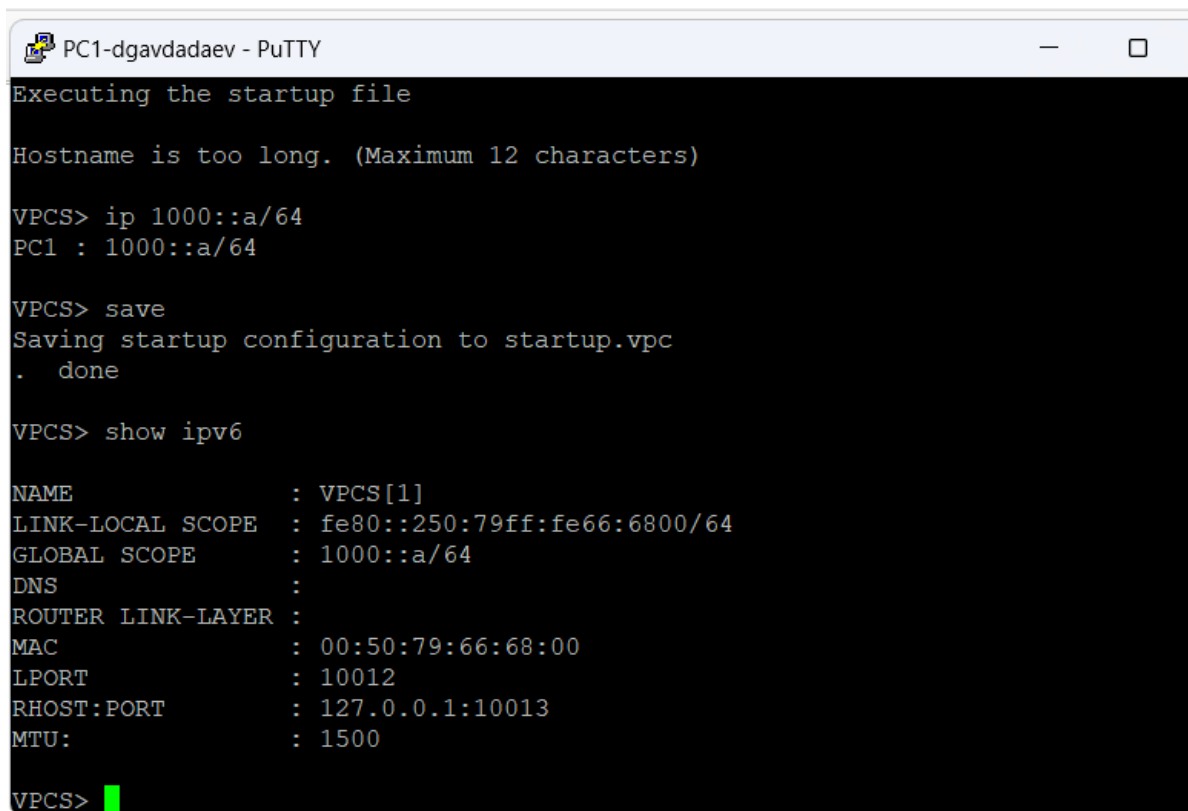


Рис. 2.1: Топология сети IPv6–IPv4

### 2.1.1 Назначение IPv6-адресов оконечным устройствам

После запуска узлов на оконечных устройствах **VPCS** были вручную настроены IPv6-адреса в соответствии с заданием.

На **PC1-dgavdadaev** был назначен глобальный IPv6-адрес **1000::a/64**, после чего конфигурация была сохранена командой `save`. Далее с помощью команды `show ipv6` была проверена корректность назначения адреса и наличие link-local адреса.



```
PC1-dgavdadaev - PuTTY
Executing the startup file

Hostname is too long. (Maximum 12 characters)

VPCS> ip 1000::a/64
PC1 : 1000::a/64

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

VPCS> show ipv6

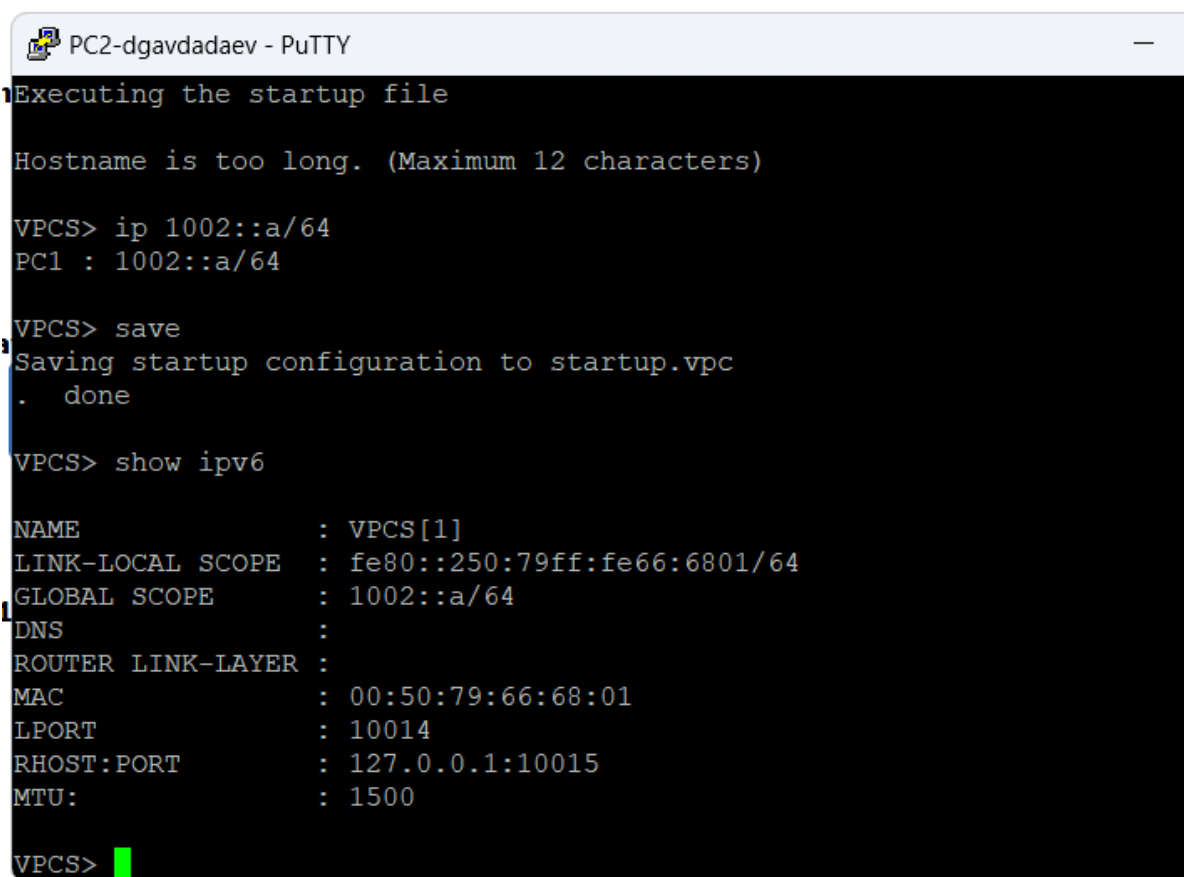
NAME                : VPCS[1]
LINK-LOCAL SCOPE    : fe80::250:79ff:fe66:6800/64
GLOBAL SCOPE        : 1000::a/64
DNS                 :
ROUTER LINK-LAYER   :
MAC                 : 00:50:79:66:68:00
LPORT               : 10012
RHOST:PORT          : 127.0.0.1:10013
MTU                 : 1500

VPCS>
```

Рис. 2.2: Настройка IPv6 на PC1

Аналогичным образом на **PC2-dgavdadaev** был настроен IPv6-адрес **1002::a/64**. После сохранения конфигурации также была выполнена проверка текущих IPv6-параметров.





```
PC2-dgavdadaev - PuTTY
Executing the startup file

Hostname is too long. (Maximum 12 characters)

VPCS> ip 1002::a/64
PC1 : 1002::a/64

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

VPCS> show ipv6

NAME                : VPCS[1]
LINK-LOCAL SCOPE    : fe80::250:79ff:fe66:6801/64
GLOBAL SCOPE        : 1002::a/64
DNS                 :
ROUTER LINK-LAYER   :
MAC                 : 00:50:79:66:68:01
LPORT               : 10014
RHOST:PORT          : 127.0.0.1:10015
MTU                 : 1500

VPCS> 
```

Рис. 2.3: Настройка IPv6 на PC2

Сообщение «*Hostname is too long (Maximum 12 characters)*», выводимое при старте VPCS, связано с внутренними ограничениями эмулятора и не влияет на работу сети.

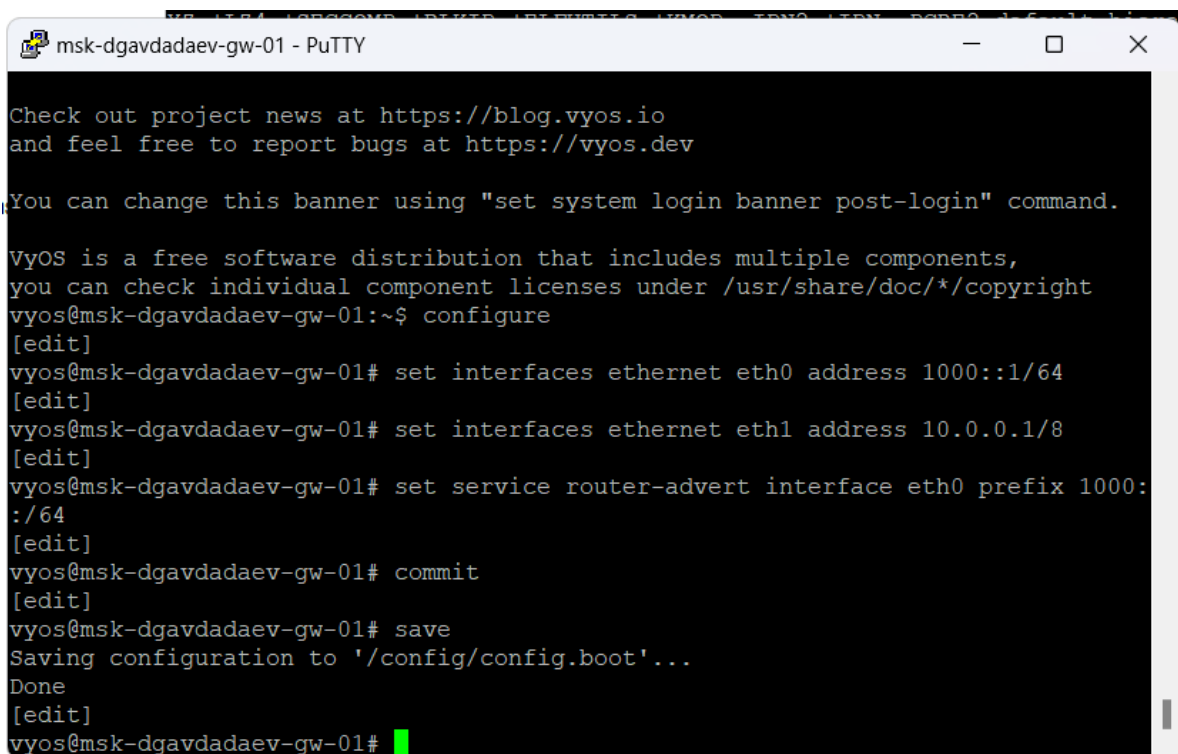
### 2.1.2 Установка и первичная настройка маршрутизаторов VyOS

На всех маршрутизаторах была выполнена установка системы **VyOS**. После завершения диалога установки устройства были перезагружены.

Далее каждый маршрутизатор был переведён в режим конфигурирования, где было изменено имя устройства в соответствии с принятой схемой именования. После внесения изменений конфигурация была применена и сохранена.

На маршрутизаторе **msk-dgavdadaev-gw-01** были настроены следующие пара-

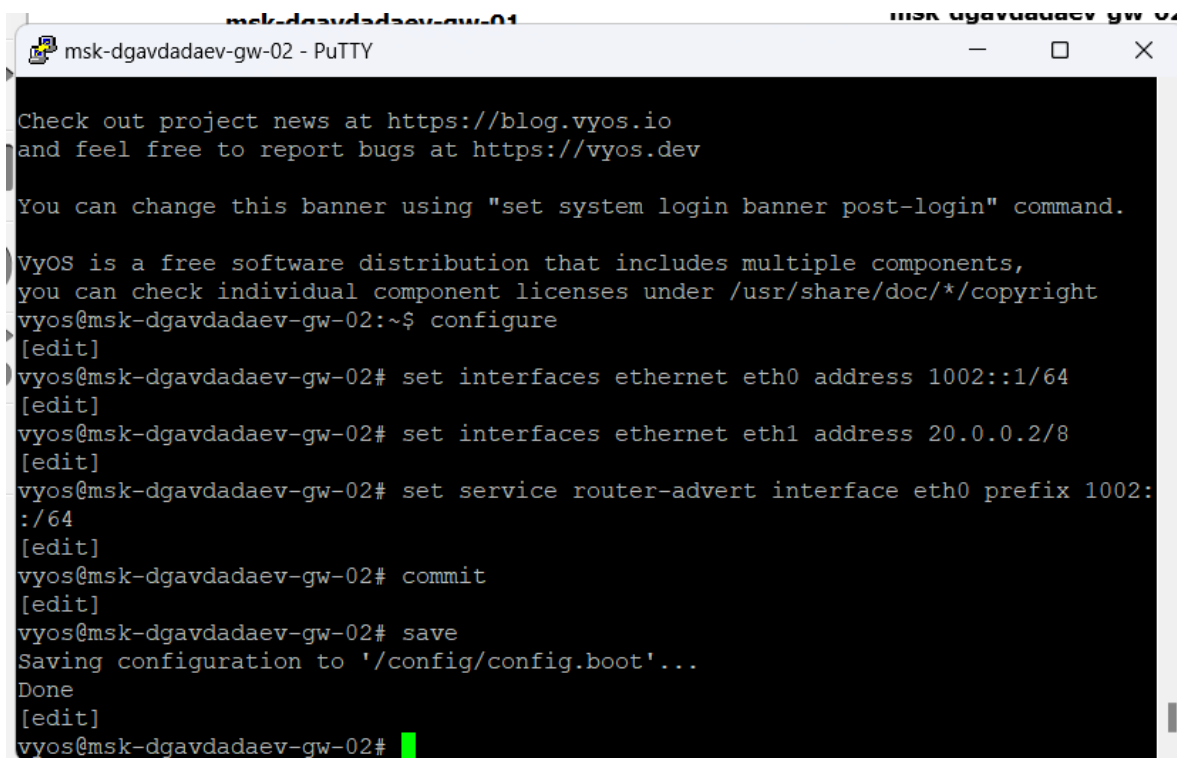
метры: - интерфейсу **eth0** назначен IPv6-адрес **1000::1/64** для связи с локальной сетью; - интерфейсу **eth1** назначен IPv4-адрес **10.0.0.1/8** для соединения с маршрутизатором **gw-03**; - включена рассылка Router Advertisement для префикса **1000::/64**.



```
msk-dgavdadaev-gw-01 - PuTTY
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@msk-dgavdadaev-gw-01:~$ configure
[edit]
vyos@msk-dgavdadaev-gw-01# set interfaces ethernet eth0 address 1000::1/64
[edit]
vyos@msk-dgavdadaev-gw-01# set interfaces ethernet eth1 address 10.0.0.1/8
[edit]
vyos@msk-dgavdadaev-gw-01# set service router-advert interface eth0 prefix 1000:
:/64
[edit]
vyos@msk-dgavdadaev-gw-01# commit
[edit]
vyos@msk-dgavdadaev-gw-01# save
Saving configuration to '/config/config.boot'...
Done
[edit]
vyos@msk-dgavdadaev-gw-01#
```

Рис. 2.4: Конфигурация msk-dgavdadaev-gw-01

На маршрутизаторе **msk-dgavdadaev-gw-02** была выполнена аналогичная настройка: - интерфейс **eth0** получил IPv6-адрес **1002::1/64**; - интерфейс **eth1** — IPv4-адрес **20.0.0.2/8**; - включена рассылка Router Advertisement для сети **1002::/64**.



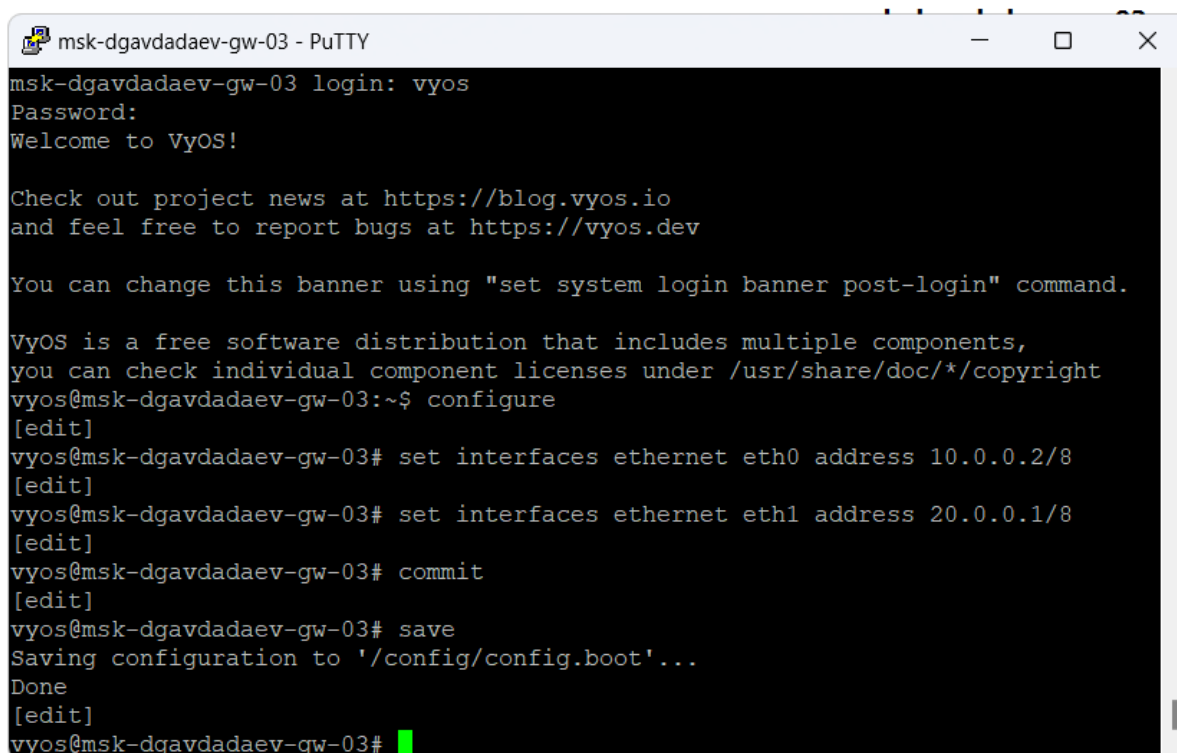
```
msk-dgavdadaev-gw-02 - PuTTY
Check out project news at https://blog.vyos.io
and feel free to report bugs at https://vyos.dev

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@msk-dgavdadaev-gw-02:~$ configure
[edit]
vyos@msk-dgavdadaev-gw-02# set interfaces ethernet eth0 address 1002::1/64
[edit]
vyos@msk-dgavdadaev-gw-02# set interfaces ethernet eth1 address 20.0.0.2/8
[edit]
vyos@msk-dgavdadaev-gw-02# set service router-advert interface eth0 prefix 1002:
:/64
[edit]
vyos@msk-dgavdadaev-gw-02# commit
[edit]
vyos@msk-dgavdadaev-gw-02# save
Saving configuration to '/config/config.boot'...
Done
[edit]
vyos@msk-dgavdadaev-gw-02#
```

Рис. 2.5: Конфигурация msk-dgavdadaev-gw-02

Маршрутизатор **msk-dgavdadaev-gw-03** выполняет роль транзитного IPv4-узла. На нём были настроены только IPv4-адреса: - **eth0** — **10.0.0.2/8**; - **eth1** — **20.0.0.1/8**.



```
msk-dgavdadaev-gw-03 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@msk-dgavdadaev-gw-03:~$ configure
[edit]
vyos@msk-dgavdadaev-gw-03# set interfaces ethernet eth0 address 10.0.0.2/8
[edit]
vyos@msk-dgavdadaev-gw-03# set interfaces ethernet eth1 address 20.0.0.1/8
[edit]
vyos@msk-dgavdadaev-gw-03# commit
[edit]
vyos@msk-dgavdadaev-gw-03# save
Saving configuration to '/config/config.boot'...
Done
[edit]
vyos@msk-dgavdadaev-gw-03#
```

Рис. 2.6: Конфигурация msk-dgavdadaev-gw-03

### 2.1.3 Проверка получения маршрутизатора по IPv6 на оконечных устройствах

После завершения настройки маршрутизаторов на оконечных устройствах была повторно выполнена команда `show ipv6`.

В выводе команд отображаются link-local адреса маршрутизаторов, что подтверждает корректную работу механизма Router Advertisement и автоматическое обнаружение шлюза по умолчанию.



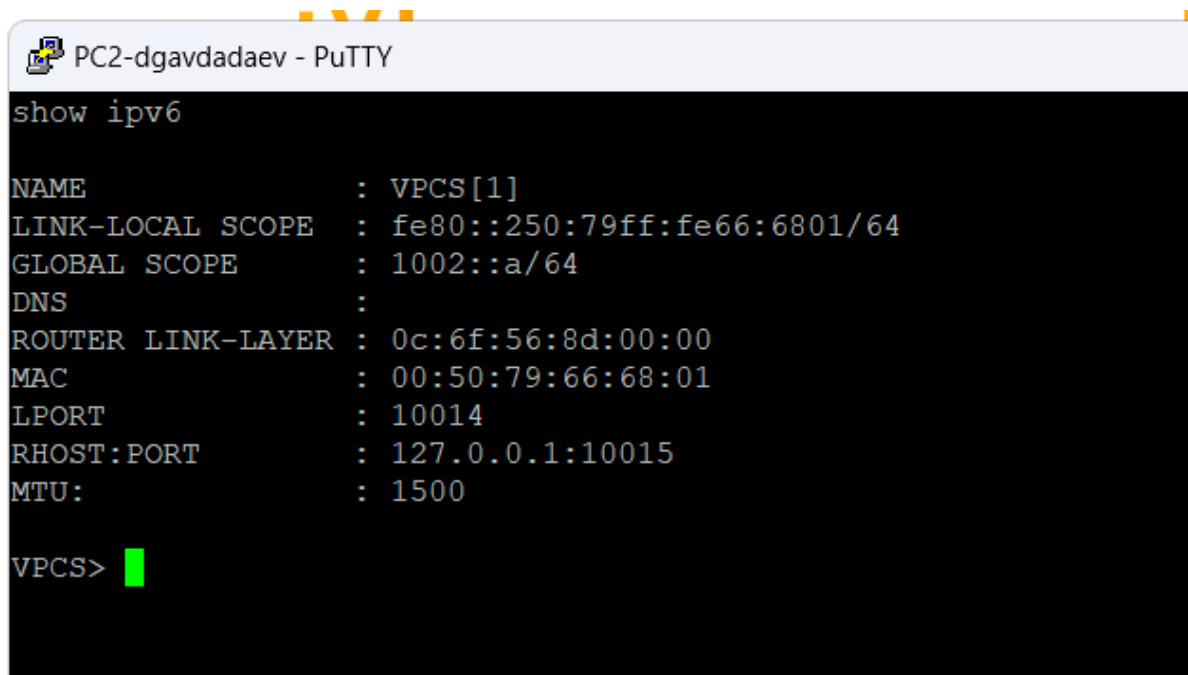
```
PC1-dgavdadaev - PuTTY

VPCS> show ipv6

NAME           : VPCS[1]
LINK-LOCAL SCOPE : fe80::250:79ff:fe66:6800/64
GLOBAL SCOPE    : 1000::a/64
DNS             :
ROUTER LINK-LAYER : 0c:5c:01:be:00:00
MAC             : 00:50:79:66:68:00
LPORT           : 10012
RHOST:PORT      : 127.0.0.1:10013
MTU:            : 1500

VPCS> 
```

Рис. 2.7: Проверка IPv6 на PC1



```
PC2-dgavdadaev - PuTTY

show ipv6

NAME           : VPCS[1]
LINK-LOCAL SCOPE : fe80::250:79ff:fe66:6801/64
GLOBAL SCOPE    : 1002::a/64
DNS             :
ROUTER LINK-LAYER : 0c:6f:56:8d:00:00
MAC             : 00:50:79:66:68:01
LPORT           : 10014
RHOST:PORT      : 127.0.0.1:10015
MTU:            : 1500

VPCS> 
```

Рис. 2.8: Проверка IPv6 на PC2

### 2.1.4 Проверка связности и анализ трафика

С маршрутизатора **msk-dgavdadaev-gw-01** была выполнена проверка IPv4-связности: - соединение с узлом **10.0.0.2** устанавливается успешно; - попытки обмена ICMP-пакетами с адресами **20.0.0.1** и **20.0.0.2** завершаются ошибкой *Network is unreachable*.

```
vyos@msk-dgavdadaev-gw-01:~$  
vyos@msk-dgavdadaev-gw-01:~$ ping 10.0.0.2  
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.  
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=1.09 ms  
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=1.98 ms  
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=1.51 ms  
^C  
--- 10.0.0.2 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 5ms  
rtt min/avg/max/mdev = 1.091/1.529/1.982/0.363 ms  
vyos@msk-dgavdadaev-gw-01:~$ ping 20.0.0.1  
connect: Network is unreachable  
vyos@msk-dgavdadaev-gw-01:~$ ping 20.0.0.2  
connect: Network is unreachable  
vyos@msk-dgavdadaev-gw-01:~$
```

Рис. 2.9: Проверка ping с gw-01

Полученный результат объясняется отсутствием маршрутов к сети **20.0.0.0/8** на маршрутизаторе **msk-dgavdadaev-gw-01**. Маршрутизация между сегментами IPv4 в данной конфигурации не настроена, поэтому доступ возможен только в пределах непосредственно подключённой сети **10.0.0.0/8**.

Для подтверждения обмена данными был проанализирован трафик в анализаторе. В захваченных пакетах наблюдаются ICMP Echo Request и Echo Reply, а также ARP-запросы и ответы, используемые для определения MAC-адресов соседних узлов.

No.	Time	Source	Destination	Protocol	Length	Info
13	5.008882	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x0849, seq=6/153
14	5.010653	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x0849, seq=6/153
15	5.277145	0c:f6:4a:e9:00:00	0c:5c:01:be:00:01	ARP	60	Who has 10.0.0.1? Tell 10.0.0.2
16	5.277389	0c:5c:01:be:00:01	0c:f6:4a:e9:00:00	ARP	60	10.0.0.1 is at 0c:5c:01:be:00:01
17	6.010888	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x0849, seq=7/179
18	6.012828	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x0849, seq=7/179
19	7.012195	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x0849, seq=8/204
20	7.012895	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x0849, seq=8/204
21	8.014522	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x0849, seq=9/230
22	8.015906	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x0849, seq=9/230
23	9.016657	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x0849, seq=10/25
24	9.017603	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x0849, seq=10/25
25	19.091609	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x0871, seq=1/256
26	19.092227	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x0871, seq=1/256
27	20.093444	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x0871, seq=2/512
28	20.094623	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x0871, seq=2/512
29	21.094823	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x0871, seq=3/768
30	21.095494	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x0871, seq=3/768

> Ethernet II, Src: 0c:f6:4a:e9:00:00 (0c:f6:4a:e9:00:00),  
 > Internet Protocol Version 4, Src: 10.0.0.2, Dst: 10.0.0.1  
 ✓ Internet Control Message Protocol  
   Type: Echo (ping) reply (0)  
   Code: 0  
   Checksum: 0xbacc [correct]  
   [Checksum Status: Good]  
   Identifier (BE): 2161 (0x0871)

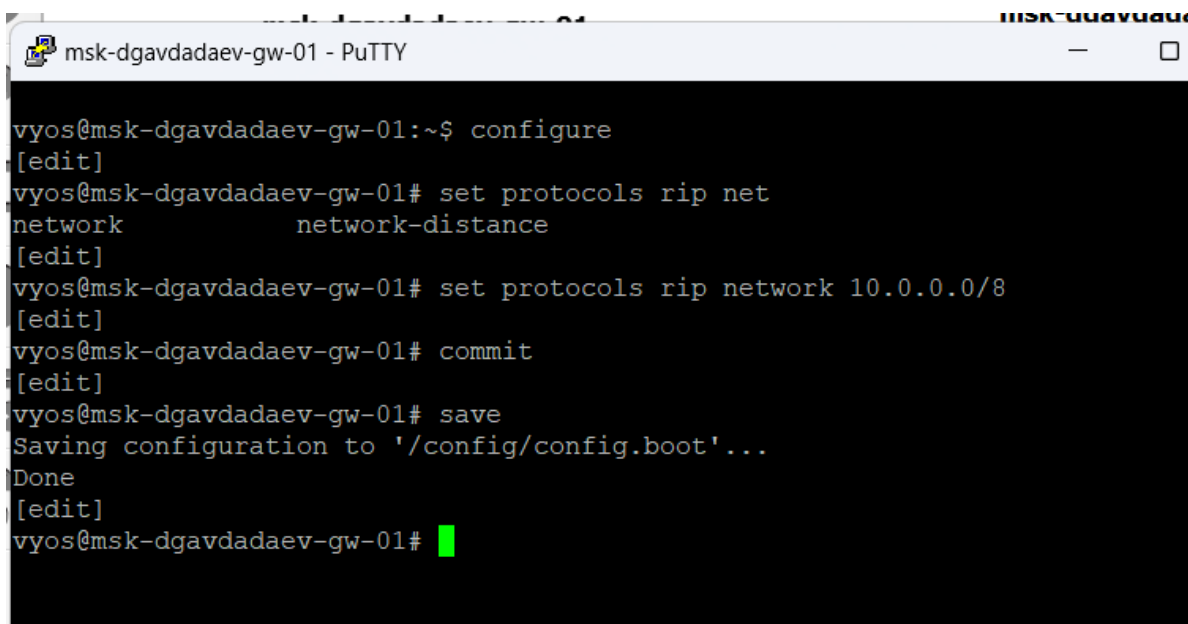
0000 0c 5c 01 be 00 01 0c f6 4a e9 00 00 08 00 45 00 -\.  
 0010 00 54 0a e0 00 00 40 01 5b c7 0a 00 00 02 0a 00 -T.  
 0020 00 01 00 00 ba cc 08 71 00 03 1d 71 84 69 00 00 -..  
 0030 00 00 d6 11 06 00 00 00 00 00 10 11 12 13 14 15 -..  
 0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 -..  
 0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 -&'(  
 0060 36 37 67

Рис. 2.10: Анализ ICMP и ARP трафика

## 2.1.5 Настройка динамической маршрутизации IPv4 (RIP)

На следующем этапе лабораторной работы была выполнена настройка динамической маршрутизации IPv4 с использованием протокола **RIP**. Протокол RIP был выбран для автоматического распространения информации о маршрутах между маршрутизаторами, соединяющими IPv4-сегменты сети.

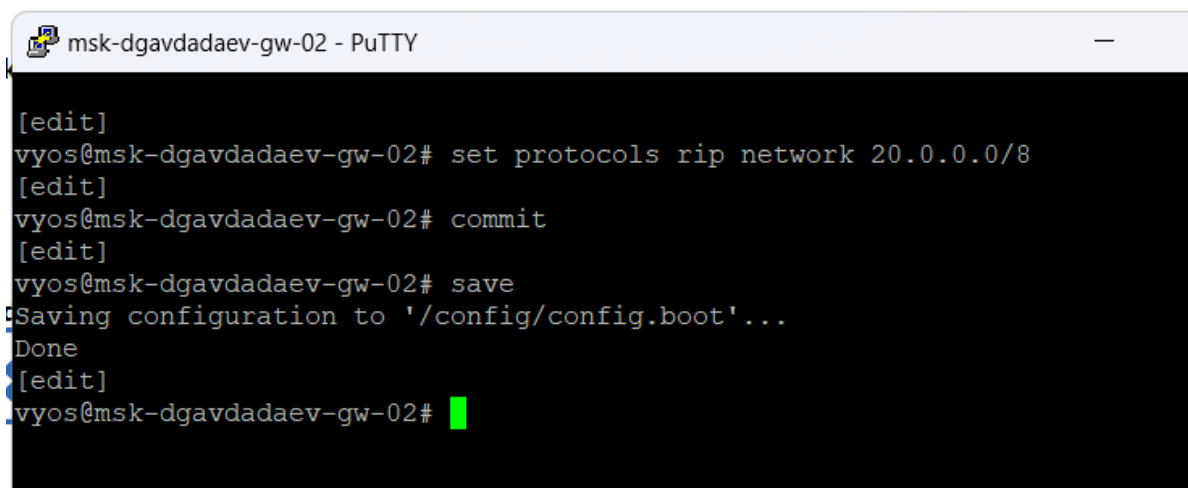
На маршрутизаторе **msk-dgavdadaev-gw-01** в конфигурации протоколов был добавлен IPv4-сегмент **10.0.0.0/8**, соответствующий сети, напрямую подключённой к интерфейсу **eth1**. После внесения изменений конфигурация была применена и сохранена.



```
vyos@msk-dgavdadaev-gw-01:~$ configure
[edit]
vyos@msk-dgavdadaev-gw-01# set protocols rip net
network                network-distance
[edit]
vyos@msk-dgavdadaev-gw-01# set protocols rip network 10.0.0.0/8
[edit]
vyos@msk-dgavdadaev-gw-01# commit
[edit]
vyos@msk-dgavdadaev-gw-01# save
Saving configuration to '/config/config.boot'...
Done
[edit]
vyos@msk-dgavdadaev-gw-01#
```

Рис. 2.11: Настройка RIP на msk-dgavdadaev-gw-01

На маршрутизаторе **msk-dgavdadaev-gw-02** был настроен протокол RIP для сети **20.0.0.0/8**, связанной с интерфейсом **eth1**. После применения и сохранения конфигурации маршрутизатор начал обмен маршрутной информацией с соседними узлами.



```
[edit]
vyos@msk-dgavdadaev-gw-02# set protocols rip network 20.0.0.0/8
[edit]
vyos@msk-dgavdadaev-gw-02# commit
[edit]
vyos@msk-dgavdadaev-gw-02# save
Saving configuration to '/config/config.boot'...
Done
[edit]
vyos@msk-dgavdadaev-gw-02#
```

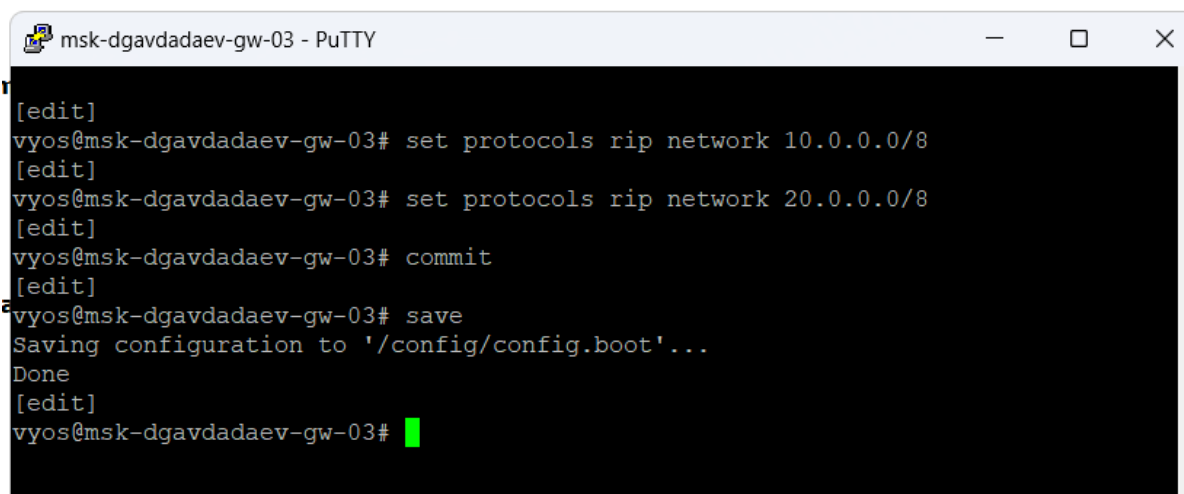
Рис. 2.12: Настройка RIP на msk-dgavdadaev-gw-02

Маршрутизатор **msk-dgavdadaev-gw-03**, выполняющий роль транзитного узла между двумя IPv4-сетями, был настроен на участие в протоколе RIP сразу для



двух сетей: - **10.0.0.0/8**; - **20.0.0.0/8**.

Это позволило обеспечить распространение маршрутной информации между сегментами сети.



```
[edit]
vyos@msk-dgavdadaev-gw-03# set protocols rip network 10.0.0.0/8
[edit]
vyos@msk-dgavdadaev-gw-03# set protocols rip network 20.0.0.0/8
[edit]
vyos@msk-dgavdadaev-gw-03# commit
[edit]
vyos@msk-dgavdadaev-gw-03# save
Saving configuration to '/config/config.boot'...
Done
[edit]
vyos@msk-dgavdadaev-gw-03#
```

Рис. 2.13: Настройка RIP на msk-dgavdadaev-gw-03

### 2.1.6 Проверка маршрутизации IPv4

После завершения настройки динамической маршрутизации была повторно выполнена проверка связности с маршрутизатора **msk-dgavdadaev-gw-01**.

В результате выполнения ICMP-запросов было установлено: - соединение с узлом **10.0.0.2** устанавливается успешно; - соединение с узлом **20.0.0.1** также устанавливается успешно; - узел **20.0.0.2** стал доступен, что свидетельствует о корректном распространении маршрутов по протоколу RIP.

```
msk-dgavdadaev-gw-01
msk-dgavdadaev-gw-01 - PuTTY

[edit]
vyos@msk-dgavdadaev-gw-01# ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=1.56 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=1.49 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=2.46 ms
^C
--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 6ms
rtt min/avg/max/mdev = 1.492/1.836/2.457/0.439 ms
[edit]
vyos@msk-dgavdadaev-gw-01# ping 20.0.0.1
PING 20.0.0.1 (20.0.0.1) 56(84) bytes of data.
64 bytes from 20.0.0.1: icmp_seq=1 ttl=64 time=1.59 ms
64 bytes from 20.0.0.1: icmp_seq=2 ttl=64 time=2.11 ms
^C
--- 20.0.0.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 3ms
rtt min/avg/max/mdev = 1.587/1.849/2.111/0.262 ms
[edit]
vyos@msk-dgavdadaev-gw-01# ping 20.0.0.2
PING 20.0.0.2 (20.0.0.2) 56(84) bytes of data.
64 bytes from 20.0.0.2: icmp_seq=1 ttl=63 time=4.05 ms
64 bytes from 20.0.0.2: icmp_seq=2 ttl=63 time=6.46 ms
^C
--- 20.0.0.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 3ms
rtt min/avg/max/mdev = 4.049/5.253/6.457/1.204 ms
[edit]
vyos@msk-dgavdadaev-gw-01#
```

Рис. 2.14: Проверка ping после настройки RIP

Подтверждение доступности всех IPv4-сегментов указывает на успешную настройку динамической маршрутизации между маршрутизаторами.

### 2.1.7 Анализ трафика протокола RIP и ICMP

Для подтверждения корректной работы протокола RIP и маршрутизации IPv4 был выполнен анализ трафика на соединении между маршрутизаторами **msk-dgavdadaev-gw-01** и **msk-dgavdadaev-gw-03**.

В захваченном трафике наблюдаются: - пакеты **RIPv2**, передаваемые на мультикаст-адрес **224.0.0.9**, используемые для обмена маршрутной ин-

формацией; - ICMP Echo Request и Echo Reply, подтверждающие успешную маршрутизацию пакетов между различными IPv4-сетями; - ARP-запросы и ответы, обеспечивающие сопоставление IP- и MAC-адресов на канальном уровне.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.2	224.0.0.9	RIPv2	66	Response
2	16.106195	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x097f, seq=1/256, ttl=64 (reply in
3	16.107161	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x097f, seq=1/256, ttl=64 (request
4	17.108633	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x097f, seq=2/512, ttl=64 (reply in
5	17.109468	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x097f, seq=2/512, ttl=64 (request
6	18.110575	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x097f, seq=3/768, ttl=64 (reply in
7	18.111881	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x097f, seq=3/768, ttl=64 (request
8	21.374460	0c:f6:4a:e9:00:00	0c:5c:01:be:00:01	ARP	60	Who has 10.0.0.1? Tell 10.0.0.2
9	21.377104	0c:5c:01:be:00:01	0c:f6:4a:e9:00:00	ARP	60	10.0.0.1 is at 0c:5c:01:be:00:01
10	21.409632	0c:5c:01:be:00:01	0c:f6:4a:e9:00:00	ARP	60	Who has 10.0.0.2? Tell 10.0.0.1
11	21.410577	0c:f6:4a:e9:00:00	0c:5c:01:be:00:01	ARP	60	10.0.0.2 is at 0c:f6:4a:e9:00:00
12	22.936415	10.0.0.1	20.0.0.1	ICMP	98	Echo (ping) request id=0x0980, seq=1/256, ttl=64 (reply in
13	22.937379	20.0.0.1	10.0.0.1	ICMP	98	Echo (ping) reply id=0x0980, seq=1/256, ttl=64 (request
14	23.938829	10.0.0.1	20.0.0.1	ICMP	98	Echo (ping) request id=0x0980, seq=2/512, ttl=64 (reply in
15	23.939911	20.0.0.1	10.0.0.1	ICMP	98	Echo (ping) reply id=0x0980, seq=2/512, ttl=64 (request
16	28.072011	10.0.0.1	20.0.0.2	ICMP	98	Echo (ping) request id=0x0981, seq=1/256, ttl=64 (reply in
17	28.075541	20.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x0981, seq=1/256, ttl=63 (request
18	29.074066	10.0.0.1	20.0.0.2	ICMP	98	Echo (ping) request id=0x0981, seq=2/512, ttl=64 (reply in
19	29.079071	20.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x0981, seq=2/512, ttl=63 (request
20	33.002513	10.0.0.2	224.0.0.9	RIPv2	66	Response
21	63.004054	10.0.0.2	224.0.0.9	RIPv2	66	Response

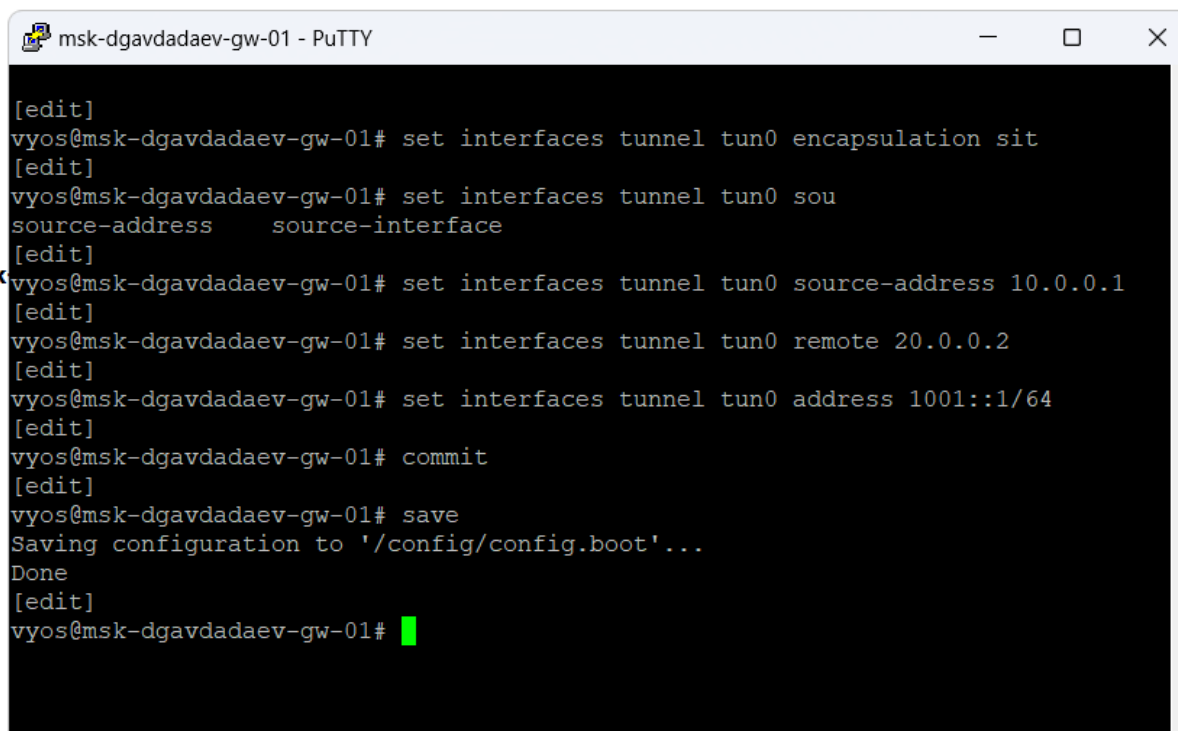
> Frame 1: Packet, 66 bytes on wire (528 bits), 66 bytes captured (512 bits) on interface 0	0000	01 00 5e 00 00 09 0c f6 4a e9 00 00 08 00 45 c0	..^.....J.
> Ethernet II, Src: 0c:f6:4a:e9:00:00 (0c:f6:4a:e9:00:00), Dst: IPv4mcast (01:00:5e:00:00:09)	0010	00 34 30 03 40 00 01 11 5e eb 0a 00 00 02 e0 00	..40.@...^.
> Internet Protocol Version 4, Src: 10.0.0.2, Dst: 224.0.0.9	0020	00 09 02 08 02 08 00 20 fc 8c 02 02 00 00 00 02	.....
> User Datagram Protocol, Src Port: 520, Dst Port: 520	0030	00 00 14 00 00 00 ff 00 00 00 00 00 00 00 00 00	.....
> Routing Information Protocol	0040	00 01	..

Рис. 2.15: Анализ RIPv2 и ICMP трафика

## 2.1.8 Настройка туннеля

На маршрутизаторе **msk-dgavdadaev-gw-01** был создан туннельный интерфейс **tun0** со следующими параметрами: - тип инкапсуляции — **SIT**; - IPv4-адрес источника — **10.0.0.1**; - IPv4-адрес удалённого узла — **20.0.0.2**; - IPv6-адрес туннельного интерфейса — **1001::1/64**.

После применения и сохранения конфигурации туннельный интерфейс был успешно активирован.

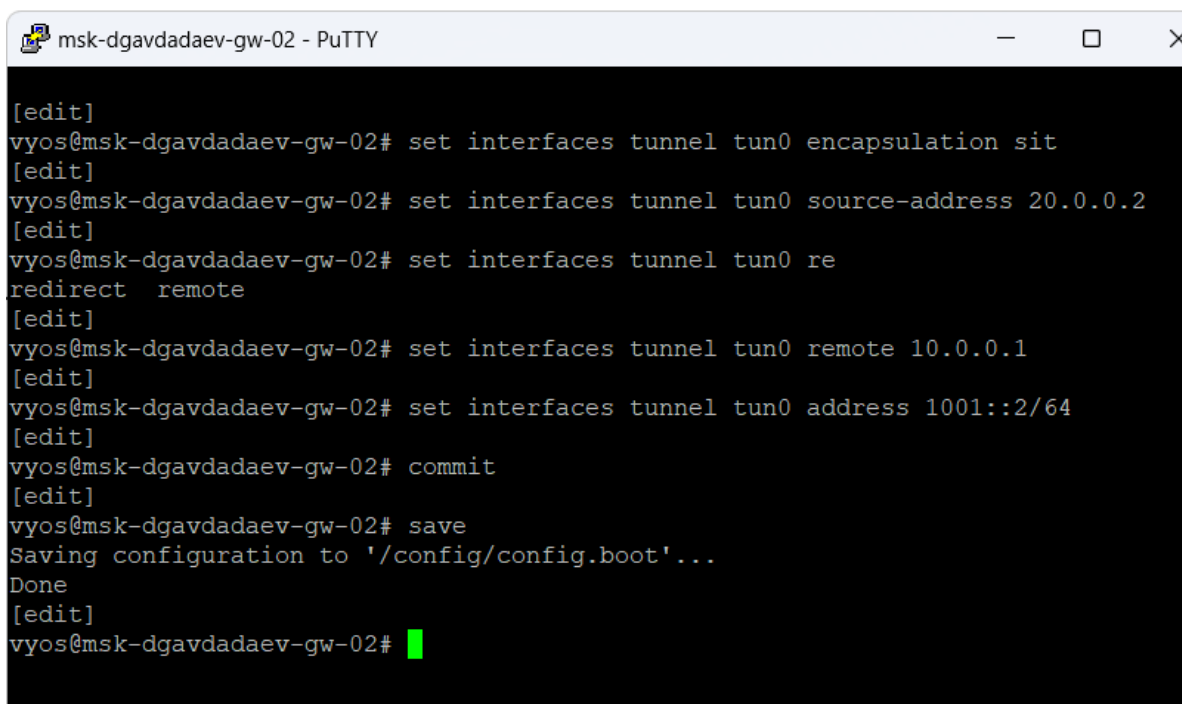


```
[edit]
vyos@msk-dgavdadaev-gw-01# set interfaces tunnel tun0 encapsulation sit
[edit]
vyos@msk-dgavdadaev-gw-01# set interfaces tunnel tun0 sou
source-address      source-interface
[edit]
vyos@msk-dgavdadaev-gw-01# set interfaces tunnel tun0 source-address 10.0.0.1
[edit]
vyos@msk-dgavdadaev-gw-01# set interfaces tunnel tun0 remote 20.0.0.2
[edit]
vyos@msk-dgavdadaev-gw-01# set interfaces tunnel tun0 address 1001::1/64
[edit]
vyos@msk-dgavdadaev-gw-01# commit
[edit]
vyos@msk-dgavdadaev-gw-01# save
Saving configuration to '/config/config.boot'...
Done
[edit]
vyos@msk-dgavdadaev-gw-01#
```

Рис. 2.16: Настройка туннеля на msk-dgavdadaev-gw-01

На маршрутизаторе **msk-dgavdadaev-gw-02** была выполнена симметричная настройка туннеля **tun0**: - тип инкапсуляции — **SIT**; - IPv4-адрес источника — **20.0.0.2**; - IPv4-адрес удалённого узла — **10.0.0.1**; - IPv6-адрес туннельного интерфейса — **1001::2/64**.

Данная конфигурация обеспечивает логическое IPv6-соединение между двумя маршрутизаторами поверх IPv4-инфраструктуры.



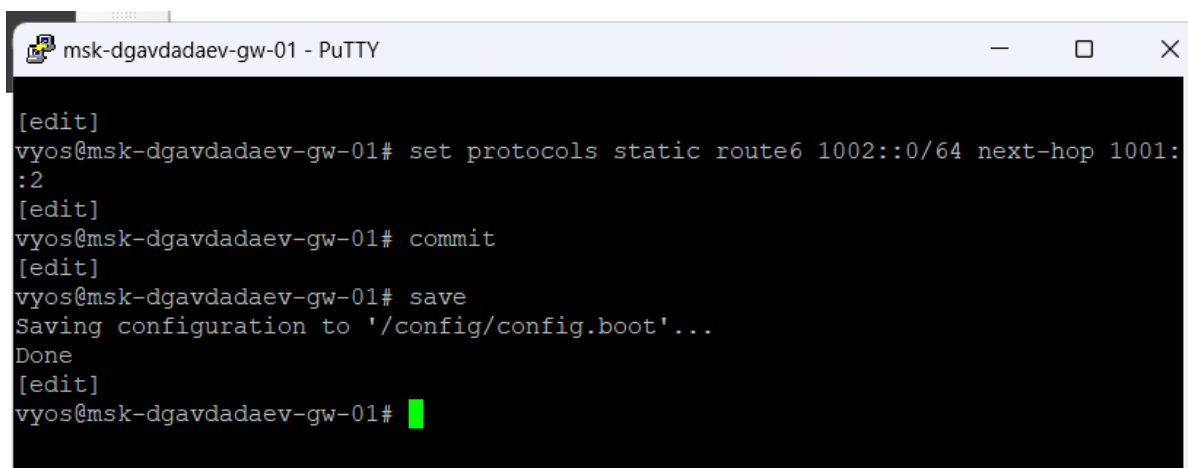
```
[edit]
vyos@msk-dgavdadaev-gw-02# set interfaces tunnel tun0 encapsulation sit
[edit]
vyos@msk-dgavdadaev-gw-02# set interfaces tunnel tun0 source-address 20.0.0.2
[edit]
vyos@msk-dgavdadaev-gw-02# set interfaces tunnel tun0 re
redirect remote
[edit]
vyos@msk-dgavdadaev-gw-02# set interfaces tunnel tun0 remote 10.0.0.1
[edit]
vyos@msk-dgavdadaev-gw-02# set interfaces tunnel tun0 address 1001::2/64
[edit]
vyos@msk-dgavdadaev-gw-02# commit
[edit]
vyos@msk-dgavdadaev-gw-02# save
Saving configuration to '/config/config.boot'...
Done
[edit]
vyos@msk-dgavdadaev-gw-02#
```

Рис. 2.17: Настройка туннеля на msk-dgavdadaev-gw-02

### 2.1.9 Настройка статической маршрутизации IPv6

После создания туннеля была выполнена настройка статической маршрутизации IPv6 для обеспечения сквозной связности между IPv6-сегментами.

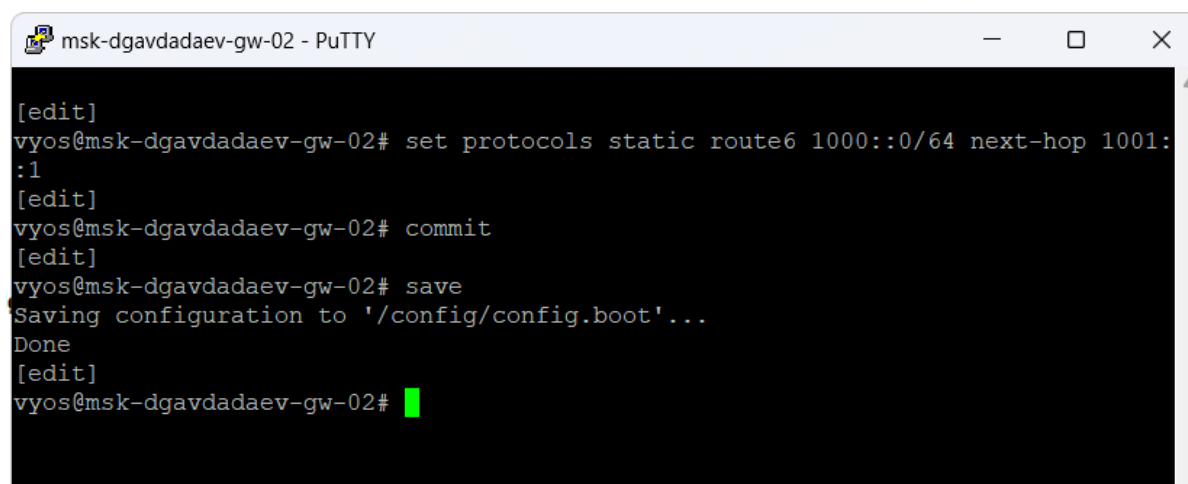
На маршрутизаторе **msk-dgavdadaev-gw-01** был добавлен маршрут к сети **1002::/64** с указанием следующего перехода через туннельный адрес **1001::2**.



```
[edit]
vyos@msk-dgavdadaev-gw-01# set protocols static route6 1002::0/64 next-hop 1001::2
[edit]
vyos@msk-dgavdadaev-gw-01# commit
[edit]
vyos@msk-dgavdadaev-gw-01# save
Saving configuration to '/config/config.boot'...
Done
[edit]
vyos@msk-dgavdadaev-gw-01#
```

Рис. 2.18: Статический IPv6-маршрут на msk-dgavdadaev-gw-01

На маршрутизаторе **msk-dgavdadaev-gw-02** был настроен статический маршрут к сети **1000::/64** через следующий переход **1001::1**.



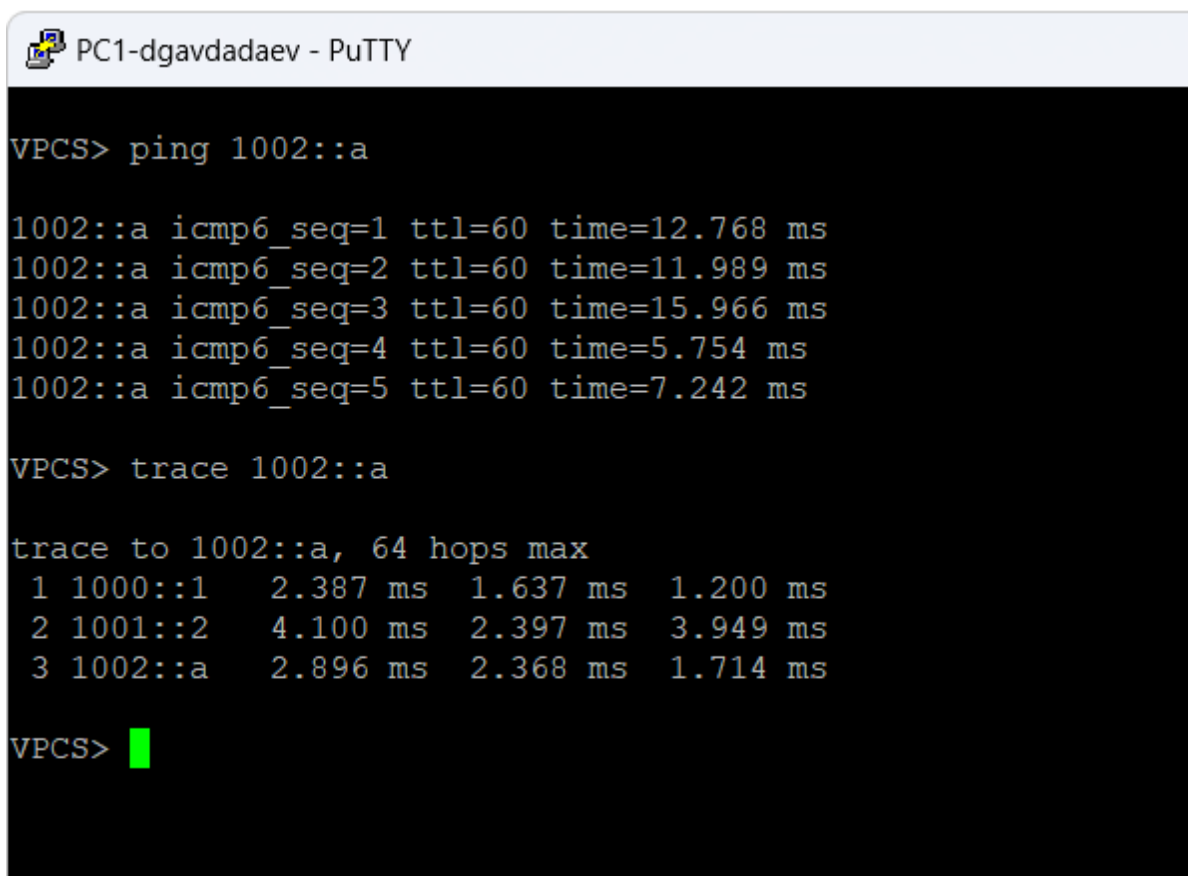
```
[edit]
vyos@msk-dgavdadaev-gw-02# set protocols static route6 1000::0/64 next-hop 1001::1
[edit]
vyos@msk-dgavdadaev-gw-02# commit
[edit]
vyos@msk-dgavdadaev-gw-02# save
Saving configuration to '/config/config.boot'...
Done
[edit]
vyos@msk-dgavdadaev-gw-02#
```

Рис. 2.19: Статический IPv6-маршрут на msk-dgavdadaev-gw-02

### 2.1.10 Проверка доступности конечных устройств по IPv6

После завершения настройки туннеля и маршрутизации была подтверждена IPv6-доступность между конечными устройствами.

С узла **PC1-dgavdadaev** была выполнена проверка связности с **PC2-dgavdadaev**. ICMPv6-запросы успешно достигали адреса **1002::a**, а трассировка показала прохождение пакетов через IPv6-шлюзы и туннельный интерфейс.



```
PC1-dgavdadaev - PuTTY

VPCS> ping 1002::a

1002::a icmp6_seq=1 ttl=60 time=12.768 ms
1002::a icmp6_seq=2 ttl=60 time=11.989 ms
1002::a icmp6_seq=3 ttl=60 time=15.966 ms
1002::a icmp6_seq=4 ttl=60 time=5.754 ms
1002::a icmp6_seq=5 ttl=60 time=7.242 ms

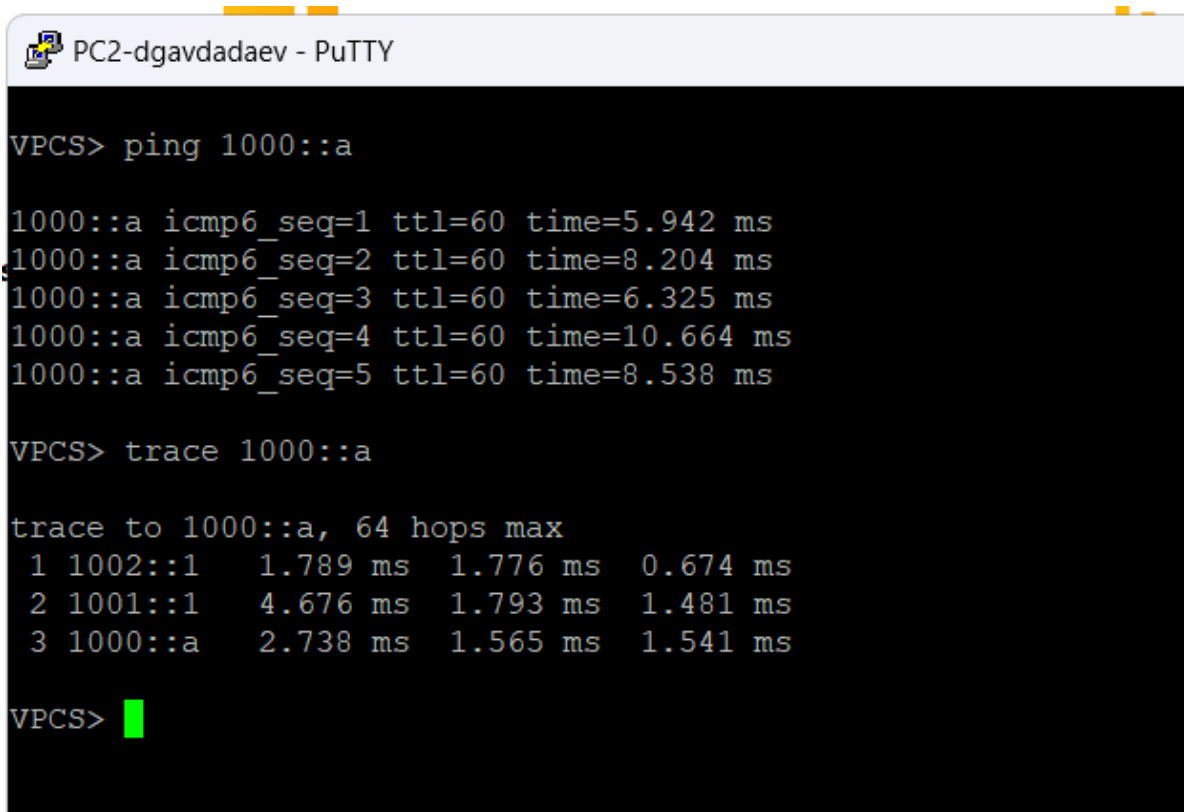
VPCS> trace 1002::a

trace to 1002::a, 64 hops max
 1 1000::1    2.387 ms   1.637 ms   1.200 ms
 2 1001::2    4.100 ms   2.397 ms   3.949 ms
 3 1002::a    2.896 ms   2.368 ms   1.714 ms

VPCS> █
```

Рис. 2.20: Ping и traceroute с PC1

Аналогичная проверка была выполнена с узла **PC2-dgavdadaev** в направлении **PC1-dgavdadaev**. Результаты подтверждают симметричную доступность IPv6-узлов.



```
VPCS> ping 1000::a

1000::a icmp6_seq=1 ttl=60 time=5.942 ms
1000::a icmp6_seq=2 ttl=60 time=8.204 ms
1000::a icmp6_seq=3 ttl=60 time=6.325 ms
1000::a icmp6_seq=4 ttl=60 time=10.664 ms
1000::a icmp6_seq=5 ttl=60 time=8.538 ms

VPCS> trace 1000::a

trace to 1000::a, 64 hops max
 1 1002::1    1.789 ms  1.776 ms  0.674 ms
 2 1001::1    4.676 ms  1.793 ms  1.481 ms
 3 1000::a    2.738 ms  1.565 ms  1.541 ms

VPCS>
```

Рис. 2.21: Ping и traceroute с PC2

### 2.1.11 Анализ прохождения пакетов по туннелю IPv6–IPv4

Для детального изучения работы туннеля был проанализирован трафик с использованием анализатора, подключённого к каналу между маршрутизаторами **msk-dgavdadaev-gw-01** и **msk-dgavdadaev-gw-03**.

В захваченном трафике наблюдается следующая картина: - исходные IPv6-пакеты (ICMPv6 Echo Request / Echo Reply) между адресами **1000::a** и **1002::a**; - инкапсуляция IPv6-пакетов внутрь IPv4-пакетов с протоколом **41**; - в заголовке IPv4 указываются адреса **20.0.0.2** и **10.0.0.1**, соответствующие концам туннеля; - полезная нагрузка IPv4-пакета содержит полный IPv6-заголовок и ICMPv6-сообщение.



No.	Time	Source	Destination	Protocol	Length	Info
23	13.496483	1002::a	1000::a	ICMPv6	194	Destination Unreachable (Port unreachable)[Malformed Pack
24	13.497840	1000::a	1002::a	UDP	146	43184 → 43185 Len=64
25	13.499439	1002::a	1000::a	ICMPv6	194	Destination Unreachable (Port unreachable)[Malformed Pack
26	13.501231	1000::a	1002::a	UDP	146	43184 → 43185 Len=64
27	13.502351	1002::a	1000::a	ICMPv6	194	Destination Unreachable (Port unreachable)[Malformed Pack
28	39.898232	10.0.0.2	224.0.0.9	RIPv2	66	Response
29	47.318765	1002::a	1000::a	ICMPv6	138	Echo (ping) request id=0xca73, seq=1, hop limit=63 (reply
30	47.319559	1000::a	1002::a	ICMPv6	138	Echo (ping) reply id=0xca73, seq=1, hop limit=61 (request
31	48.324749	1002::a	1000::a	ICMPv6	138	Echo (ping) request id=0xca73, seq=2, hop limit=63 (reply
32	48.327373	1000::a	1002::a	ICMPv6	138	Echo (ping) reply id=0xca73, seq=2, hop limit=61 (request
33	49.334818	1002::a	1000::a	ICMPv6	138	Echo (ping) request id=0xca73, seq=3, hop limit=63 (reply
34	49.337270	1000::a	1002::a	ICMPv6	138	Echo (ping) reply id=0xca73, seq=3, hop limit=61 (request
35	50.341896	1002::a	1000::a	ICMPv6	138	Echo (ping) request id=0xca73, seq=4, hop limit=63 (reply
36	50.346467	1000::a	1002::a	ICMPv6	138	Echo (ping) reply id=0xca73, seq=4, hop limit=61 (request
37	51.352965	1002::a	1000::a	ICMPv6	138	Echo (ping) request id=0xca73, seq=5, hop limit=63 (reply
38	51.356307	1000::a	1002::a	ICMPv6	138	Echo (ping) reply id=0xca73, seq=5, hop limit=61 (request
39	52.757358	0c:f6:4a:e9:00:00	0c:5c:01:be:00:01	ARP	60	Who has 10.0.0.1? Tell 10.0.0.2
40	52.758010	0c:5c:01:be:00:01	0c:f6:4a:e9:00:00	ARP	60	10.0.0.1 is at 0c:5c:01:be:00:01
41	52.790744	0c:5c:01:be:00:01	0c:f6:4a:e9:00:00	ARP	60	Who has 10.0.0.2? Tell 10.0.0.1
42	52.791451	0c:f6:4a:e9:00:00	0c:5c:01:be:00:01	ARP	60	10.0.0.2 is at 0c:f6:4a:e9:00:00
43	58.219091	1002::a	1000::a	UDP	146	47152 → 47153 Len=64
44	58.220248	1001::1	1002::a	ICMPv6	194	Time Exceeded (Hop limit exceeded in transit)

Рис. 2.22: Общий вид трафика IPv6 через IPv4

```

> Frame 29: Packet, 138 bytes on wire (1104 bits), 138 bytes captured (1104 bits) on interface -, id 0
> Ethernet II, Src: 0c:f6:4a:e9:00:00 (0c:f6:4a:e9:00:00), Dst: 0c:5c:01:be:00:01 (0c:5c:01:be:00:01)
▼ Internet Protocol Version 4, Src: 20.0.0.2, Dst: 10.0.0.1
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
        Total Length: 124
        Identification: 0x58d9 (22745)
    > 010. .... = Flags: 0x2, Don't fragment
        ...0 0000 0000 0000 = Fragment Offset: 0
        Time to Live: 63
        Protocol: IPv6 (41)
        Header Checksum: 0xc47d [validation disabled]
        [Header checksum status: Unverified]
        Source Address: 20.0.0.2
        Destination Address: 10.0.0.1
        [Stream index: 0]
▼ Internet Protocol Version 6, Src: 1002::a, Dst: 1000::a
    0110 .... = Version: 6
    > .... 0000 0000 .... = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)
    > .... 0000 0000 0000 0000 = Flow Label: 0x000000
        Payload Length: 64
        Next Header: ICMPv6 (58)
        Hop Limit: 63
    > Source Address: 1002::a
    > Destination Address: 1000::a
        [Stream index: 0]
▼ Internet Control Message Protocol v6
    Type: Echo (ping) request (128)
    Code: 0
    Checksum: 0x9de7 [correct]

```

Рис. 2.23: Детализация инкапсуляции IPv6 в IPv4

## 3 Заключение

В ходе выполнения лабораторной работы была изучена и практически реализована технология взаимодействия сетей IPv4 и IPv6 с использованием туннелирования в среде моделирования **GNS3**. Была построена составная сетевая топология с несколькими маршрутизаторами **VyOS**, оконечными устройствами **VPCS** и Ethernet-коммутаторами, а также выполнена их корректная адресация и настройка в соответствии с заданием.

В процессе работы была настроена динамическая маршрутизация IPv4 с использованием протокола **RIP**, что обеспечило автоматическое распространение маршрутной информации между сегментами сети. После этого был создан туннель **IPv6-over-IPv4 (SIT)**, позволивший организовать логическое соединение между удалёнными IPv6-сетями поверх существующей IPv4-инфраструктуры. Для обеспечения связности IPv6-сегментов была дополнительно настроена статическая маршрутизация IPv6.

Проверка доступности оконечных устройств подтвердила корректность настройки туннеля и маршрутизации: ICMPv6-пакеты успешно передавались между узлами, а результаты трассировки показали прохождение трафика через туннельные интерфейсы. Анализ трафика с использованием **Wireshark** позволил наглядно проследить процесс инкапсуляции IPv6-пакетов в IPv4-пакеты с протоколом 41, а также подтвердить корректную работу протоколов **ARP**, **ICMP**, **ICMPv6** и **RIP**.