

Отчёт по лабораторной работе 6

Адресация IPv4 и IPv6. Двойной стек

Сейдалиев Тагиетдин Ровшенович

Содержание

1 Цель работы	5
2 Разбиение IPv4-сетей на подсети	6
2.1 Сеть 172.16.20.0/24	6
2.1.1 Основные характеристики исходной сети	6
2.1.2 Разбиение на три подсети (126, 62, 62 узла)	6
2.2 Сеть 10.10.1.64/26	8
2.2.1 Характеристики исходной сети 10.10.1.64/26	8
2.2.2 Выделение подсети на 30 узлов внутри 10.10.1.64/26	8
2.3 Сеть 10.10.1.0/26	9
2.3.1 Характеристики сети 10.10.1.0/26	9
2.3.2 Выделение подсети на 14 узлов	9
3 Разбиение IPv6-сети на подсети	11
3.1 Сеть 2001:db8:c0de::/48	11
3.1.1 Характеристика исходной сети	11
3.1.2 Разбиение на 2 подсети с использованием идентификатора подсети	12
3.1.3 Разбиение на 2 подсети с использованием идентификатора интерфейса	12
3.2 Сеть 2a02:6b8::/64	14
3.2.1 Характеристика исходной сети	14
3.2.2 Разбиение на 2 подсети с использованием идентификатора подсети	14
3.2.3 Разбиение на 2 подсети с использованием идентификатора интерфейса	15
4 Выполнение	17
4.1 Настройка двойного стека адресации IPv4 и IPv6 в локальной сети	17
4.2 Задание для самостоятельного выполнения	28
5 Заключение	34

Список иллюстраций

4.1 Топология лабораторного стенда	17
4.2 Конфигурация PC1	18
4.3 Конфигурация PC2	18
4.4 Конфигурация сервера	19
4.5 Процесс настройки интерфейсов FRR	20
4.6 Вывод running-config	20
4.7 Информация об интерфейсах FRR	21
4.8 Ping и trace с PC1	21
4.9 ARP-трафик	22
4.10 ICMP-ответ сервера	23
4.11 Конфигурация PC3	24
4.12 Конфигурация PC4	24
4.13 IPv6-конфигурация сервера	25
4.14 Назначение IPv6-адресов на VyOS	25
4.15 Вывод show interfaces на VyOS	26
4.16 Ping и trace с PC4	27
4.17 Ping с сервера IPv6 и IPv4	27
4.18 Фрагмент ICMPv6-трафика в Wireshark	28
4.19 PC1 вывод show ip	30
4.20 PC2 вывод show ip	30
4.21 Топология сети	31
4.22 Настройка VyOS интерфейсов	31
4.23 Ping и trace с PC1	32
4.24 IPv6 ICMP проверка	33
4.25 Проверка IPv4 с PC1	33

Список таблиц

1 Цель работы

Изучение принципов распределения и настройки адресного пространства на устройствах сети.

2 Разбиение IPv4-сетей на подсети

2.1 Сеть 172.16.20.0/24

2.1.1 Основные характеристики исходной сети

- Префикс: /24
- Маска: 255.255.255.0
- Адрес сети: 172.16.20.0
- Broadcast-адрес: 172.16.20.255
- Число адресов в сети: $2^{(32-24)} = 2^8 = 256$
- Число адресов узлов: $256 - 2 = 254$
- Диапазон адресов узлов: 172.16.20.1 – 172.16.20.254

Если рассматривать эту сеть как часть класса В (исходный префикс /16), то число возможных подсетей с префиксом /24: - Количество подсетей: $2^{(24-16)} = 256$

2.1.2 Разбиение на три подсети (126, 62, 62 узла)

Требуемое количество узлов:

- Подсеть 1: 126 узлов → нужно минимум 7 бит для хостов
 $2^7 - 2 = 126 \rightarrow$ префикс /25
- Подсеть 2: 62 узла → нужно 6 бит
 $2^6 - 2 = 62 \rightarrow$ префикс /26

- Подсеть 3: 62 узла → тоже /26

Размещаем подсети последовательно внутри 172.16.20.0/24 (от меньшего адреса к большему).

2.1.2.1 Подсеть 1 (126 узлов)

- Адрес сети: **172.16.20.0/25**
- Маска: **255.255.255.128**
- Диапазон узлов: **172.16.20.1 – 172.16.20.126**
- Broadcast: **172.16.20.127**
- Число адресов узлов: **126**

2.1.2.2 Подсеть 2 (62 узла)

Оставшаяся часть исходной сети – 172.16.20.128/25. Делим её на две /26.

Первая /26:

- Адрес сети: **172.16.20.128/26**
- Маска: **255.255.255.192**
- Диапазон узлов: **172.16.20.129 – 172.16.20.190**
- Broadcast: **172.16.20.191**
- Число адресов узлов: **62**

2.1.2.3 Подсеть 3 (62 узла)

Вторая /26 из того же диапазона:

- Адрес сети: **172.16.20.192/26**
- Маска: **255.255.255.192**
- Диапазон узлов: **172.16.20.193 – 172.16.20.254**
- Broadcast: **172.16.20.255**
- Число адресов узлов: **62**

2.2 Сеть 10.10.1.64/26

2.2.1 Характеристики исходной сети 10.10.1.64/26

- Префикс: /26
- Маска: 255.255.255.192
- Адрес сети: 10.10.1.64
- Broadcast-адрес: 10.10.1.127
- Число адресов в сети: $2^{(32-26)} = 2^6 = 64$
- Число адресов узлов: $64 - 2 = 62$
- Диапазон адресов узлов: 10.10.1.65 – 10.10.1.126

Если считать, что базовая сеть – 10.10.1.0/24, то с префиксом /26 это разбиение на:
- Количество возможных подсетей: $2^{(26-24)} = 4$ подсети
(10.10.1.0/26, 10.10.1.64/26, 10.10.1.128/26, 10.10.1.192/26)

2.2.2 Выделение подсети на 30 узлов внутри 10.10.1.64/26

Нужно 30 узлов $\rightarrow 2^5 - 2 = 30 \rightarrow$ префикс /27.

Сеть 10.10.1.64/26 (64–127) делится на две /27:

1. 10.10.1.64/27 (64–95)

2. 10.10.1.96/27 (96–127)

Выберем первую, с минимальным адресом.

Искомая подсеть (на 30 узлов):

- Адрес сети: 10.10.1.64/27
- Маска: 255.255.255.224
- Число адресов в подсети: $2^{(32-27)} = 32$
- Число адресов узлов: $32 - 2 = 30$
- Диапазон узлов: 10.10.1.65 – 10.10.1.94
- Broadcast: 10.10.1.95

2.3 Сеть 10.10.1.0/26

2.3.1 Характеристики сети 10.10.1.0/26

- Префикс: /26
- Маска: 255.255.255.192
- Адрес сети: 10.10.1.0
- Broadcast-адрес: 10.10.1.63
- Число адресов в сети: $2^{(32-26)} = 64$
- Число адресов узлов: $64 - 2 = 62$
- Диапазон адресов узлов: 10.10.1.1 – 10.10.1.62

В пределах 10.10.1.0/24 при маске /26 также возможны **4 подсети** (0/26, 64/26, 128/26, 192/26).

2.3.2 Выделение подсети на 14 узлов

Нужно 14 узлов $\rightarrow 2^4 - 2 = 14 \rightarrow$ префикс /28.

Сеть 10.10.1.0/26 (диапазон 0–63) делится на четыре /28:

1. 10.10.1.0/28 (0–15)
2. 10.10.1.16/28 (16–31)
3. 10.10.1.32/28 (32–47)
4. 10.10.1.48/28 (48–63)

Выберем первую.

Искомая подсеть (на 14 узлов):

- Адрес сети: 10.10.1.0/28

- Маска: **255.255.255.240**
- Число адресов в подсети: 16
- Число адресов узлов: $16 - 2 = \mathbf{14}$
- Диапазон узлов: **10.10.1.1 – 10.10.1.14**
- Broadcast: **10.10.1.15**

3 Разбиение IPv6-сети на подсети

3.1 Сеть 2001:db8:c0de::/48

3.1.1 Характеристика исходной сети

Адрес в полной форме:

- 2001:0db8:c0de:0000:0000:0000:0000

Параметры сети:

- Префикс: /48
- Маска (в двоичном виде): 48 единиц и 80 нулей
- Часть сети: первые три блока (hexet) — 2001:0db8:c0de
- Диапазон адресов внутри префикса:
 - минимальный: 2001:db8:c0de:0:0:0:0
 - максимальный: 2001:db8:c0de:ffff:ffff:ffff:ffff

В IPv6 нет broadcast-адреса, вместо него используется многоадресная рассылка.

Однако «краевыми» адресами диапазона считаются показанные выше минимальный и максимальный адрес.

3.1.2 Разбиение на 2 подсети с использованием идентификатора подсети

Считаем, что для идентификатора подсети используются биты с 49-го по 64-й (четвёртый hextet).

Добавляем один бит к префиксу: /49. Это даёт две подсети:

1. Подсеть 1:

- Адрес сети: 2001:db8:c0de:0000::/49
- Диапазон адресов:
 - от 2001:db8:c0de:0:0:0:0
 - до 2001:db8:c0de:7fff:ffff:ffff:ffff:ffff

2. Подсеть 2:

- Адрес сети: 2001:db8:c0de:8000::/49
- Диапазон адресов:
 - от 2001:db8:c0de:8000:0:0:0
 - до 2001:db8:c0de:ffff:ffff:ffff:ffff:ffff

Интерпретация: меняется старший бит четвёртого блока (0 → первая подсеть, 8 → вторая подсеть), то есть мы используем именно поле идентификатора подсети.

3.1.3 Разбиение на 2 подсети с использованием идентификатора интерфейса

Теперь внутри одной из /64-подсетей сделаем разбиение по битам идентификатора интерфейса.

Сначала выберем одну /64 внутри исходного /48, например:

- 2001:db8:c0de:1::/64

В этой сети биты 65–128 образуют идентификатор интерфейса.

Добавим один бит к префиксу: получим две /65-подсети, отличающиеся старшим битом интерфейсного идентификатора:

1. Подсеть А (старший бит интерфейсного ID = 0):

- Адрес сети: 2001:db8:c0de:1:0000:0000:0000:0000/65
- Краевые адреса:
 - минимальный: 2001:db8:c0de:1:0:0:0:0
 - максимальный: 2001:db8:c0de:1:7fff:ffff:ffff:ffff

2. Подсеть В (старший бит интерфейсного ID = 1):

- Адрес сети: 2001:db8:c0de:1:8000:0000:0000:0000/65
- Краевые адреса:
 - минимальный: 2001:db8:c0de:1:8000:0:0:0
 - максимальный: 2001:db8:c0de:1:ffff:ffff:ffff:ffff

Интерпретация: префикс /64 остаётся одним и тем же с точки зрения глобального маршрутизирования (одно значение идентификатора подсети ...:1:), но мы «одолживаем» один бит из зоны интерфейсного ID и превращаем его в дополнительный бит префикса. Так получаются две меньшие подсети /65 внутри одной «классической» /64-подсети.

3.2 Сеть 2a02:6b8::/64

3.2.1 Характеристика исходной сети

Адрес в полной форме:

- 2a02:06b8:0000:0000:0000:0000:0000

Параметры сети:

- Префикс: /64
- Маска: 64 единицы и 64 нуля
- Часть сети: первые четыре блока – 2a02:06b8:0000:0000
- Диапазон адресов внутри префикса:
 - минимальный: 2a02:6b8:0:0:0:0:0
 - максимальный: 2a02:6b8:0:0:ffff:ffff:ffff

Это «тиpичный» размер IPv6-подсети для локальной сети: /64.

3.2.2 Разбиение на 2 подсети с использованием идентификатора подсети

Формально, при классическом планировании IPv6 биты 65–128 считаются идентификатором интерфейса.

Но мы можем «одолжить» один из этих битов и интерпретировать его как дополнительный бит идентификатора подсети, т.е. увеличить длину префикса до /65.

Получим две подсети /65:

1. Подсеть 1:

- Адрес сети: 2a02:6b8:0:0:0000:0000:0000/65 (часто записывают просто 2a02:6b8::/65)
- Диапазон адресов:
 - от 2a02:6b8:0:0:0:0:0
 - до 2a02:6b8:0:0:ffff:ffff:ffff:ffff

2. Подсеть 2:

- Адрес сети: 2a02:6b8:0:0:8000:0000:0000:0000/65
- Диапазон адресов:
 - от 2a02:6b8:0:0:8000:0:0:0
 - до 2a02:6b8:0:0:ffff:ffff:ffff:ffff

Интерпретация: мы «разрезали» /64 на две части, использовав старший бит интерфейсного ID как дополнительный бит сети — то есть превратили его в часть идентификатора подсети.

3.2.3 Разбиение на 2 подсети с использованием идентификатора интерфейса

Во втором варианте сам префикс /64 не изменяется: с точки зрения маршрутизаторов это по-прежнему одна сеть 2a02:6b8::/64.

Однако мы логически делим пространство интерфейсных идентификаторов, например, по старшему байту или старшему слову.

Пример:

- Подсеть A (условно «подсеть» по интерфейсному ID):

- используем интерфейсные идентификаторы в диапазоне
0000:0000:0000:0000 – 7fff:ffff:ffff:ffff
- Подсеть B:
 - используем интерфейсные идентификаторы в диапазоне
8000:0000:0000:0000 – ffff:ffff:ffff:ffff

Если закрепить за каждой такой группой, например, отдельный VLAN или отдельный пул адресов на DHCPv6-сервере, то на уровне хостов получится два логических сегмента, хотя маршрутизатор видит один и тот же префикс /64.

4 Выполнение

4.1 Настройка двойного стека адресации IPv4 и IPv6 в локальной сети

В рабочем пространстве GNS3 была развернута сеть согласно заданной топологии.

После размещения оборудования каждому устройству были назначены имена по требуемому формату.

Итоговая схема проекта представлена ниже:

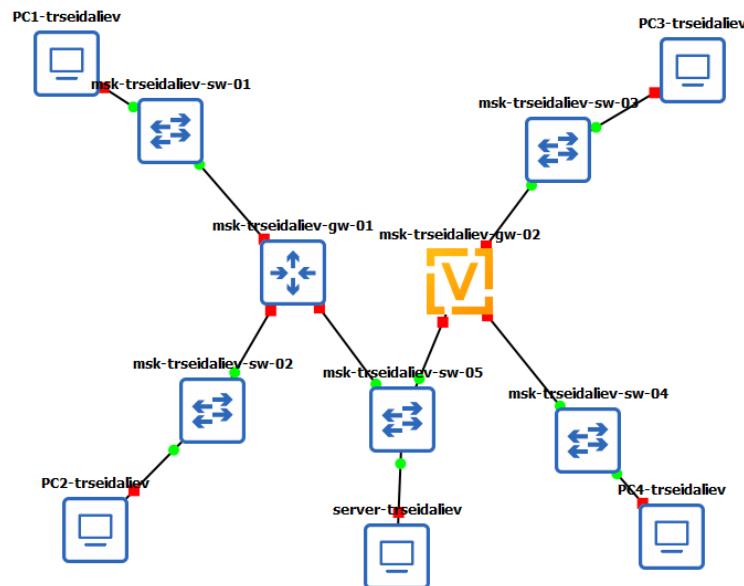
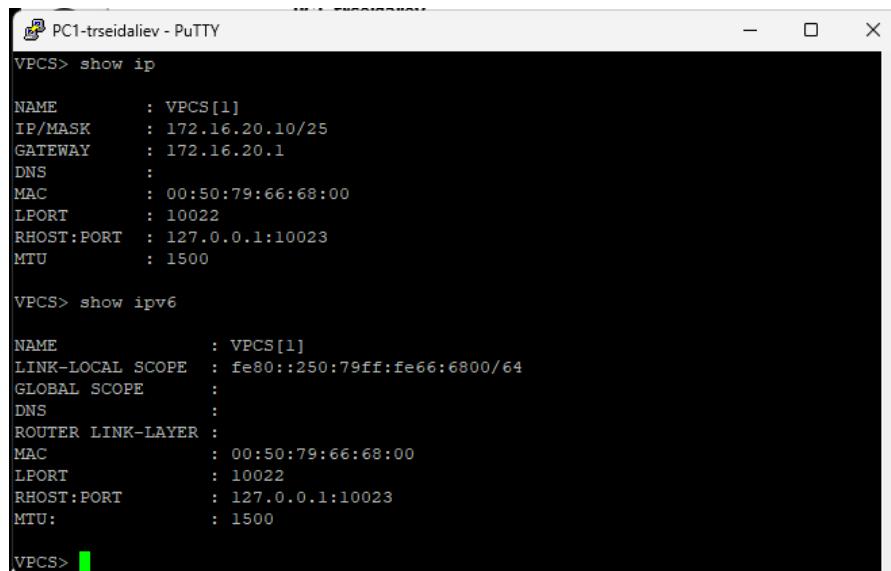


Рис. 4.1: Топология лабораторного стенда

На ПК1 была выполнена настройка адреса из подсети 172.16.20.0/25.

Команда `show ip` подтверждает корректное назначение параметров, включая шлюз 172.16.20.1.

Также выведена конфигурация IPv6.



```
VPCS> show ip
NAME      : VPCS[1]
IP/MASK   : 172.16.20.10/25
GATEWAY   : 172.16.20.1
DNS       :
MAC       : 00:50:79:66:68:00
LPORT     : 10022
RHOST:PORT : 127.0.0.1:10023
MTU       : 1500

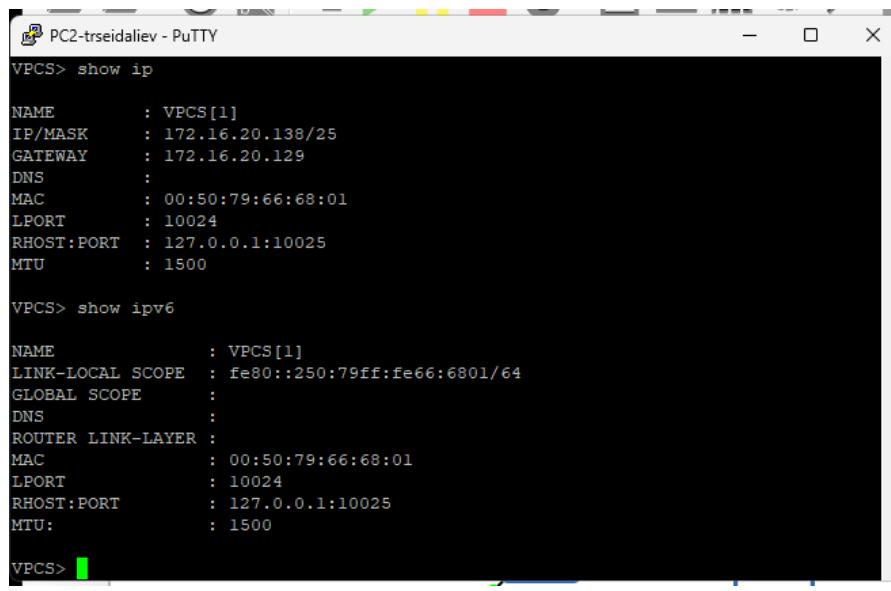
VPCS> show ipv6
NAME      : VPCS[1]
LINK-LOCAL SCOPE : fe80::250:79ff:fe66:6800/64
GLOBAL SCOPE   :
DNS       :
ROUTER LINK-LAYER :
MAC       : 00:50:79:66:68:00
LPORT     : 10022
RHOST:PORT   : 127.0.0.1:10023
MTU:      : 1500

VPCS>
```

Рис. 4.2: Конфигурация PC1

ПК2 настроен в подсети 172.16.20.128/25 с адресом 172.16.20.138.

Вывод IPv4 и IPv6 подтверждает корректность параметров.



```
VPCS> show ip
NAME      : VPCS[1]
IP/MASK   : 172.16.20.138/25
GATEWAY   : 172.16.20.129
DNS       :
MAC       : 00:50:79:66:68:01
LPORT     : 10024
RHOST:PORT : 127.0.0.1:10025
MTU       : 1500

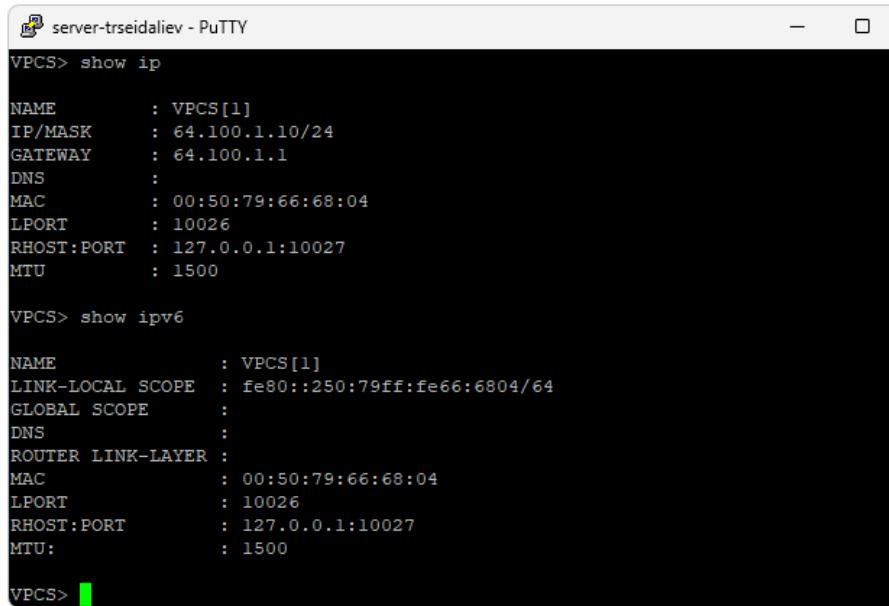
VPCS> show ipv6
NAME      : VPCS[1]
LINK-LOCAL SCOPE : fe80::250:79ff:fe66:6801/64
GLOBAL SCOPE   :
DNS       :
ROUTER LINK-LAYER :
MAC       : 00:50:79:66:68:01
LPORT     : 10024
RHOST:PORT   : 127.0.0.1:10025
MTU:      : 1500

VPCS>
```

Рис. 4.3: Конфигурация PC2

Сервер получил адрес в сети 64.100.1.0/24.

Вывод `show ip` демонстрирует корректную установку адреса и шлюза.



The screenshot shows a PuTTY terminal window titled "server-trseidaliev - PuTTY". The command `VPCS> show ip` is run, displaying the following configuration:

```
NAME      : VPCS[1]
IP/MASK   : 64.100.1.10/24
GATEWAY   : 64.100.1.1
DNS       :
MAC       : 00:50:79:66:68:04
LPORT     : 10026
RHOST:PORT: 127.0.0.1:10027
MTU       : 1500
```

Then, the command `VPCS> show ipv6` is run, displaying the following configuration:

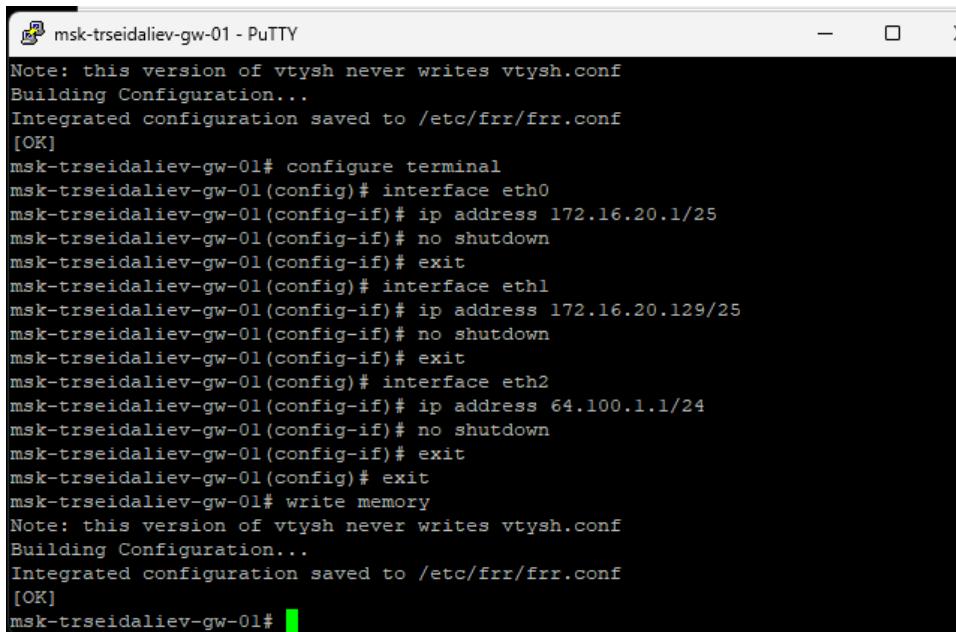
```
NAME      : VPCS[1]
LINK-LOCAL SCOPE : fe80::250:79ff:fe66:6804/64
GLOBAL SCOPE   :
DNS       :
ROUTER LINK-LAYER :
MAC       : 00:50:79:66:68:04
LPORT     : 10026
RHOST:PORT   : 127.0.0.1:10027
MTU       : 1500
```

Рис. 4.4: Конфигурация сервера

На маршрутизаторе FRR выполнено назначение имени хоста и настройка трёх интерфейсов:

- eth0: 172.16.20.1/25
- eth1: 172.16.20.129/25
- eth2: 64.100.1.1/24

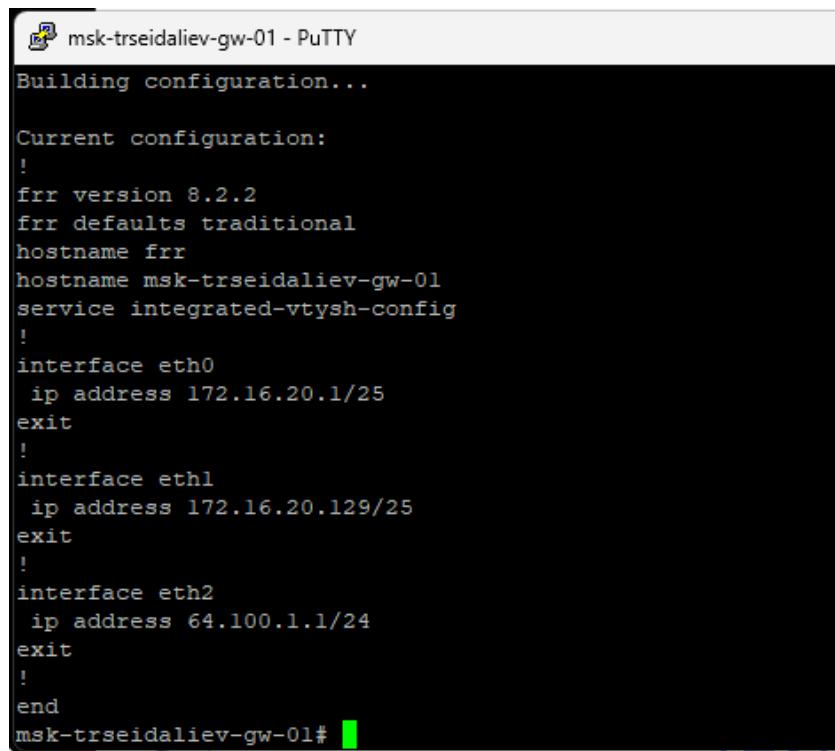
Все интерфейсы были активированы.



```
msk-trseidaliev-gw-01 - PuTTY
Note: this version of vtysh never writes vtysh.conf
Building Configuration...
Integrated configuration saved to /etc/frr/frr.conf
[OK]
msk-trseidaliev-gw-01# configure terminal
msk-trseidaliev-gw-01(config)# interface eth0
msk-trseidaliev-gw-01(config-if)# ip address 172.16.20.1/25
msk-trseidaliev-gw-01(config-if)# no shutdown
msk-trseidaliev-gw-01(config-if)# exit
msk-trseidaliev-gw-01(config)# interface eth1
msk-trseidaliev-gw-01(config-if)# ip address 172.16.20.129/25
msk-trseidaliev-gw-01(config-if)# no shutdown
msk-trseidaliev-gw-01(config-if)# exit
msk-trseidaliev-gw-01(config)# interface eth2
msk-trseidaliev-gw-01(config-if)# ip address 64.100.1.1/24
msk-trseidaliev-gw-01(config-if)# no shutdown
msk-trseidaliev-gw-01(config-if)# exit
msk-trseidaliev-gw-01(config)# exit
msk-trseidaliev-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-trseidaliev-gw-01#
```

Рис. 4.5: Процесс настройки интерфейсов FRR

Конфигурация маршрутизатора отображается через `show running-config`.



```
msk-trseidaliev-gw-01 - PuTTY
Building configuration...

Current configuration:
!
frr version 8.2.2
frr defaults traditional
hostname frr
hostname msk-trseidaliev-gw-01
service integrated-vtysh-config
!
interface eth0
  ip address 172.16.20.1/25
exit
!
interface eth1
  ip address 172.16.20.129/25
exit
!
interface eth2
  ip address 64.100.1.1/24
exit
!
end
msk-trseidaliev-gw-01#
```

Рис. 4.6: Вывод `running-config`

Команда `show interface brief` показывает состояние и назначенные адреса.

```
msk-trseidaliev-gw-01# show interface brief
Interface      Status   VRF      Addresses
-----      -----   ---      -----
eth0          up       default   172.16.20.1/25
eth1          up       default   172.16.20.129/25
eth2          up       default   64.100.1.1/24
eth3          down     default
eth4          down     default
eth5          down     default
eth6          down     default
eth7          down     default
lo            up       default
pimreg        up       default

msk-trseidaliev-gw-01#
```

Рис. 4.7: Информация об интерфейсах FRR

PC1 успешно отправляет:

- ICMP-эхо-запросы на PC2 (172.16.20.138),
- ICMP-эхо-запросы на сервер (64.100.1.10).

Трассировка маршрута показывает прохождение трафика через FRR.

```
PC1-trseidaliev - PuTTY

VPCS> ping 172.16.20.138
84 bytes from 172.16.20.138 icmp_seq=1 ttl=63 time=3.258 ms
84 bytes from 172.16.20.138 icmp_seq=2 ttl=63 time=4.224 ms
84 bytes from 172.16.20.138 icmp_seq=3 ttl=63 time=1.336 ms
84 bytes from 172.16.20.138 icmp_seq=4 ttl=63 time=4.678 ms
84 bytes from 172.16.20.138 icmp_seq=5 ttl=63 time=4.397 ms

VPCS> ping 64.100.1.10
84 bytes from 64.100.1.10 icmp_seq=1 ttl=63 time=5.028 ms
84 bytes from 64.100.1.10 icmp_seq=2 ttl=63 time=4.257 ms
84 bytes from 64.100.1.10 icmp_seq=3 ttl=63 time=3.892 ms
84 bytes from 64.100.1.10 icmp_seq=4 ttl=63 time=3.494 ms
84 bytes from 64.100.1.10 icmp_seq=5 ttl=63 time=2.916 ms

VPCS> trace 172.16.20.138
trace to 172.16.20.138, 8 hops max, press Ctrl+C to stop
1  172.16.20.1  5.462 ms  0.897 ms  0.817 ms
2  *172.16.20.138  1.146 ms (ICMP type:3, code:3, Destination port unreachable)

VPCS> trace 64.100.1.10
trace to 64.100.1.10, 8 hops max, press Ctrl+C to stop
1  172.16.20.1  1.944 ms  2.141 ms  1.598 ms
2  *64.100.1.10  1.763 ms (ICMP type:3, code:3, Destination port unreachable)

VPCS>
```

Рис. 4.8: Ping и trace с PC1

На линии между сервером и коммутатором был включён захват трафика.

Фиксируются ARP-запросы, ICMP-эхо-запросы и ответы.

На фрагменте виден запрос «Who has 64.100.1.10? Tell 64.100.1.1».

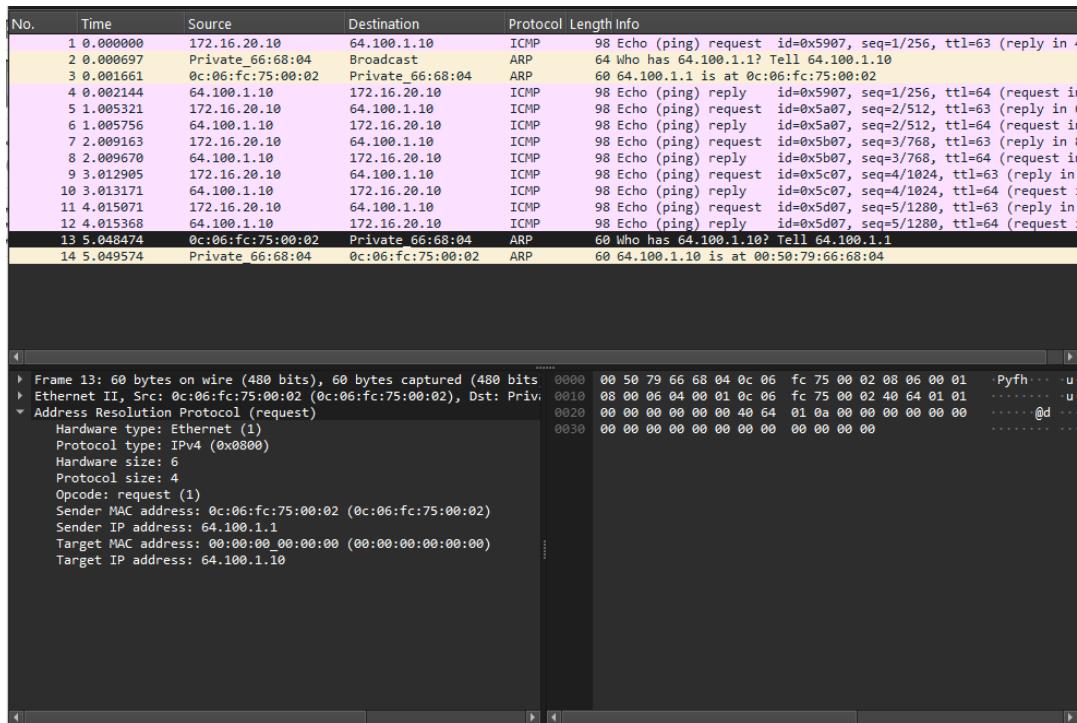


Рис. 4.9: ARP-трафик

Пример успешного ответа сервера на эхо-запрос.

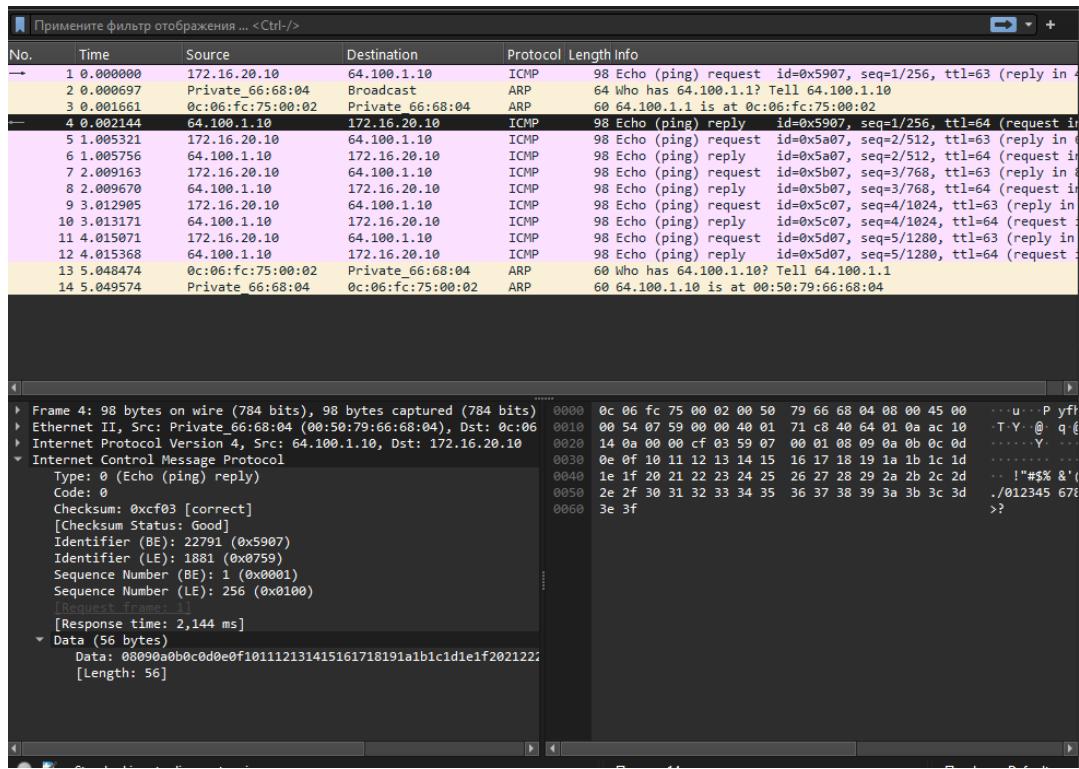


Рис. 4.10: ICMP-ответ сервера

В соответствии с таблицей 6.6 на узлы подсети IPv6 были назначены глобальные адреса.

Узлу PC3 назначен адрес из подсети 2001:db8:c0de:12::/64.

Команды `show ip` и `show ipv6` отображают отсутствие IPv4-конфигурации и наличие корректного глобального IPv6-адреса.

```
VPCS> show ip
NAME      : VPCS[1]
IP/MASK   : 0.0.0.0/0
GATEWAY   : 0.0.0.0
DNS       :
MAC       : 00:50:79:66:68:02
LPORT     : 10044
RHOST:PORT : 127.0.0.1:10045
MTU       : 1500

VPCS> show ipv6
NAME      : VPCS[1]
LINK-LOCAL SCOPE : fe80::250:79ff:fe66:6802/64
GLOBAL SCOPE    : 2001:db8:c0de:12::a/64
DNS       :
ROUTER LINK-LAYER :
MAC       : 00:50:79:66:68:02
LPORT     : 10044
RHOST:PORT    : 127.0.0.1:10045
MTU       : 1500

VPCS>
```

Рис. 4.11: Конфигурация PC3

ПК4 получил IPv6-адрес из подсети 2001:db8:c0de:13::/64.

Вывод показывает только IPv6-конфигурацию, аналогично PC3.

```
VPCS> show ip
NAME      : VPCS[1]
IP/MASK   : 0.0.0.0/0
GATEWAY   : 0.0.0.0
DNS       :
MAC       : 00:50:79:66:68:03
LPORT     : 10046
RHOST:PORT : 127.0.0.1:10047
MTU       : 1500

VPCS> show ipv6
NAME      : VPCS[1]
LINK-LOCAL SCOPE : fe80::250:79ff:fe66:6803/64
GLOBAL SCOPE    : 2001:db8:c0de:13::a/64
DNS       :
ROUTER LINK-LAYER :
MAC       : 00:50:79:66:68:03
LPORT     : 10046
RHOST:PORT    : 127.0.0.1:10047
MTU       : 1500

VPCS>
```

Рис. 4.12: Конфигурация PC4

Сервер имеет два стека адресации:

IPv4 – ранее настроена в подсети 64.100.1.0/24,

IPv6 – адрес из подсети 2001:db8:c0de:11::/64.

```
VPCS> show ip
NAME      : VPCS[1]
IP/MASK   : 64.100.1.10/24
GATEWAY   : 64.100.1.1
DNS       :
MAC       : 00:50:79:66:68:04
LPORT     : 10026
RHOST:PORT: 127.0.0.1:10027
MTU       : 1500

VPCS> show ipv6
NAME      : VPCS[1]
LINK-LOCAL SCOPE : fe80::250:79ff:fe66:6804/64
GLOBAL SCOPE    : 2001:db8:c0de:11::a/64
DNS       :
ROUTER LINK-LAYER :
MAC       : 00:50:79:66:68:04
LPORT     : 10026
RHOST:PORT  : 127.0.0.1:10027
MTU       : 1500

VPCS>
```

Рис. 4.13: IPv6-конфигурация сервера

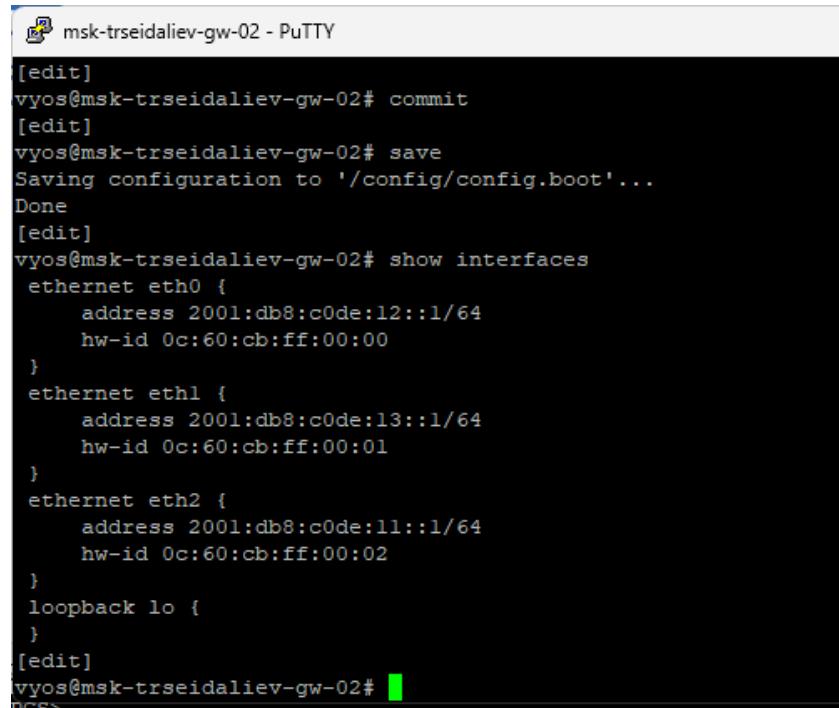
После установки VyOS и изменения имени хоста выполнялась настройка трёх интерфейсов маршрутизатора.

Каждый интерфейс получил адрес и был включён в RA-сервис для автоматической раздачи префиксов.

```
VyOS is a free software distribution that includes multiple components,
you can check individual component licenses under /usr/share/doc/*copyright
vyos@msk-trseidaliev-gw-02:~$ configure
[edit]
vyos@msk-trseidaliev-gw-02# set interfaces ethernet eth0 address 2001:db8:c0de:12
::1/64
[edit]
vyos@msk-trseidaliev-gw-02# set service router-advert interface eth0 prefix 2001
:db8:c0de:12::/64
[edit]
vyos@msk-trseidaliev-gw-02# set interfaces ethernet eth1 address 2001:db8:c0de:13
::1/64
[edit]
vyos@msk-trseidaliev-gw-02# set service router-advert interface eth1 prefix 2001
:db8:c0de:13::/64
[edit]
vyos@msk-trseidaliev-gw-02# set interfaces ethernet eth2 address 2001:db8:c0de:11
::1/64
[edit]
vyos@msk-trseidaliev-gw-02# set service router-advert interface eth2 prefix 2001
:db8:c0de:11::/64
[edit]
vyos@msk-trseidaliev-gw-02#
```

Рис. 4.14: Назначение IPv6-адресов на VyOS

После применения конфигурации был выполнен просмотр интерфейсов.



```
[edit]
vyos@msk-trseidaliev-gw-02# commit
[edit]
vyos@msk-trseidaliev-gw-02# save
Saving configuration to '/config/config.boot'...
Done
[edit]
vyos@msk-trseidaliev-gw-02# show interfaces
  ethernet eth0 {
    address 2001:db8:c0de:12::1/64
    hw-id 0c:60:cb:ff:00:00
  }
  ethernet eth1 {
    address 2001:db8:c0de:13::1/64
    hw-id 0c:60:cb:ff:00:01
  }
  ethernet eth2 {
    address 2001:db8:c0de:11::1/64
    hw-id 0c:60:cb:ff:00:02
  }
  loopback lo {
  }
[edit]
vyos@msk-trseidaliev-gw-02#
```

Рис. 4.15: Вывод show interfaces на VyOS

PC4 успешно отправляет ICMPv6-эхо-запросы к узлам:

- PC3: 2001:db8:c0de:12::a
- Server: 2001:db8:c0de:11::a

При этом доступ к IPv4-адресам недоступен — что соответствует требованиям разделения подсетей.

```
2001:db8:c0de:12::a icmp6_seq=2 ttl=62 time=2.153 ms
2001:db8:c0de:12::a icmp6_seq=3 ttl=62 time=2.182 ms
2001:db8:c0de:12::a icmp6_seq=4 ttl=62 time=2.143 ms
2001:db8:c0de:12::a icmp6_seq=5 ttl=62 time=1.484 ms

VPCS> ping 2001:db8:c0de:11::a/64

2001:db8:c0de:11::a icmp6_seq=1 ttl=62 time=3.799 ms
2001:db8:c0de:11::a icmp6_seq=2 ttl=62 time=2.657 ms
2001:db8:c0de:11::a icmp6_seq=3 ttl=62 time=3.253 ms
2001:db8:c0de:11::a icmp6_seq=4 ttl=62 time=2.837 ms
2001:db8:c0de:11::a icmp6_seq=5 ttl=62 time=1.424 ms

VPCS> trace 2001:db8:c0de:12::a/64

trace to 2001:db8:c0de:12::a, 64 hops max
 1 2001:db8:c0de:13::1    2.569 ms   1.390 ms   1.719 ms
 2 2001:db8:c0de:12::a    1.837 ms   1.448 ms   1.615 ms

VPCS> trace 2001:db8:c0de:11::a/64

trace to 2001:db8:c0de:11::a, 64 hops max
 1 2001:db8:c0de:13::1    1.128 ms   1.328 ms   1.041 ms
 2 2001:db8:c0de:11::a    1.968 ms   2.501 ms   2.157 ms

VPCS>
VPCS> ping 64.100.1.10

host (64.100.1.10) not reachable

VPCS> ping 172.16.20.138

host (172.16.20.138) not reachable

VPCS>
```

Рис. 4.16: Ping и trace с PC4

Сервер успешно пингует узлы IPv6-сегмента и узлы IPv4-подсети.

```
VPCS> ping 2001:db8:c0de:12::a

2001:db8:c0de:12::a icmp6_seq=1 ttl=62 time=4.216 ms
2001:db8:c0de:12::a icmp6_seq=2 ttl=62 time=1.529 ms
2001:db8:c0de:12::a icmp6_seq=3 ttl=62 time=2.119 ms
2001:db8:c0de:12::a icmp6_seq=4 ttl=62 time=3.367 ms
2001:db8:c0de:12::a icmp6_seq=5 ttl=62 time=1.972 ms

VPCS> ping 172.16.20.10

84 bytes from 172.16.20.10 icmp_seq=1 ttl=63 time=4.709 ms
84 bytes from 172.16.20.10 icmp_seq=2 ttl=63 time=1.584 ms
84 bytes from 172.16.20.10 icmp_seq=3 ttl=63 time=2.463 ms
84 bytes from 172.16.20.10 icmp_seq=4 ttl=63 time=1.759 ms
84 bytes from 172.16.20.10 icmp_seq=5 ttl=63 time=2.506 ms

VPCS>
```

Рис. 4.17: Ping с сервера IPv6 и IPv4

На захваченном трафике видно:

- ICMPv6 Echo Request,
- ICMPv6 Echo Reply,
- Neighbor Solicitation/Advertisement.

Это подтверждает корректное функционирование IPv6 и SLAAC.

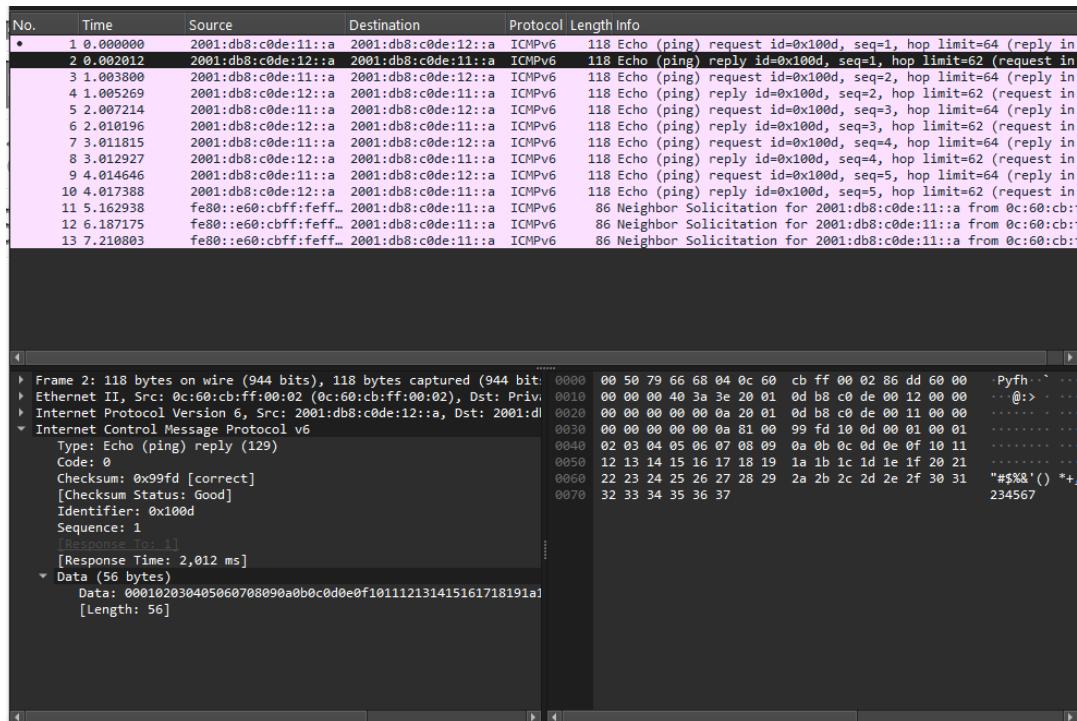


Рис. 4.18: Фрагмент ICMPv6-трафика в Wireshark

4.2 Задание для самостоятельного выполнения

Для топологии заданы две локальные подсети — каждая содержит как IPv4-, так и IPv6-адресное пространство.

Подсеть 1 - IPv4: **10.10.1.96/27**

Размер подсети — 32 адреса.

Диапазон:

- сеть: 10.10.1.96

- usable: 10.10.1.97 – 10.10.1.126
- broadcast: 10.10.1.127

- IPv6: **2001:db8:1:1::/64**

Диапазон адресов: 2001:db8:1:1:: – 2001:db8:1:1:ffff:ffff:ffff:ffff

Подсеть 2 - IPv4: **10.10.1.16/28**

Размер подсети – 16 адресов.

Диапазон:

- сеть: 10.10.1.16
- usable: 10.10.1.17 – 10.10.1.30
- broadcast: 10.10.1.31

- IPv6: **2001:db8:1:4::/64**

Диапазон адресов: 2001:db8:1:4:: – 2001:db8:1:4:ffff:ffff:ffff:ffff

Для интерфейсов маршрутизатора используются **минимальные доступные адреса**.

Устройство	Интерфейс	IPv4-адрес	Маска	IPv6-адрес	Префикс
msk-trseidaliev-gw-01	eth0	10.10.1.97	/27	2001:db8:1:1::1	/64
msk-trseidaliev-gw-01	eth1	10.10.1.17	/28	2001:db8:1:4::1	/64
PC1	vpcs	10.10.1.100	/27	2001:db8:1:1::a	/64
PC2	vpcs	10.10.1.20	/28	2001:db8:1:4::a	/64

PC1 Настроенные IPv4 и IPv6 отображаются в выводе:

```
VPCS> show ip

NAME      : VPCS[1]
IP/MASK   : 10.10.1.100/27
GATEWAY   : 10.10.1.97
DNS       :
MAC       : 00:50:79:66:68:00
LPORT     : 10008
RHOST:PORT : 127.0.0.1:10009
MTU       : 1500

VPCS> show ipv6

NAME      : VPCS[1]
LINK-LOCAL SCOPE : fe80::250:79ff:fe66:6800/64
GLOBAL SCOPE    : 2001:db8:1:1::a/64
DNS       :
ROUTER LINK-LAYER :
MAC       : 00:50:79:66:68:00
LPORT     : 10008
RHOST:PORT    : 127.0.0.1:10009
MTU:        : 1500

VPCS>
```

Рис. 4.19: PC1 вывод show ip

Для PC2 настроены адреса второй подсети:

```
VPCS> show ip

NAME      : VPCS[1]
IP/MASK   : 10.10.1.20/28
GATEWAY   : 10.10.1.17
DNS       :
MAC       : 00:50:79:66:68:01
LPORT     : 10016
RHOST:PORT : 127.0.0.1:10017
MTU       : 1500

VPCS> show ipv6

NAME      : VPCS[1]
LINK-LOCAL SCOPE : fe80::250:79ff:fe66:6801/64
GLOBAL SCOPE    : 2001:db8:1:4::a/64
DNS       :
ROUTER LINK-LAYER : 0c:76:30:16:00:01
MAC       : 00:50:79:66:68:01
LPORT     : 10016
RHOST:PORT    : 127.0.0.1:10017
MTU:        : 1500

VPCS>
```

Рис. 4.20: PC2 вывод show ip

Используемая схема сети:

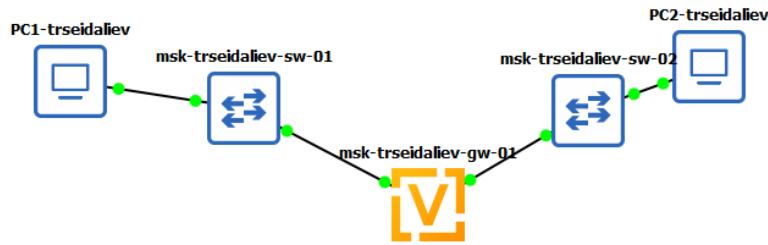


Рис. 4.21: Топология сети

Маршрутизатор установлен и переведён в режим конфигурирования.

Для каждого интерфейса заданы IPv4 и IPv6-адреса, соответствующие минимальным адресам подсети.

```
PC1-trseidaliev - PuTTY
VPCS> show ip

NAME      : VPCS[1]
IP/MASK   : 10.10.1.100/27
GATEWAY   : 10.10.1.97
DNS       :
MAC       : 00:50:79:66:68:00
LPORT     : 10008
RHOST:PORT : 127.0.0.1:10009
MTU       : 1500

VPCS> show ipv6

NAME      : VPCS[1]
LINK-LOCAL SCOPE : fe80::250:79ff:fe66:6800/64
GLOBAL SCOPE   : 2001:db8:1:1::a/64
DNS       :
ROUTER LINK-LAYER : 0c:76:30:16:00:00
MAC       : 00:50:79:66:68:00
LPORT     : 10008
RHOST:PORT : 127.0.0.1:10009
MTU       : 1500

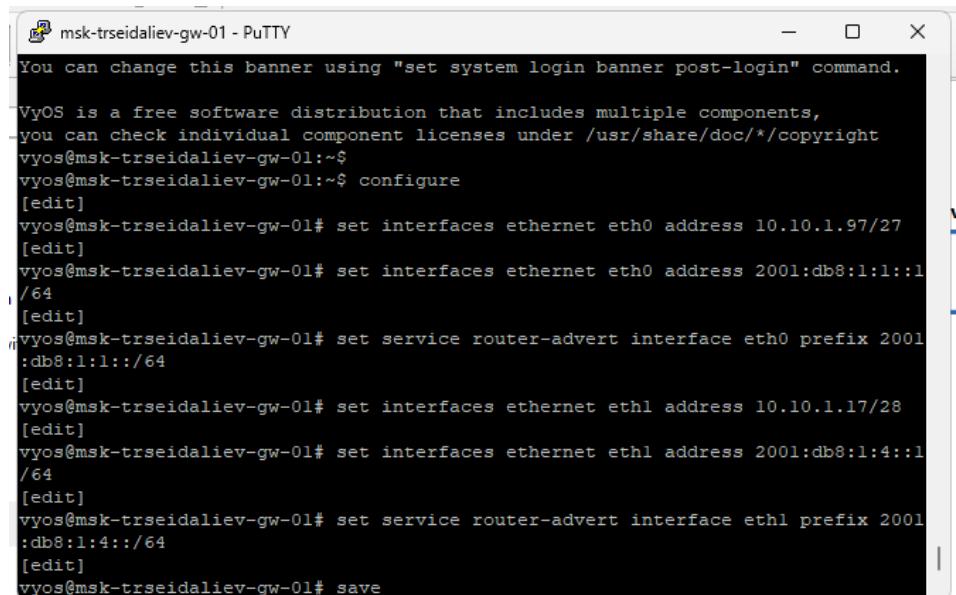
VPCS>
```

Рис. 4.22: Настройка VyOS интерфейсов

PC1 успешно пингует:

- PC2 по IPv4: 10.10.1.20
- PC2 по IPv6: 2001:db8:1:4::a

Также выполнена трассировка маршрута.



```
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-trseidaliev-gw-01:~$ configure
[edit]
vyos@msk-trseidaliev-gw-01# set interfaces ethernet eth0 address 10.10.1.97/27
[edit]
vyos@msk-trseidaliev-gw-01# set interfaces ethernet eth0 address 2001:db8:1:1::1
/64
[edit]
vyos@msk-trseidaliev-gw-01# set service router-advert interface eth0 prefix 2001
:db8:1:1::/64
[edit]
vyos@msk-trseidaliev-gw-01# set interfaces ethernet eth1 address 10.10.1.17/28
[edit]
vyos@msk-trseidaliev-gw-01# set interfaces ethernet eth1 address 2001:db8:1:4::1
/64
[edit]
vyos@msk-trseidaliev-gw-01# set service router-advert interface eth1 prefix 2001
:db8:1:4::/64
[edit]
vyos@msk-trseidaliev-gw-01# save
```

Рис. 4.23: Ping и trace с PC1

Проверяются маршруты и ICMPv6-ответы.

```
VPCS> [ing 10.10.1.20
Bad command: "[ing 10.10.1.20". Use ? for help.

VPCS> ping 10.10.1.20

84 bytes from 10.10.1.20 icmp_seq=1 ttl=63 time=5.612 ms
84 bytes from 10.10.1.20 icmp_seq=2 ttl=63 time=3.600 ms
84 bytes from 10.10.1.20 icmp_seq=3 ttl=63 time=3.566 ms
84 bytes from 10.10.1.20 icmp_seq=4 ttl=63 time=2.339 ms
84 bytes from 10.10.1.20 icmp_seq=5 ttl=63 time=3.560 ms

VPCS> ping 2001:db8:1:4::a

2001:db8:1:4::a icmp6_seq=1 ttl=62 time=9.651 ms
2001:db8:1:4::a icmp6_seq=2 ttl=62 time=5.577 ms
2001:db8:1:4::a icmp6_seq=3 ttl=62 time=6.361 ms
2001:db8:1:4::a icmp6_seq=4 ttl=62 time=1.455 ms
2001:db8:1:4::a icmp6_seq=5 ttl=62 time=1.925 ms

VPCS>
```

Рис. 4.24: IPv6 ICMP проверка

Связность между подсетями обеспечивается только через маршрутизатор.

```
VPCS> trace 10.10.1.20
trace to 10.10.1.20, 8 hops max, press Ctrl+C to stop
 1  10.10.1.97    1.117 ms  0.895 ms  0.509 ms
 2  *10.10.1.20   1.552 ms (ICMP type:3, code:3, Destination port unreachable)

VPCS> trace 2001:db8:1:4::a

trace to 2001:db8:1:4::a, 64 hops max
 1 2001:db8:1:1::1   3.251 ms  3.012 ms  2.810 ms
 2 2001:db8:1:4::a   3.986 ms  2.684 ms  1.610 ms

VPCS>
```

Рис. 4.25: Проверка IPv4 с PC1

5 Заключение

В ходе выполнения работы:

- выполнено детальное разбиение IPv4- и IPv6-адресных пространств на подсети с расчётом масок, префиксов, диапазонов адресов и краевых значений;
- разработаны таблицы адресации для всех вариантов разбиения, включая выбор минимальных адресов для интерфейсов маршрутизаторов;
- настроены IPv4- и IPv6-адреса на конечных узлах и маршрутизаторах VyOS в соответствии с заданными топологиями;
- проверена работоспособность сетей с помощью утилит `ping` и `trace`, подтверждена корректность маршрутизации и изоляции подсетей;
- выполнен анализ трафика ICMPv4 и ICMPv6 в Wireshark, подтверждающий правильность функционирования адресного стека и соседских взаимодействий;
- обеспечена корректная работа двойного стека IPv4/IPv6, включая взаимодействие сервера с обеими подсетями и изоляцию сетей разного протокола.

Работа позволила закрепить навыки разбиения адресных пространств, настройки маршрутизаторов и анализа сетевого трафика в среде виртуального моделирования.