

Отчёт по лабораторной работе 6

Адресация IPv4 и IPv6. Двойной стек

Гафоров Нурмухаммад

Содержание

1 Введение	5
1.1 Цель работы	5
2 Ход выполнения	6
2.1 Разбиение IPv4-сетей на подсети	6
2.1.1 Сеть 172.16.20.0/24	6
2.1.2 Сеть 10.10.1.64/26	7
2.1.3 Сеть 10.10.1.0/26	7
2.2 Разбиение IPv6-сети на подсети	7
2.2.1 Сеть 2001:db8:c0de::/48	7
2.2.2 Сеть 2a02:6b8::/64	8
2.3 Настройка двойного стека адресации IPv4 и IPv6 в локальной сети	9
2.3.1 PC1	10
2.3.2 PC2	10
2.3.3 Настройка IPv4 на сервере с двойным стеком	11
2.3.4 Настройка маршрутизатора FRR (msk-ngaforov-gw-01)	12
2.3.5 Проверка связности между узлами	15
2.3.6 Настройка IPv6 на PC3, PC4 и сервере Dual Stack	17
2.3.7 Настройка IPv6 на маршрутизаторе VyOS (msk-ngaforov-gw-02)	19
2.3.8 Проверка связности по IPv6	21
2.3.9 Анализ захваченного трафика ARP, ICMP и ICMPv6	23
2.4 Самостоятельное задание: настройка двух подсетей IPv4/IPv6 и проверка связности	27
2.4.1 Подсеть 1 (IPv4)	27
2.4.2 Подсеть 2 (IPv4)	27
2.4.3 Подсети IPv6	27
2.4.4 Таблица адресации	28
2.4.5 Настройка устройств	29
2.4.6 Проверка связности (ping/trace)	31
3 Вывод	32

Список иллюстраций

2.1	Топология сети GNS3	9
2.2	Конфигурация PC1	10
2.3	Конфигурация PC2	11
2.4	Сервер Dual Stack	12
2.5	Назначение адресов на FRR	13
2.6	show running-config	14
2.7	show interface brief	15
2.8	ping+trace с PC1	16
2.9	ping+trace с PC2	16
2.10	Конфигурация PC3 IPv6	17
2.11	Конфигурация PC4 IPv6	18
2.12	Конфигурация сервера IPv6	19
2.13	Настройка адресов на VyOS	20
2.14	Состояние интерфейсов VyOS	20
2.15	PC3: проверка IPv6	21
2.16	PC4: проверка IPv6	22
2.17	Dual Stack Server: проверки IPv6	23
2.18	ARP-запрос	24
2.19	ICMP Echo (IPv4)	25
2.20	ICMPv6 Echo (IPv6)	26
2.21	Топология	27
2.22	PC1: show ip/ipv6	29
2.23	PC2: show ip/ipv6	29
2.24	VyOS	30
2.25	VyOS: show interfaces	30
2.26	PC1: ping IPv4/IPv6	31

Список таблиц

1 Введение

1.1 Цель работы

Изучение принципов распределения и настройки адресного пространства на устройствах сети.

2 Ход выполнения

2.1 Разбиение IPv4-сетей на подсети

2.1.1 Сеть 172.16.20.0/24

База сети - Префикс: /24 - Маска: 255.255.255.0 - Кол-во адресов в сети: 256 - Кол-во адресов хостов: 254 - Broadcast: 172.16.20.255 - Диапазон хостов: 172.16.20.1 – 172.16.20.254

Замечание про “число возможных подсетей”

Внутри /24 можно дробить на подсети произвольного размера (VLSM). Например:
- /25 → 2 подсети, - /26 → 4 подсети, - /27 → 8 подсетей, и т. д. (каждый «заём» одного бита удваивает число подсетей).

Требуемое разбиение: 126, 62, 62 адресов хостов (VLSM)

- Подсеть А: /25 (128 адресов, 126 хостов)
- Сеть: 172.16.20.0/25
- Хосты: 172.16.20.1 – 172.16.20.126
- Broadcast: 172.16.20.127 - Подсеть В: /26 (64 адреса, 62 хоста)
- Сеть: 172.16.20.128/26
- Хосты: 172.16.20.129 – 172.16.20.190
- Broadcast: 172.16.20.191 - Подсеть С: /26 (64 адреса, 62 хоста)
- Сеть: 172.16.20.192/26
- Хосты: 172.16.20.193 – 172.16.20.254
- Broadcast: 172.16.20.255

2.1.2 Сеть 10.10.1.64/26

База сети - Префикс: /26 - Маска: **255.255.255.192** - Кол-во адресов в сети: **64** - Кол-во адресов хостов: **62** - Сеть: **10.10.1.64** - Broadcast: **10.10.1.127** - Диапазон хостов: **10.10.1.65 – 10.10.1.126**

Подсеть на 30 узлов (нужно ≥ 30 хостов \rightarrow /27, т.е. 32 адреса, 30 хостов): - Выделяем **10.10.1.64/27** - Хосты: **10.10.1.65 – 10.10.1.94** - Broadcast: **10.10.1.95**

2.1.3 Сеть 10.10.1.0/26

База сети - Префикс: /26 - Маска: **255.255.255.192** - Кол-во адресов в сети: **64** - Кол-во адресов хостов: **62** - Сеть: **10.10.1.0** - Broadcast: **10.10.1.63** - Диапазон хостов: **10.10.1.1 – 10.10.1.62**

Подсеть на 14 узлов (нужно ≥ 14 хостов \rightarrow /28, т.е. 16 адресов, 14 хостов): - Выделяем **10.10.1.0/28** - Хосты: **10.10.1.1 – 10.10.1.14** - Broadcast: **10.10.1.15**

2.2 Разбиение IPv6-сети на подсети

2.2.1 Сеть 2001:db8:c0de::/48

Характеристика адреса - Префикс: /48

- Маска (hex): **ffff:ffff:ffff::**

- Фиксированная часть: **2001:db8:c0de**

- Диапазон адресов всей /48:

2001:db8:c0de:0000:0000:0000:0000:0000 – 2001:db8:c0de:ffff:ffff:ffff:ffff:ffff

- Внутри /48 стандартно выделяют /64-подсети: 16 бит **Subnet ID** (четвёртый хекстет), 64 бита **Interface ID**.

В IPv6 нет broadcast; используются multicast и ICMPv6 (ND).

Вариант А — с использованием идентификатора подсети

(увеличиваем длину префикса, делим /48 на две равные крупные подсети /49)

- **Подсеть 1:** 2001:db8:c0de:0000::/49
Содержит Subnet ID от 0000 до 7fff (т.е. 2^{15} разных /64).
- **Подсеть 2:** 2001:db8:c0de:8000::/49
Содержит Subnet ID от 8000 до ffff (ещё 2^{15} /64).

Вариант В — с использованием идентификатора интерфейса

(делим одну /64 на две части, «занимая» 1 бит из **Interface ID** → /65)

Например, возьмём одну /64: 2001:db8:c0de:0000::/64 и разделим её:

- **Часть 1 (IID старший бит = 0):** 2001:db8:c0de:0000:0000:0000:0000/65
Диапазон IID: 0000: . . . :0000 – 7fff:ffff:ffff:ffff.
- **Часть 2 (IID старший бит = 1):** 2001:db8:c0de:0000:8000:0000:0000/65
Диапазон IID: 8000: . . . :0000 – ffff:ffff:ffff:ffff.

2.2.2 Сеть 2a02:6b8::/64

Характеристика адреса - Префикс: /64

- Маска (hex): **ffff:ffff:ffff:ffff::**
- Фиксированная часть: **2a02:06b8:0000:0000**
- Диапазон адресов всей /64:

2a02:6b8:0:0:0000:0000:0000:0000 – 2a02:6b8:0:0:ffff:ffff:ffff:ffff

Вариант А — с использованием идентификатора подсети (административно расширяем префикс)

Разделим исходную /64 на две /65:

- **Подсеть 1:** 2a02:6b8::/65
(IID старший бит = 0; адреса ...:0000:0000:0000:0000 – ...:7fff:ffff:ffff:ffff)
- **Подсеть 2:** 2a02:6b8:0:0:8000::/65
(IID старший бит = 1; адреса ...:8000:0000:0000:0000 – ...:ffff:ffff:ffff:ffff)

Вариант В — с использованием идентификатора интерфейса (логическое разделение IID без смены префикса)

Оставляем префикс /**64** неизменным, но резервируем два непересекающихся диапазона **IID**:

- **Группа 1 (IID 0...7fff:ffff:ffff:ffff)**: адреса 2a02:6b8::1 ... 2a02:6b8::7fff:ffff:ffff:ffff
- **Группа 2 (IID 8000:....ffff:ffff)**: адреса 2a02:6b8::8000:0:0:0 ... 2a02:6b8::ffff:ffff:ffff:ffff

2.3 Настройка двойного стека адресации IPv4 и IPv6 в локальной сети

В среде **GNS3** создан новый проект и размещены устройства согласно предоставленной топологии. В рабочем пространстве расположены два узла VPCS, коммутаторы, маршрутизатор **FRR** и сервер с двойным стеком адресации (IPv4/IPv6).

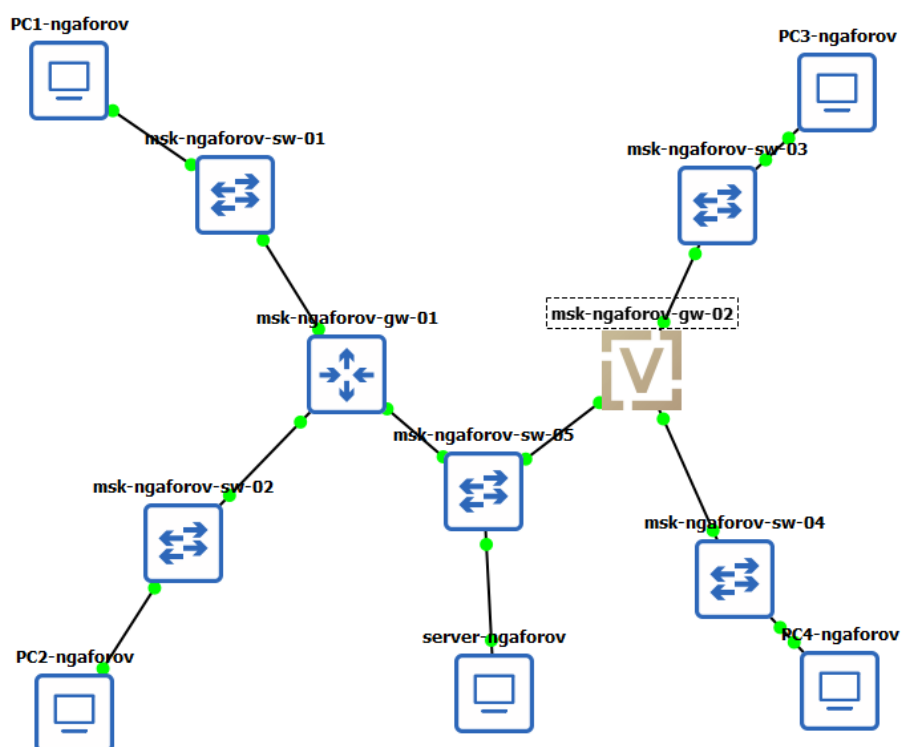


Рис. 2.1: Топология сети GNS3

2.3.1 PC1

Назначен IPv4-адрес и шлюз по умолчанию:

```
ip 172.16.20.10/25 172.16.20.1
```

save

После этого отображена конфигурация IPv4 и IPv6 (show ip, show ipv6).

```
PC1-ngaforov> show ip
NAME           : PC1-ngaforov[1]
IP/MASK        : 172.16.20.10/25
GATEWAY        : 172.16.20.1
DNS            :
MAC            : 00:50:79:66:68:00
LPORT          : 10026
RHOST:PORT     : 127.0.0.1:10027
MTU            : 1500

PC1-ngaforov> show ipv6
NAME           : PC1-ngaforov[1]
LINK-LOCAL SCOPE : fe80::250:79ff:fe66:6800/64
GLOBAL SCOPE    :
DNS             :
ROUTER LINK-LAYER :
MAC            : 00:50:79:66:68:00
LPORT          : 10026
RHOST:PORT     : 127.0.0.1:10027
MTU            : 1500

PC1-ngaforov>
```

Рис. 2.2: Конфигурация PC1

2.3.2 PC2

Назначение параметров IPv4 и шлюза по умолчанию:

```
ip 172.16.20.138/25 172.16.20.129
```

save

Выведены сведения о назначенных IPv4 и IPv6-адресах.

```

PC2-ngaforov> show ip
NAME           : PC2-ngaforov[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

PC2-ngaforov> show ipv6
NAME           : PC2-ngaforov[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

PC2-ngaforov>

```

Рис. 2.3: Конфигурация PC2

2.3.3 Настройка IPv4 на сервере с двойным стеком

Назначен IP-адрес и шлюз для сервера:

```
ip 64.100.1.10/24 64.100.1.1
```

save

```

server-ngaforov - PuTTY
VPCS> save
Saving startup configuration to startup.vpc
. done

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      : 10030
RHOST:PORT : 127.0.0.1:10031
MTU        : 1500

VPCS> show ipv6

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          : 10030
RHOST:PORT     : 127.0.0.1:10031
MTU            : 1500

VPCS>

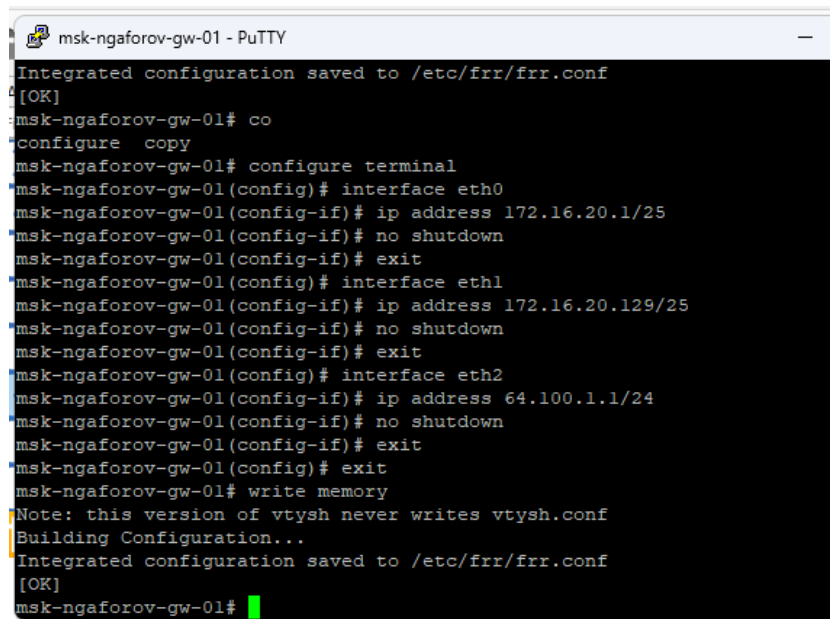
```

Рис. 2.4: Сервер Dual Stack

2.3.4 Настройка маршрутизатора FRR (msk-ngaforov-gw-01)

На маршрутизаторе **FRR** настроены интерфейсы локальных сетей:

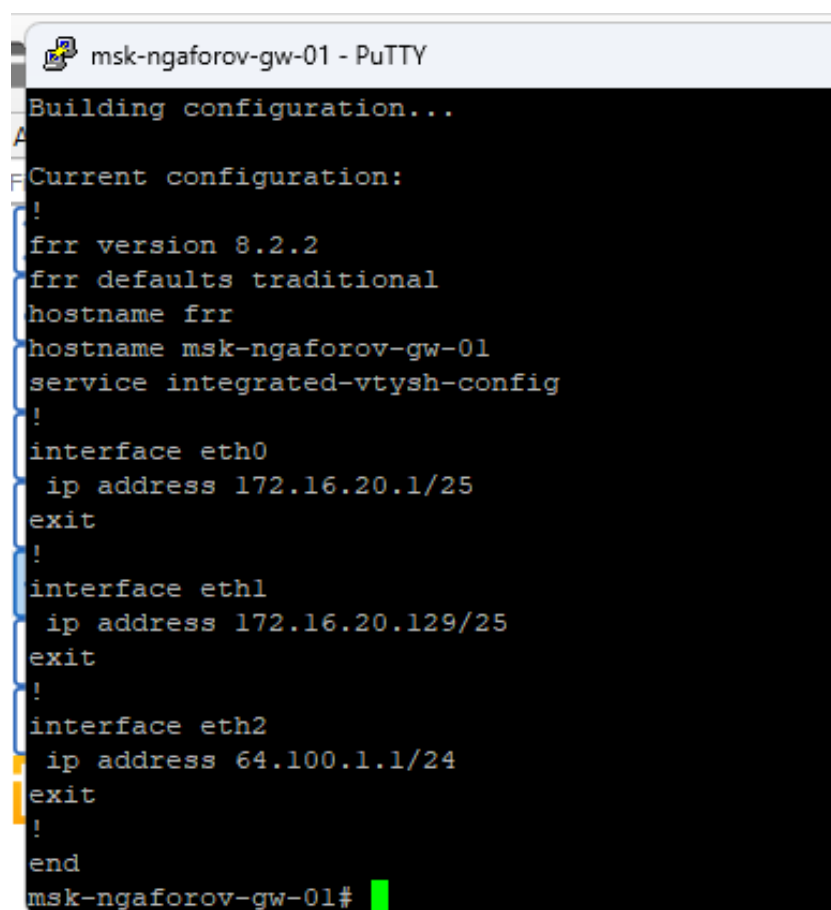
Интерфейс	Назначение сегмента	IPv4-адрес
eth0	Подсеть PC1	172.16.20.1/25
eth1	Подсеть PC2	172.16.20.129/25
eth2	Подсеть сервера	64.100.1.1/24



```
msk-ngaforov-gw-01 - PuTTY
Integrated configuration saved to /etc/frr/frr.conf
[OK]
msk-ngaforov-gw-01# co
configure copy
msk-ngaforov-gw-01# configure terminal
msk-ngaforov-gw-01(config)# interface eth0
msk-ngaforov-gw-01(config-if)# ip address 172.16.20.1/25
msk-ngaforov-gw-01(config-if)# no shutdown
msk-ngaforov-gw-01(config-if)# exit
msk-ngaforov-gw-01(config)# interface eth1
msk-ngaforov-gw-01(config-if)# ip address 172.16.20.129/25
msk-ngaforov-gw-01(config-if)# no shutdown
msk-ngaforov-gw-01(config-if)# exit
msk-ngaforov-gw-01(config)# interface eth2
msk-ngaforov-gw-01(config-if)# ip address 64.100.1.1/24
msk-ngaforov-gw-01(config-if)# no shutdown
msk-ngaforov-gw-01(config-if)# exit
msk-ngaforov-gw-01(config)# exit
msk-ngaforov-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-ngaforov-gw-01#
```

Рис. 2.5: Назначение адресов на FRR

Просмотр текущей конфигурации маршрутизатора:



```
msk-ngaforov-gw-01 - PuTTY
Building configuration...
Current configuration:
!
frr version 8.2.2
frr defaults traditional
hostname frr
hostname msk-ngaforov-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-ngaforov-gw-01#
```

Рис. 2.6: show running-config

Просмотр состояния интерфейсов:

```
msk-ngaforov-gw-01 - PuTTY
interface eth1
ip address 172.16.20.129/25
exit
!
interface eth2
ip address 64.100.1.1/24
exit
!
end
msk-ngaforov-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-ngaforov-gw-01#
```

Рис. 2.7: show interface brief

2.3.5 Проверка связности между узлами

Проверка с PC1

Выполнены ICMP-эхо-запросы к PC2 и серверу, ответы получены успешно.
Трассировка показывает прохождение через маршрутизатор FRR.

```
PC1-ngaforov - PuTTY

PC1-ngaforov> ping 172.16.20.138

84 bytes from 172.16.20.138 icmp_seq=1 ttl=63 time=4.180 ms
84 bytes from 172.16.20.138 icmp_seq=2 ttl=63 time=5.619 ms
84 bytes from 172.16.20.138 icmp_seq=3 ttl=63 time=3.362 ms
84 bytes from 172.16.20.138 icmp_seq=4 ttl=63 time=1.612 ms
84 bytes from 172.16.20.138 icmp_seq=5 ttl=63 time=4.129 ms

PC1-ngaforov> ping 64.100.1.10

84 bytes from 64.100.1.10 icmp_seq=1 ttl=63 time=9.854 ms
84 bytes from 64.100.1.10 icmp_seq=2 ttl=63 time=5.910 ms
84 bytes from 64.100.1.10 icmp_seq=3 ttl=63 time=3.891 ms
84 bytes from 64.100.1.10 icmp_seq=4 ttl=63 time=1.788 ms
84 bytes from 64.100.1.10 icmp_seq=5 ttl=63 time=1.415 ms

PC1-ngaforov> trace 172.16.20.138
trace to 172.16.20.138, 8 hops max, press Ctrl+C to stop
 1  172.16.20.1    1.644 ms  0.562 ms  0.512 ms
 2  *172.16.20.138  1.423 ms (ICMP type:3, code:3, Destination port unreachable)

PC1-ngaforov> trace 64.100.1.10
trace to 64.100.1.10, 8 hops max, press Ctrl+C to stop
 1  172.16.20.1    2.909 ms  2.544 ms  1.128 ms
 2  *64.100.1.10   1.356 ms (ICMP type:3, code:3, Destination port unreachable)

PC1-ngaforov>
```

Рис. 2.8: ping+trace с PC1

Проверка с PC2

PC2 успешно отправляет ICMP-эхо-запросы к PC1 и серверу. Ответы получены.

```
PC2-ngaforov - PuTTY

PC2-ngaforov> ping 172.16.20.10

84 bytes from 172.16.20.10 icmp_seq=1 ttl=63 time=5.396 ms
84 bytes from 172.16.20.10 icmp_seq=2 ttl=63 time=2.784 ms
84 bytes from 172.16.20.10 icmp_seq=3 ttl=63 time=2.002 ms
84 bytes from 172.16.20.10 icmp_seq=4 ttl=63 time=2.832 ms
84 bytes from 172.16.20.10 icmp_seq=5 ttl=63 time=1.864 ms

PC2-ngaforov> trace 172.16.20.10
trace to 172.16.20.10, 8 hops max, press Ctrl+C to stop
 1  172.16.20.129  2.994 ms  0.676 ms  0.530 ms
 2  *172.16.20.10  1.423 ms (ICMP type:3, code:3, Destination port unreachable)

PC2-ngaforov> ping 64.100.1.10

84 bytes from 64.100.1.10 icmp_seq=1 ttl=63 time=4.923 ms
84 bytes from 64.100.1.10 icmp_seq=2 ttl=63 time=1.944 ms
84 bytes from 64.100.1.10 icmp_seq=3 ttl=63 time=4.682 ms
84 bytes from 64.100.1.10 icmp_seq=4 ttl=63 time=1.862 ms
84 bytes from 64.100.1.10 icmp_seq=5 ttl=63 time=6.133 ms

PC2-ngaforov> trace 64.100.1.10
trace to 64.100.1.10, 8 hops max, press Ctrl+C to stop
 1  172.16.20.129  0.984 ms  0.929 ms  0.887 ms
 2  *64.100.1.10   2.262 ms (ICMP type:3, code:3, Destination port unreachable)

PC2-ngaforov>
```

Рис. 2.9: ping+trace с PC2

2.3.6 Настройка IPv6 на PC3, PC4 и сервере Dual Stack

В соответствии с таблицей задания выполнена настройка IPv6-адресов на узлах PC3, PC4 и сервере с двойным стеком.

На узле PC3 назначен IPv6-адрес с префиксом /64:

```
ip 2001:db8:c0de:12::a/64
```

save

После настройки просмотрена конфигурация IPv4/IPv6 (show ip, show ipv6). IPv4 отсутствует, IPv6 назначен корректно.

Рис. 2.10: Конфигурация PC3 IPv6

На PC4 назначен IPv6-адрес из другой подсети:

```
ip 2001:db8:c0de:13::a/64
```

save

Просмотр конфигурации подтверждает корректность назначения.

```
PC4-ngaforov> save
Saving startup configuration to startup.vpc
done

PC4-ngaforov> show ip
NAME       : PC4-ngaforov[1]
IP/MASK    : 0.0.0.0/0
GATEWAY    : 0.0.0.0
DNS        :
MAC        : 00:50:79:66:68:03
LPORT      : 10028
RHOST:PORT : 127.0.0.1:10029
MTU        : 1500

PC4-ngaforov> show ipv6
NAME           : PC4-ngaforov[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          : 10028
RHOST:PORT     : 127.0.0.1:10029
MTU            : 1500

PC4-ngaforov>
```

Рис. 2.11: Конфигурация PC4 IPv6

Сервер уже содержал IPv4-адрес, дополнительно назначен IPv6:

ip 2001:db8:c0de:11::a/64

save

```
server-ngaforov - PuTTY
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      : 10030
RHOST:PORT : 127.0.0.1:10031
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         : 10030
RHOST:PORT    : 127.0.0.1:10031
MTU           : 1500

VPCS>
```

Рис. 2.12: Конфигурация сервера IPv6

2.3.7 Настройка IPv6 на маршрутизаторе VyOS (msk-ngaforov-gw-02)

Маршрутизатору VyOS назначены адреса IPv6 для всех интерфейсов, а также включено распространение RA (Router Advertisement), чтобы узлы могли получать параметры автоматически.

Назначенные адреса:

Интерфейс	Назначение сегмента	IPv6-адрес
eth0	подсеть PC3	2001:db8:c0de:12::1/64
eth1	подсеть PC4	2001:db8:c0de:13::1/64
eth2	подсеть Dual Stack Server	2001:db8:c0de:11::1/64

Рис. 2.13: Настройка адресов на VyOS

После применения конфигурации выполнены команды commit и save, затем проверены интерфейсы:

Рис. 2.14: Состояние интерфейсов VyOS

2.3.8 Проверка связности по IPv6

На PC3 выполнен ping и trace до IPv6-адресов PC4 и сервера. Ответы получены успешно — маршрутизация через VyOS работает.

```
PC3-ngaforov> ping 2001:db8:c0de:13::a

2001:db8:c0de:13::a icmp6_seq=1 ttl=62 time=7.308 ms
2001:db8:c0de:13::a icmp6_seq=2 ttl=62 time=2.295 ms
2001:db8:c0de:13::a icmp6_seq=3 ttl=62 time=1.624 ms
2001:db8:c0de:13::a icmp6_seq=4 ttl=62 time=7.343 ms
2001:db8:c0de:13::a icmp6_seq=5 ttl=62 time=2.372 ms

PC3-ngaforov> trace 2001:db8:c0de:13::a

trace to 2001:db8:c0de:13::a, 64 hops max
 1 2001:db8:c0de:12::1  2.770 ms  0.957 ms  0.932 ms
 2 2001:db8:c0de:13::a  1.434 ms  1.272 ms  0.900 ms

PC3-ngaforov> trace 2001:db8:c0de:11::a

trace to 2001:db8:c0de:11::a, 64 hops max
 1 2001:db8:c0de:12::1  2.380 ms  1.118 ms  0.611 ms
 2 2001:db8:c0de:11::a  2.026 ms  1.125 ms  1.015 ms

PC3-ngaforov> ping 2001:db8:c0de:11::a

2001:db8:c0de:11::a icmp6_seq=1 ttl=62 time=3.666 ms
2001:db8:c0de:11::a icmp6_seq=2 ttl=62 time=1.932 ms
2001:db8:c0de:11::a icmp6_seq=3 ttl=62 time=1.880 ms
2001:db8:c0de:11::a icmp6_seq=4 ttl=62 time=2.336 ms
2001:db8:c0de:11::a icmp6_seq=5 ttl=62 time=2.485 ms

PC3-ngaforov> ping 172.16.20.138

host (172.16.20.138) not reachable

PC3-ngaforov> █
```

Рис. 2.15: PC3: проверка IPv6

На PC4 выполнены аналогичные проверки — также получены успешные ответы.

```
PC4-ngaforov>
PC4-ngaforov> ping 2001:db8:c0de:12::a

2001:db8:c0de:12::a icmp6_seq=1 ttl=62 time=2.534 ms
2001:db8:c0de:12::a icmp6_seq=2 ttl=62 time=2.703 ms
2001:db8:c0de:12::a icmp6_seq=3 ttl=62 time=1.902 ms
2001:db8:c0de:12::a icmp6_seq=4 ttl=62 time=1.685 ms
2001:db8:c0de:12::a icmp6_seq=5 ttl=62 time=4.892 ms

PC4-ngaforov> ping 2001:db8:c0de:11::a

2001:db8:c0de:11::a icmp6_seq=1 ttl=62 time=3.066 ms
2001:db8:c0de:11::a icmp6_seq=2 ttl=62 time=3.250 ms
2001:db8:c0de:11::a icmp6_seq=3 ttl=62 time=3.590 ms
2001:db8:c0de:11::a icmp6_seq=4 ttl=62 time=2.526 ms
2001:db8:c0de:11::a icmp6_seq=5 ttl=62 time=1.850 ms

PC4-ngaforov> ping 64.100.1.10

host (64.100.1.10) not reachable

PC4-ngaforov> 
```

Рис. 2.16: PC4: проверка IPv6

На сервере проверено взаимодействие с обеими IPv6-подсетями, ответы получены.

```
server-ngaforov - PuTTY

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

2001:db8:c0de:12::a icmp6_seq=1 ttl=62 time=6.356 ms
2001:db8:c0de:12::a icmp6_seq=2 ttl=62 time=4.294 ms
2001:db8:c0de:12::a icmp6_seq=3 ttl=62 time=3.550 ms
2001:db8:c0de:12::a icmp6_seq=4 ttl=62 time=2.929 ms
2001:db8:c0de:12::a icmp6_seq=5 ttl=62 time=2.267 ms

VPCS> ping 172.16.20.10

84 bytes from 172.16.20.10 icmp_seq=1 ttl=63 time=4.424 ms
84 bytes from 172.16.20.10 icmp_seq=2 ttl=63 time=5.389 ms
84 bytes from 172.16.20.10 icmp_seq=3 ttl=63 time=4.689 ms
84 bytes from 172.16.20.10 icmp_seq=4 ttl=63 time=2.225 ms
84 bytes from 172.16.20.10 icmp_seq=5 ttl=63 time=1.697 ms

VPCS> ping 2001:db8:c0de:13::a

2001:db8:c0de:13::a icmp6_seq=1 ttl=62 time=3.017 ms
2001:db8:c0de:13::a icmp6_seq=2 ttl=62 time=3.173 ms
2001:db8:c0de:13::a icmp6_seq=3 ttl=62 time=1.991 ms
2001:db8:c0de:13::a icmp6_seq=4 ttl=62 time=1.797 ms
2001:db8:c0de:13::a icmp6_seq=5 ttl=62 time=3.563 ms

VPCS> ping 172.16.20.129

84 bytes from 172.16.20.129 icmp_seq=1 ttl=64 time=2.486 ms
84 bytes from 172.16.20.129 icmp_seq=2 ttl=64 time=1.523 ms
84 bytes from 172.16.20.129 icmp_seq=3 ttl=64 time=1.064 ms
84 bytes from 172.16.20.129 icmp_seq=4 ttl=64 time=2.231 ms
84 bytes from 172.16.20.129 icmp_seq=5 ttl=64 time=2.761 ms

VPCS> █
```

Рис. 2.17: Dual Stack Server: проверки IPv6

2.3.9 Анализ захваченного трафика ARP, ICMP и ICMPv6

На соединении между сервером с двойным стеком адресации и коммутатором был выполнен захват трафика в **Wireshark**. На основании перехваченных пакетов проанализированы три протокола: ARP (для IPv4), ICMP (для IPv4) и ICMPv6 (для IPv6).

ARP используется только в IPv4-сетях и работает на канальном уровне. Во время ping-запроса сервер должен определить MAC-адрес шлюза для отправки пакета.

На перехваченном кадре видно:

- MAC-адрес отправителя (сервер)
- IPv4-адрес отправителя (64.100.1.1)
- Запрашиваемый IPv4-адрес (64.100.1.10)
- MAC-адрес назначения — 00:00:00:00:00:00 (ещё неизвестен)

No.	Time	Source	Destination	Protocol	Length	Info
47	129.042108	64.100.1.10	172.16.20.10	ICMP	98	Echo (ping) request id=0xfefba, seq=1/256, ttl=64 (reply in 48)
48	129.046162	172.16.20.10	64.100.1.10	ICMP	98	Echo (ping) reply id=0xfefba, seq=1/256, ttl=63 (request in 47)
49	130.040578	64.100.1.10	172.16.20.10	ICMP	98	Echo (ping) request id=0xfefba, seq=2/512, ttl=64 (reply in 50)
50	130.052862	172.16.20.10	64.100.1.10	ICMP	98	Echo (ping) reply id=0xfefba, seq=2/512, ttl=63 (request in 49)
51	131.054785	64.100.1.10	172.16.20.10	ICMP	98	Echo (ping) request id=0x00bb, seq=3/768, ttl=64 (reply in 52)
52	131.058696	172.16.20.10	64.100.1.10	ICMP	98	Echo (ping) reply id=0x00bb, seq=3/768, ttl=63 (request in 51)
53	132.059897	64.100.1.10	172.16.20.10	ICMP	98	Echo (ping) request id=0x01bb, seq=4/1024, ttl=64 (reply in 54)
54	132.061744	172.16.20.10	64.100.1.10	ICMP	98	Echo (ping) reply id=0x01bb, seq=4/1024, ttl=63 (request in 53)
55	133.063317	64.100.1.10	172.16.20.10	ICMP	98	Echo (ping) request id=0x02bb, seq=5/1280, ttl=64 (reply in 56)
56	133.064720	172.16.20.10	64.100.1.10	ICMP	98	Echo (ping) reply id=0x02bb, seq=5/1280, ttl=63 (request in 55)
57	134.082437	0c:61:b5:d:c:00:02	Private_66:68:04	ARP	60	who has 64.100.1.10? Tell 64.100.1.1
58	134.083506	Private_66:68:04	0c:61:b5:d:c:00:02	ARP	60	64.100.1.10 is at 00:50:79:66:68:04
59	139.800031	2001:db8::c0de:11::a	2001:db8::c0de:13::a	ICMPv6	118	Echo (ping) request id=0x09bb, seq=1, hop limit=64 (reply in 60)

> Frame 57: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface - , id 0

- Ethernet II, Src: 0c:61:b5:d:c:00:02 (0c:61:b5:d:c:00:02), Dst: Private_66:68:04 (00:50:79:66:68:04)
 - Destination: Private_66:68:04 (00:50:79:66:68:04)
 - Source: 0c:61:b5:d:c:00:02 (0c:61:b5:d:c:00:02)
 - Type: ARP (0x0806)
 - [Stream index: 4]
 - Padding: 00000000000000000000000000000000
- Address Resolution Protocol (request)
 - Hardware type: Ethernet (1)
 - Protocol type: IPv4 (0x0800)
 - Hardware size: 6
 - Protocol size: 4
 - Opcode: request (1)
 - Sender MAC address: 0c:61:b5:d:c:00:02 (0c:61:b5:d:c:00:02)
 - Sender IP address: 64.100.1.1
 - Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)
 - Target IP address: 64.100.1.10

```

0000  00 50 79 66 68 04 00 00 00 00 00 00 00 00 00 00
0010  00 00 06 00 00 00 00 00 00 00 00 00 00 00 00
0020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
    
```

Рис. 2.18: ARP-запрос

Из ARP-пакета можно извлечь:

- физический (MAC) адрес отправителя,
- IP-адрес отправителя и IP-адрес искомого узла,
- факт, что это ARP Request (opcode: request),
- тип протокола (IPv4) и тип канального уровня (Ethernet).

ICMP используется для диагностики и проверки связности сети.

После ARP-разрешения адресов происходит обмен ICMP Echo Request / Echo Reply.

В перехваченном ICMP-кадре видно:

- отправитель: 172.16.20.129
- получатель: 64.100.1.10
- тип ICMP: Echo Request
- параметры: ttl=64, checksum корректна
- также видно время отклика (Response time)

No.	Time	Source	Destination	Protocol	Length	Info
65	142.818766	2001:db8:c0de:11::a	2001:db8:c0de:13::a	ICMPv6	118	Echo (ping) request id=0x09bb, seq=4, hop limit=64 (reply in 66)
66	142.812267	2001:db8:c0de:13::a	2001:db8:c0de:11::a	ICMPv6	118	Echo (ping) reply id=0x09bb, seq=4, hop limit=62 (request in 65)
67	143.813939	2001:db8:c0de:11::a	2001:db8:c0de:13::a	ICMPv6	118	Echo (ping) request id=0x09bb, seq=5, hop limit=64 (reply in 68)
68	143.816873	2001:db8:c0de:13::a	2001:db8:c0de:11::a	ICMPv6	118	Echo (ping) reply id=0x09bb, seq=5, hop limit=62 (request in 67)
69	150.904903	64.100.1.10	172.16.20.129	ICMP	98	Echo (ping) request id=0x14bb, seq=1/256, ttl=64 (reply in 70)
70	150.907000	172.16.20.129	64.100.1.10	ICMP	98	Echo (ping) reply id=0x14bb, seq=1/256, ttl=64 (request in 69)
71	151.908947	64.100.1.10	172.16.20.129	ICMP	98	Echo (ping) request id=0x15bb, seq=2/512, ttl=64 (reply in 72)
72	151.910060	172.16.20.129	64.100.1.10	ICMP	98	Echo (ping) reply id=0x15bb, seq=2/512, ttl=64 (request in 71)
73	152.911924	64.100.1.10	172.16.20.129	ICMP	98	Echo (ping) request id=0x16bb, seq=3/768, ttl=64 (reply in 74)
74	152.912662	172.16.20.129	64.100.1.10	ICMP	98	Echo (ping) reply id=0x16bb, seq=3/768, ttl=64 (request in 73)
75	153.913439	64.100.1.10	172.16.20.129	ICMP	98	Echo (ping) request id=0x17bb, seq=4/1024, ttl=64 (reply in 76)
76	153.915011	172.16.20.129	64.100.1.10	ICMP	98	Echo (ping) reply id=0x17bb, seq=4/1024, ttl=64 (request in 75)
77	154.916613	64.100.1.10	172.16.20.129	ICMP	98	Echo (ping) request id=0x18bb, seq=5/1280, ttl=64 (reply in 78)

<p>> Frame 70: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface -, id 0</p> <p>Ethernet II, Src: 0c:61:b5:dc:00:02 (0c:61:b5:dc:00:02), Dst: Private_66:68:04 (00:50:79:66:68:04)</p> <p>> Destination: Private_66:68:04 (00:50:79:66:68:04)</p> <p>> Source: 0c:61:b5:dc:00:02 (0c:61:b5:dc:00:02)</p> <p>Type: IPv4 (0x0800)</p> <p>[Stream index: 4]</p> <p>Internet Protocol Version 4, Src: 172.16.20.129, Dst: 64.100.1.10</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: 0xcfa8 (53064)</p> <p>000. = Flags: 0x0</p> <p>...0 0000 0000 0000 = Fragment Offset: 0</p> <p>Time to Live: 64</p> <p>Protocol: ICMP (1)</p> <p>Header Checksum: 0xa961 [validation disabled]</p> <p>[Header checksum status: Unverified]</p> <p>Source Address: 172.16.20.129</p> <p>Destination Address: 64.100.1.10</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: 0x1350 [correct]</p> <p>[Checksum Status: Good]</p> <p>Identifier (BE): 5307 (0x14bb)</p> <p>Identifier (LE): 47892 (0xbb14)</p> <p>Sequence Number (BE): 1 (0x0001)</p> <p>Sequence Number (LE): 256 (0x0100)</p> <p>[Request frame: 69]</p> <p>[Response time: 2,097 ms]</p> <p>> Data (56 bytes)</p>	<p>0000 00 50 79</p> <p>0010 00 54 cf</p> <p>0020 01 0a 00</p> <p>0030 0e 0f 10</p> <p>0040 1e 1f 20</p> <p>0050 2e 2f 30</p> <p>0060 3e 3f</p>
--	---

Рис. 2.19: ICMP Echo (IPv4)

Из ICMP-пакета можно извлечь:

- исходный и целевой IPv4-адрес,
- TTL (время жизни) — сколько маршрутизаторов может пройти пакет,
- информацию о типе ICMP (запрос или ответ),
- задержку (время отклика).

ICMPv6 протокол выполняет в IPv6 ту же роль, что ICMP в IPv4 — диагностика и контроль.

При обмене эхо-запросами между хостами была перехвачена ICMPv6 Echo Reply.

В пакете видно:

- IPv6-адрес источника: 2001:db8:c0de:13::a
- IPv6-адрес назначения: 2001:db8:c0de:11::a
- тип сообщения: Echo (ping) reply
- hop-limit (аналог TTL)

No.	Time	Source	Destination	Protocol	Length	Info
54	132.061744	172.16.20.10	64.100.1.10	ICMP	98	Echo (ping) reply id=0x01bb, seq=4/1024, ttl=63 (request in 53)
55	133.063317	64.100.1.10	172.16.20.10	ICMP	98	Echo (ping) request id=0x02bb, seq=5/1280, ttl=64 (reply in 56)
56	133.064720	172.16.20.10	64.100.1.10	ICMP	98	Echo (ping) reply id=0x02bb, seq=5/1280, ttl=63 (request in 55)
57	134.082437	0c:61:b5:dc:00:02	Private_66:68:04	ARP	60	Who has 64.100.1.10? Tell 64.100.1.1
58	134.083506	Private_66:68:04	0c:61:b5:dc:00:02	ARP	60	64.100.1.10 is at 00:50:79:66:68:04
59	139.080031	2001:db8:c0de:13::a	2001:db8:c0de:13::a	ICMPv6	118	Echo (ping) request id=0x09bb, seq=1, hop limit=64 (reply in 60)
60	139.080606	2001:db8:c0de:13::a	2001:db8:c0de:11::a	ICMPv6	118	Echo (ping) reply id=0x09bb, seq=1, hop limit=62 (request in 59)
61	140.080454	2001:db8:c0de:11::a	2001:db8:c0de:13::a	ICMPv6	118	Echo (ping) request id=0x09bb, seq=2, hop limit=64 (reply in 62)
62	140.080762	2001:db8:c0de:13::a	2001:db8:c0de:11::a	ICMPv6	118	Echo (ping) reply id=0x09bb, seq=2, hop limit=62 (request in 61)
63	141.080892	2001:db8:c0de:11::a	2001:db8:c0de:13::a	ICMPv6	118	Echo (ping) request id=0x09bb, seq=3, hop limit=64 (reply in 64)
64	141.080970	2001:db8:c0de:13::a	2001:db8:c0de:11::a	ICMPv6	118	Echo (ping) reply id=0x09bb, seq=3, hop limit=62 (request in 63)
65	142.810766	2001:db8:c0de:11::a	2001:db8:c0de:13::a	ICMPv6	118	Echo (ping) request id=0x09bb, seq=4, hop limit=64 (reply in 66)
66	142.812267	2001:db8:c0de:13::a	2001:db8:c0de:11::a	ICMPv6	118	Echo (ping) reply id=0x09bb, seq=4, hop limit=62 (request in 65)

> Frame 62: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface -, id 0	0000 00 50 79 66 6
> Ethernet II, Src: 0c:71:94:92:00:02 (0c:71:94:92:00:02), Dst: Private_66:68:04 (00:50:79:66:68:04)	0010 00 00 00 40 3
> Destination: Private_66:68:04 (00:50:79:66:68:04)	0020 00 00 00 00 0
> Source: 0c:71:94:92:00:02 (0c:71:94:92:00:02)	0030 00 00 00 00 0
Type: IPv6 (0x86dd)	0040 02 03 04 05 0
[Stream index: 2]	0050 12 13 14 15 1
> Internet Protocol Version 6, Src: 2001:db8:c0de:13::a, Dst: 2001:db8:c0de:11::a	0060 22 23 24 25 2
> 0000 0000 = Version: 6	0070 32 33 34 35 3
> 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: 62	
> Source Address: 2001:db8:c0de:13::a	
> Destination Address: 2001:db8:c0de:11::a	
[Stream index: 4]	
> Internet Control Message Protocol v6	
Type: Echo (ping) reply (129)	
Code: 0	
Checksum: 0xa04d [correct]	
[Checksum Status: Good]	
Identifier: 0x09bb	
Sequence: 2	
[Response To: 61]	
[Response Time: 2,708 ms]	
> Data (56 bytes)	

Рис. 2.20: ICMPv6 Echo (IPv6)

Из ICMPv6-пакета можно извлечь:

- исходный и целевой IPv6-адрес,
- hop limit (аналог TTL в IPv6),
- тип ICMPv6-сообщения (Echo Request или Echo Reply),
- время отклика.

2.4 Самостоятельное задание: настройка двух подсетей IPv4/IPv6 и проверка связности

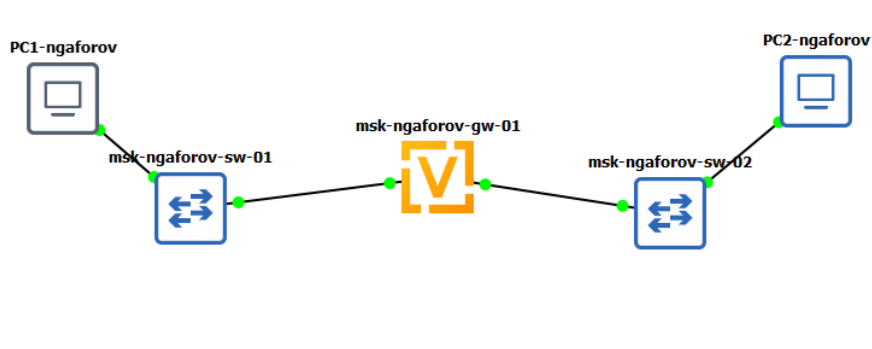


Рис. 2.21: Топология

2.4.1 Подсеть 1 (IPv4)

- Сеть: **10.10.1.96/27** (маска 255.255.255.224)
- Диапазон хостов: **10.10.1.97 – 10.10.1.126**
- Широковещательный адрес: **10.10.1.127**
- Кол-во адресов для хостов: **30**

2.4.2 Подсеть 2 (IPv4)

- Сеть: **10.10.1.16/28** (маска 255.255.255.240)
- Диапазон хостов: **10.10.1.17 – 10.10.1.30**
- Широковещательный адрес: **10.10.1.31**
- Кол-во адресов для хостов: **14**

2.4.3 Подсети IPv6

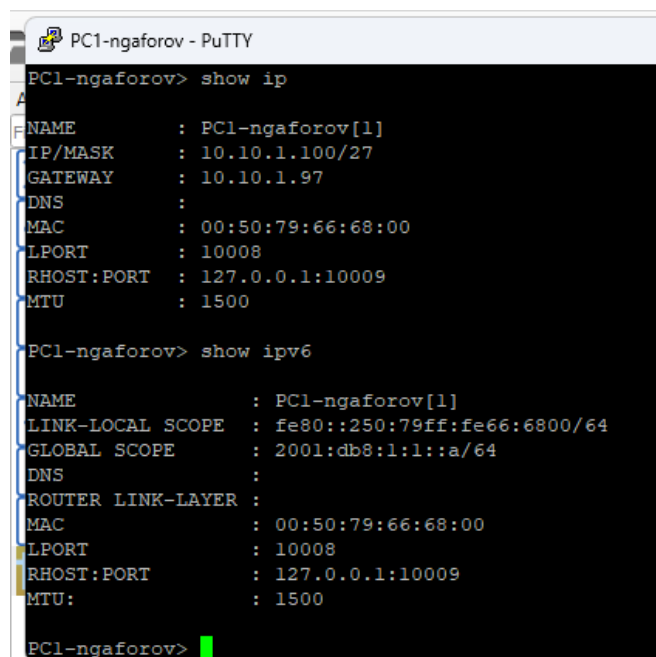
- Подсеть 1: **2001:db8:1:1::/64**
 - Маршрутизатор: **2001:db8:1:1::1**
 - Пример хоста (PC1): **2001:db8:1:1::a**

- В IPv6 нет broadcast; используется **multicast** и **ICMPv6/ND**.
- Подсеть 2: **2001:db8:1:4::/64**
 - Маршрутизатор: **2001:db8:1:4::1**
 - Пример хоста (PC2): **2001:db8:1:4::a**

2.4.4 Таблица адресации

Устройство /			
Интерфейс	Подсеть	IPv4	IPv6
VyOS msk-ngaforov-gw-01 – eth0	Подсеть 1	10.10.1.97/27	2001:db8:1:1::1/64
VyOS msk-ngaforov-gw-01 – eth1	Подсеть 2	10.10.1.17/28	2001:db8:1:4::1/64
PC1	Подсеть 1	10.10.1.100/27, GW 10.10.1.97	2001:db8:1:1::a/64
PC2	Подсеть 2	10.10.1.20/28, GW 10.10.1.17	2001:db8:1:4::a/64
Коммутаторы	L2	—	—

2.4.5 Настройка устройств

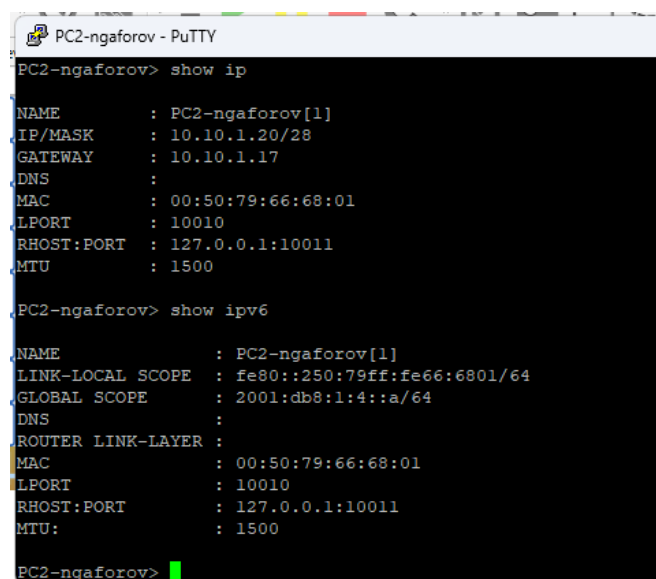


```
PC1-ngaforov> show ip
NAME       : PC1-ngaforov[1]
IP/MASK    : 10.10.1.100/27
GATEWAY    : 10.10.1.97
DNS        :
MAC        : 00:50:79:66:68:00
LPORT      : 10008
RHOST:PORT : 127.0.0.1:10009
MTU        : 1500

PC1-ngaforov> show ipv6
NAME       : PC1-ngaforov[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

PC1-ngaforov>
```

Рис. 2.22: PC1: show ip/ipv6



```
PC2-ngaforov> show ip
NAME       : PC2-ngaforov[1]
IP/MASK    : 10.10.1.20/28
GATEWAY    : 10.10.1.17
DNS        :
MAC        : 00:50:79:66:68:01
LPORT      : 10010
RHOST:PORT : 127.0.0.1:10011
MTU        : 1500

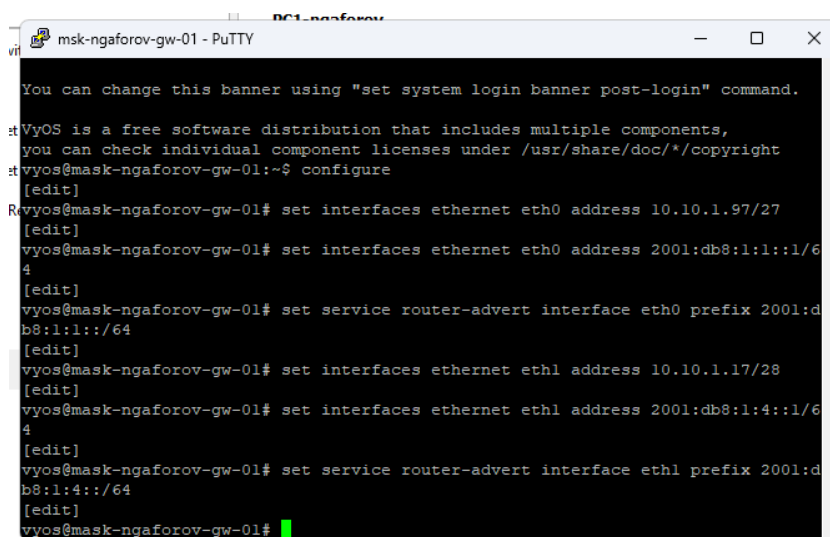
PC2-ngaforov> show ipv6
NAME       : PC2-ngaforov[1]
LINK-LOCAL SCOPE : fe80::250:79ff:fe66:6801/64
GLOBAL SCOPE   : 2001:db8:1:4::a/64
DNS          :
ROUTER LINK-LAYER :
MAC          : 00:50:79:66:68:01
LPORT        : 10010
RHOST:PORT    : 127.0.0.1:10011
MTU          : 1500

PC2-ngaforov>
```

Рис. 2.23: PC2: show ip/ipv6

Маршрутизатор VyOS - На **eth0**: 10.10.1.97/27, 2001:db8:1:1::1/64, включён RA для 2001:db8:1:1::/64 - На **eth1**: 10.10.1.17/28, 2001:db8:1:4::1/64, включён

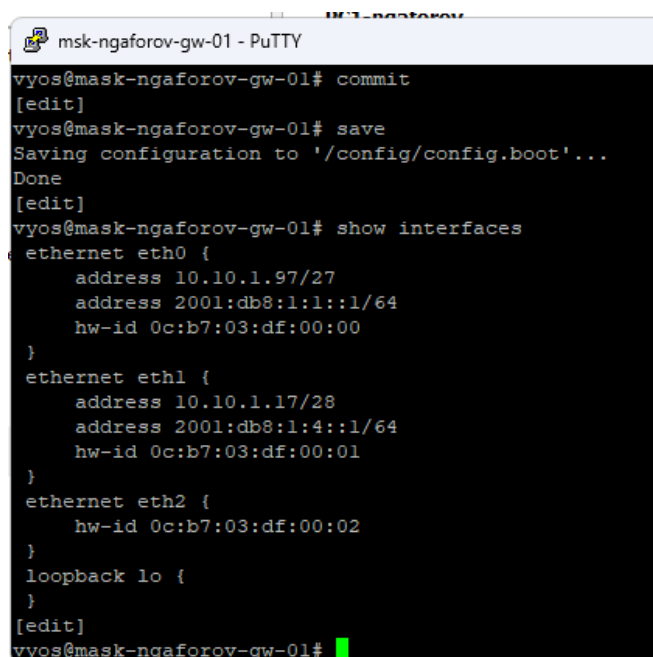
РА для 2001:db8:1:4::/64



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

Рис. 2.24: VyOS

Проверка интерфейсов после commit/save:



```
msk-ngaforov-gw-01 - PuTTY
vyos@mask-ngaforov-gw-01# commit
[edit]
vyos@mask-ngaforov-gw-01# save
Saving configuration to '/config/config.boot'...
Done
[edit]
vyos@mask-ngaforov-gw-01# show interfaces
ethernet eth0 {
    address 10.10.1.97/27
    address 2001:db8:1:1::1/64
    hw-id 0c:b7:03:df:00:00
}
ethernet eth1 {
    address 10.10.1.17/28
    address 2001:db8:1:4::1/64
    hw-id 0c:b7:03:df:00:01
}
ethernet eth2 {
    hw-id 0c:b7:03:df:00:02
}
loopback lo {
}
[edit]
vyos@mask-ngaforov-gw-01#
```

Рис. 2.25: VyOS: show interfaces

2.4.6 Проверка связности (ping/trace)

С PC1: - **IPv4**: успешный ping до **PC2 (10.10.1.20)** - **IPv6**: успешный ping до **PC2 (2001:db8:1:4::a)**

```
PC1-ngaforov> ping 10.10.1.20
84 bytes from 10.10.1.20 icmp_seq=1 ttl=63 time=3.714 ms
84 bytes from 10.10.1.20 icmp_seq=2 ttl=63 time=2.351 ms
84 bytes from 10.10.1.20 icmp_seq=3 ttl=63 time=2.748 ms
84 bytes from 10.10.1.20 icmp_seq=4 ttl=63 time=5.354 ms
84 bytes from 10.10.1.20 icmp_seq=5 ttl=63 time=1.919 ms

PC1-ngaforov> ping 2001:db8:1:4::a
2001:db8:1:4::a icmp6_seq=1 ttl=62 time=6.078 ms
2001:db8:1:4::a icmp6_seq=2 ttl=62 time=3.623 ms
2001:db8:1:4::a icmp6_seq=3 ttl=62 time=2.383 ms
2001:db8:1:4::a icmp6_seq=4 ttl=62 time=3.066 ms
2001:db8:1:4::a icmp6_seq=5 ttl=62 time=3.962 ms

PC1-ngaforov> █
```

Рис. 2.26: PC1: ping IPv4/IPv6

3 Вывод

В ходе работы был проанализирован процесс установления соединения по протоколу **TCP** с использованием анализатора трафика **Wireshark**. Исследование показало, что трёхэтапное рукопожатие (**Three-Way Handshake**) обеспечивает надёжное установление соединения между клиентом и сервером перед началом передачи данных. В результате анализа зафиксированы стадии обмена сегментами **SYN**, **SYN/ACK** и **ACK**, подтверждающие успешную синхронизацию номеров последовательности и готовность сторон к коммуникации. Полученные данные подтверждают корректную реализацию транспортного уровня в модели OSI и демонстрируют принцип работы TCP при передаче HTTP-запросов.