

# **Отчет по лабораторной работе 6**

Генералов Даниил, НПИбд-01-21, 1032202280

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>4</b>	<b>Выводы</b>	<b>24</b>

# Список иллюстраций

3.1	gns	. . . . .	9
3.2	gns	. . . . .	10
3.3	gns	. . . . .	11
3.4	gns	. . . . .	12
3.5	gns	. . . . .	12
3.6	gns	. . . . .	13
3.7	gns	. . . . .	14
3.8	gns	. . . . .	15
3.9	gns	. . . . .	16
3.10	gns	. . . . .	17
3.11	wireshark	. . . . .	18
3.12	gns	. . . . .	19
3.13	wireshark	. . . . .	20
3.14	gns	. . . . .	21
3.15	gns	. . . . .	21
3.16	gns	. . . . .	22
3.17	gns	. . . . .	23

## **Список таблиц**

# 1 Цель работы

Изучение принципов распределения и настройки адресного пространства на устройствах сети.

## 2 Задание

6.3.1. Разбиение сети на подсети 6.3.2. Настройка двойного стека адресации IPv4 и IPv6 в локальной сети 6.3.3. Задание для самостоятельного выполнения

### 3 Выполнение лабораторной работы

Сначала мы делаем вычисления над подсетями IPv4:

- Для сети 172.16.20.0/24 префикс равен 172.16.20.0, потому что маска подсети равна 255.255.255.0 (24 бита сначала равны 1); broadcast-адрес равен 172.16.20.255 (все биты хоста равны 1), число подсетей  $2^{24} = 16777216$ , а хосты могут иметь адреса от 172.16.20.1 до 172.16.20.254 включительно. Чтобы разбить эту сеть на подсети с 126, 62 и 62 узла, эти сети должны иметь длину префикса 25, 26 и 26 битов соответственно; например, это может быть 172.16.20.0/25 + 172.16.20.128/26 + 172.16.20.192/26.
- Для сети 10.10.1.64/26 префикс равен 10.10.1.64, маска подсети равна 255.255.255.192 (26 битов равны 1); broadcast-адрес – 10.10.1.127, число подсетей  $2^{26} = 67108864$ , и хосты могут иметь адреса от 10.10.1.65 до 10.10.1.126 включительно. Подсеть этой сети с 30 хостами будет иметь маску длиной 27, например 10.10.1.64/27: здесь, маска подсети равна 255.255.255.224, broadcast-адрес – 10.10.1.95, а хосты – от 10.10.1.65 до 10.10.1.94.
- Для сети 10.10.1.0/26 префикс равен 10.10.1.0, маска подсети равна 255.255.255.192 (26 битов равны 1); broadcast-адрес – 10.10.1.63, число подсетей  $2^{26} = 67108864$ , и хосты могут иметь адреса от 10.10.1.1 до 10.10.1.62 включительно. Подсеть с 14 хостами будет иметь маску длиной 28, например 10.10.1.0/28: здесь, маска подсети равна 255.255.255.240, broadcast-адрес – 10.10.1.15, а хосты – от 10.10.1.1 до 10.10.1.14.

После этого мы делаем вычисления над сетями IPv6:

- Сеть 2001:db8:c0de::/48 является подсетью зарезервированной для документации сети (а также одним из глобальных адресов), в которой полностью задан префикс глобальной маршрутизации. Его маска равна ffff:ffff:ffff:: (48 битов сначала равны 1), префикс сети равен 2001:db8:c0de::, и хосты могут иметь адреса от 2001:db8:c0de:0:0:0:0 до 2001:db8:c0de:ffff:ffff:ffff:ffff. Можно выделить две подсети, либо на подсети с маской /64 с помощью идентификатора подсети (например, когда он равен 0 и 1: 2001:db8:c0de:0::/64 и 2001:db8:c0de:1::/64), либо на подсети с маской /68 с помощью первых 4 бит идентификатора интерфейса (например, когда они равны a или b: 2001:db8:c0de:1337:a000::/68 и 2001:db8:c0de:1337:b000::/68).
- Сеть 2a02:6b8::/64 является глобальным адресом, в котором полностью задан префикс глобальной маршрутизации и номер подсети, но не определен интерфейс. Его маска равна ffff:ffff:ffff:ffff:: (64 бита сначала равны 1), префикс сети равен 2a02:6b8:0:0::, и хосты (а именно интерфейсы) могут иметь адреса от 2a02:6b8:0:0:0:0:0:0 до 2a02:6b8:0:0:ffff:ffff:ffff:ffff. Саму эту сеть невозможно разбить с помощью идентификатора подсети, потому что идентификатор подсети находится внутри номера сети; если идентификатор подсети изменить, то получатся адреса вроде 2a02:6b8:0:1337::/64 и 2a02:6b8:0:6942::/64, которые не пересекаются с исходной подсетью. Однако разбить эту подсеть с помощью идентификатора интерфейса можно: например, если выделить для этого 4 бита, то можно выделить подсети 2a02:6b8:0:0:a000::/68 и 2a02:6b8:0:0:b000::/68.

После этого, мы можем приступить к работе в GNS3. Сначала нужно разместить устройства так, как показано на диаграмме.



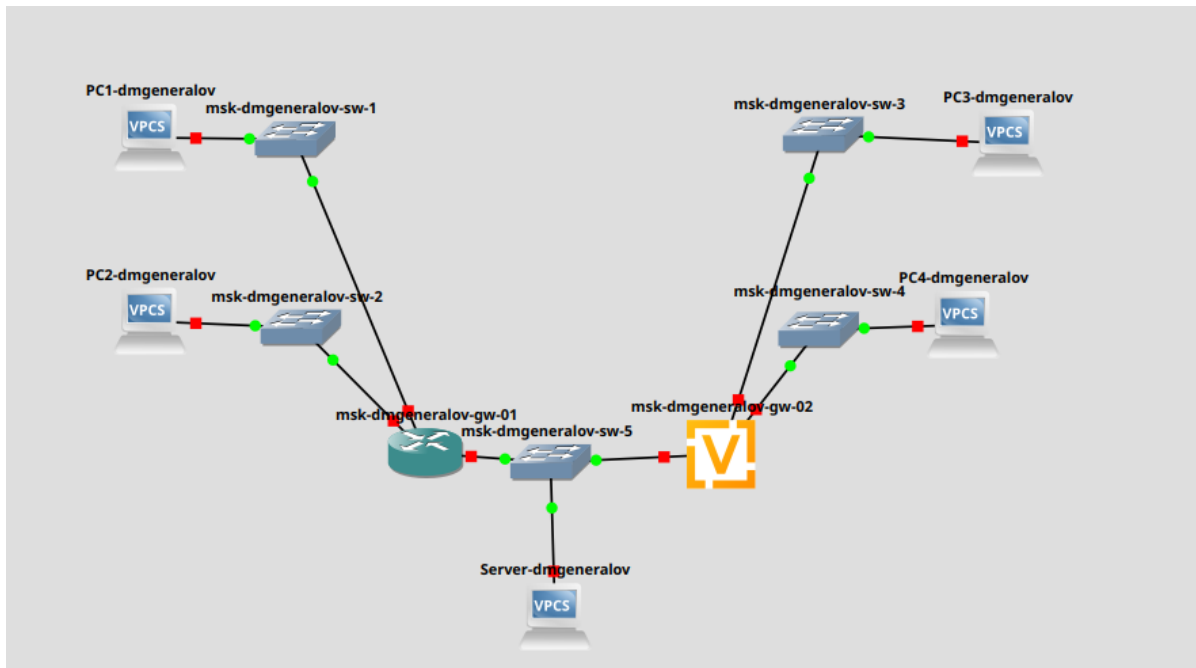


Рис. 3.1: gns

Затем я включил половину сети для IPv4 и настроил там IP-адреса. Из характеристик VPCS видно, что у них есть IPv4-адреса, и только link-local адреса IPv6 (то есть, их нельзя использовать для связи с внешним интернетом, а только с локальной сетью).

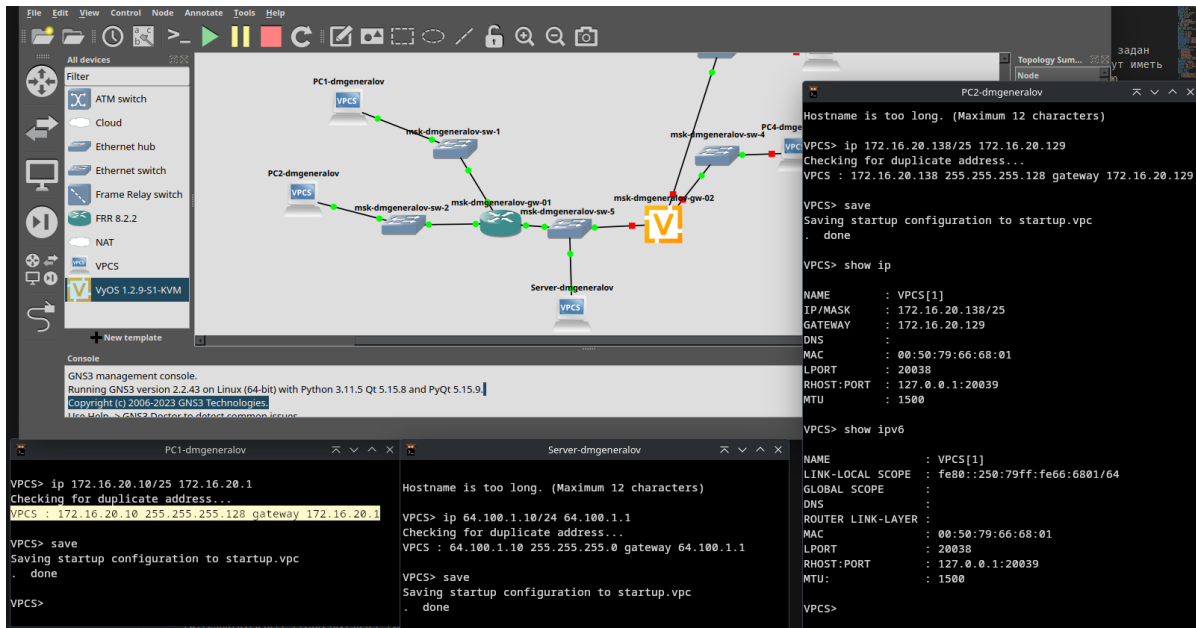


Рис. 3.2: gns

После этого я настроил роутер, чтобы он имел правильные IP-адреса на всех интерфейсах, и подтвердил, что они были настроены.

The screenshot displays the GNS3 network simulation environment. On the left, a network topology is visible with several nodes connected. The nodes include two VPCS (Virtual PCs), a Server-dmgeneralov, and several routers labeled msk-dmgeneralov-gw-01 through msk-dmgeneralov-gw-05. The topology shows a central router (gw-01) connected to other routers and VPCS. A red line indicates a specific path or connection.

On the right, a terminal window shows the configuration of msk-dmgeneralov-gw-01. The configuration includes setting interfaces eth0, eth1, and eth2 with specific IP addresses and subnets. The configuration is saved to /etc/frr/frr.conf.

```

msk-dmgeneralov-gw-01# conf t
msk-dmgeneralov-gw-01(config)# interface eth0
msk-dmgeneralov-gw-01(config-if)# ip address 64.100.1.1/24
msk-dmgeneralov-gw-01(config-if)# no sh
msk-dmgeneralov-gw-01(config-if)# exit
msk-dmgeneralov-gw-01(config)# inter eth1
msk-dmgeneralov-gw-01(config-if)# ip addr 172.16.20.129/25
msk-dmgeneralov-gw-01(config-if)# no sh
msk-dmgeneralov-gw-01(config-if)# exit
msk-dmgeneralov-gw-01(config)# inter eth2
msk-dmgeneralov-gw-01(config-if)# ip addr 172.16.20.1/25
msk-dmgeneralov-gw-01(config-if)# no sh
msk-dmgeneralov-gw-01(config-if)# exit
msk-dmgeneralov-gw-01(config)# exit
msk-dmgeneralov-gw-01# wr mem
Note: this version of vtysh never writes vtysh.conf
Building Configuration...
Integrated configuration saved to /etc/frr/frr.conf
[OK]
msk-dmgeneralov-gw-01# show running-conf
Building configuration...

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

```

Interface	Status	VRF	Addresses
eth0	up	default	64.100.1.1/24
eth1	up	default	172.16.20.129/25
eth2	up	default	172.16.20.1/25
eth3	down	default	

The terminal also shows the GNS3 version (2.2.43) and the user's command history, including the command to start the GNS3 GUI.

Рис. 3.3: gns

Теперь, два компьютера могут успешно связываться друг с другом и с сервером. Проверка пути показывает, что все соединения проходят через роутер.

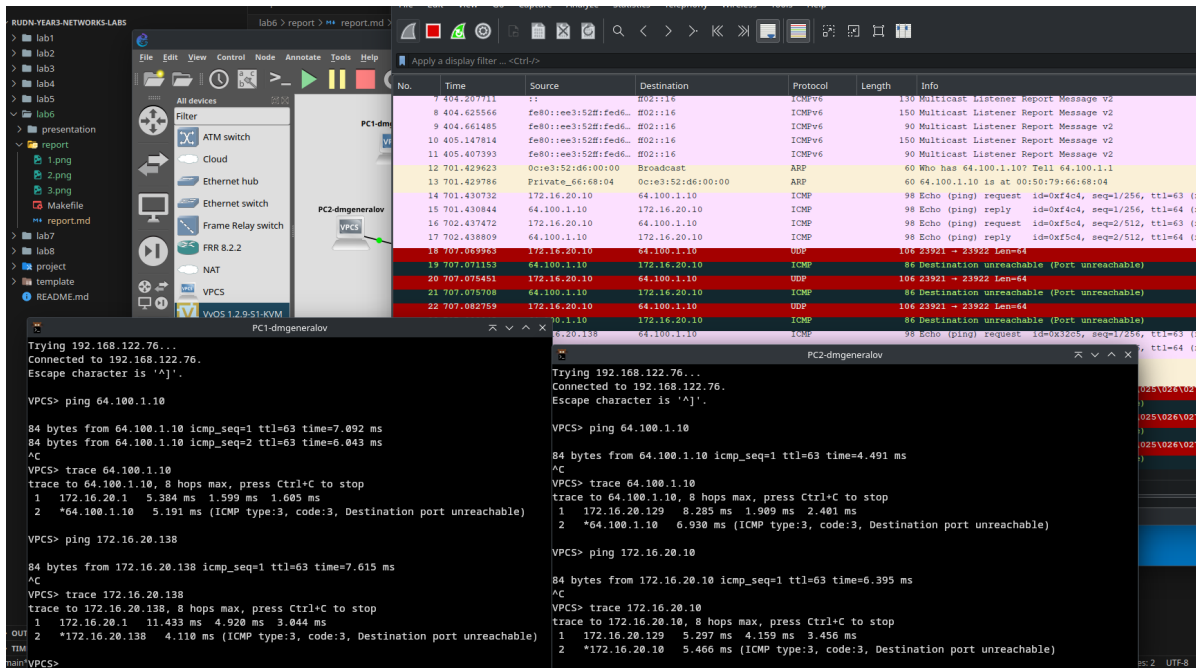


Рис. 3.4: gns

После этого нужно сделать такую же настройку с противоположной стороны сети: два компьютера и сервер должны получить свои IPv6-адреса. Здесь компьютеры имеют только IPv6 адрес, и поэтому их IPv4-адрес равен 0.0.0.0.

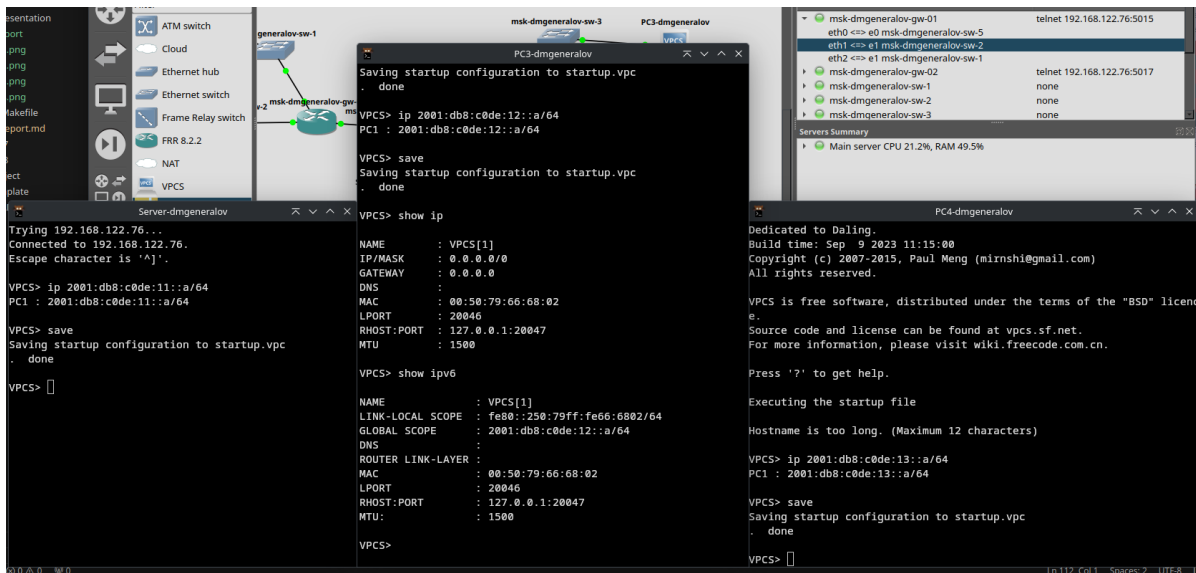
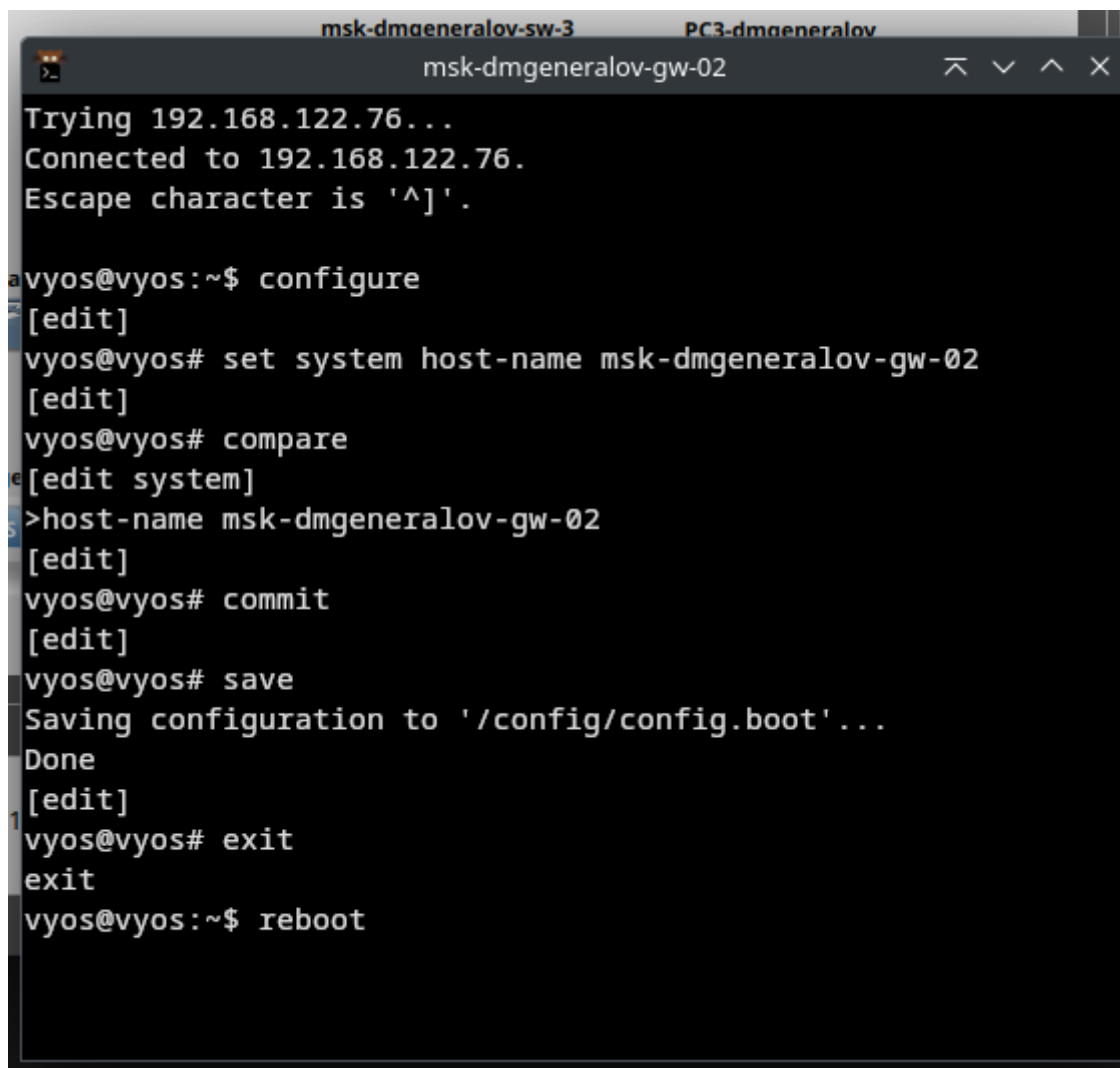


Рис. 3.5: gns

Затем мы запускаем роутер на VyOS и задаем его hostname.



```
msk-dmgeneralov-sw-3  PC3-dmgeneralov
msk-dmgeneralov-gw-02
Trying 192.168.122.76...
Connected to 192.168.122.76.
Escape character is '^]'.

vyos@vyos:~$ configure
[edit]
vyos@vyos# set system host-name msk-dmgeneralov-gw-02
[edit]
vyos@vyos# compare
[edit system]
>host-name msk-dmgeneralov-gw-02
[edit]
vyos@vyos# commit
[edit]
vyos@vyos# save
Saving configuration to '/config/config.boot'...
Done
[edit]
vyos@vyos# exit
exit
vyos@vyos:~$ reboot
```

Рис. 3.6: gns

После этого мы настраиваем IP-адреса. По какой-то причине, на VyOS версии 1.2.9 не работает команда `set service router-advert`. Кажется, на этой версии эта команда отсутствует, что видно по документации: <https://docs.vyos.io/en/crux/configuration/service/index.html>

```

[edit]
[edit]sk-dmgeneralov-gw-02# set interfaces ethernet eth1 address 2001:db8:c0de:12::1
3::1/64k-dmgeneralov-gw-02# set interfaces ethernet eth2 address 2001:db8:c0de:1
[edit]
vyos@msk-dmgeneralov-gw-02# compare
[edit interfaces ethernet eth0]
+address 2001:db8:c0de:11::1/64
[edit interfaces ethernet eth1]
+address 2001:db8:c0de:12::1/64
[edit interfaces ethernet eth2]
+address 2001:db8:c0de:13::1/64
[edit]
vyos@msk-dmgeneralov-gw-02# set service router-advert

Configuration path: service [router-advert] is not valid
Set failed

[edit]
vyos@msk-dmgeneralov-gw-02# commit
[edit]
vyos@msk-dmgeneralov-gw-02# save
Saving configuration to '/config/config.boot'...
Done
[edit]
vyos@msk-dmgeneralov-gw-02# show interfaces
  ethernet eth0 {
    address 2001:db8:c0de:11::1/64
    duplex auto
    hw-id 0c:95:f2:d6:00:00
    smp-affinity auto
    speed auto
  }
  ethernet eth1 {

```

Рис. 3.7: gns

Для того, чтобы это сделать, требуется установить более новую версию, например 1.3.3.

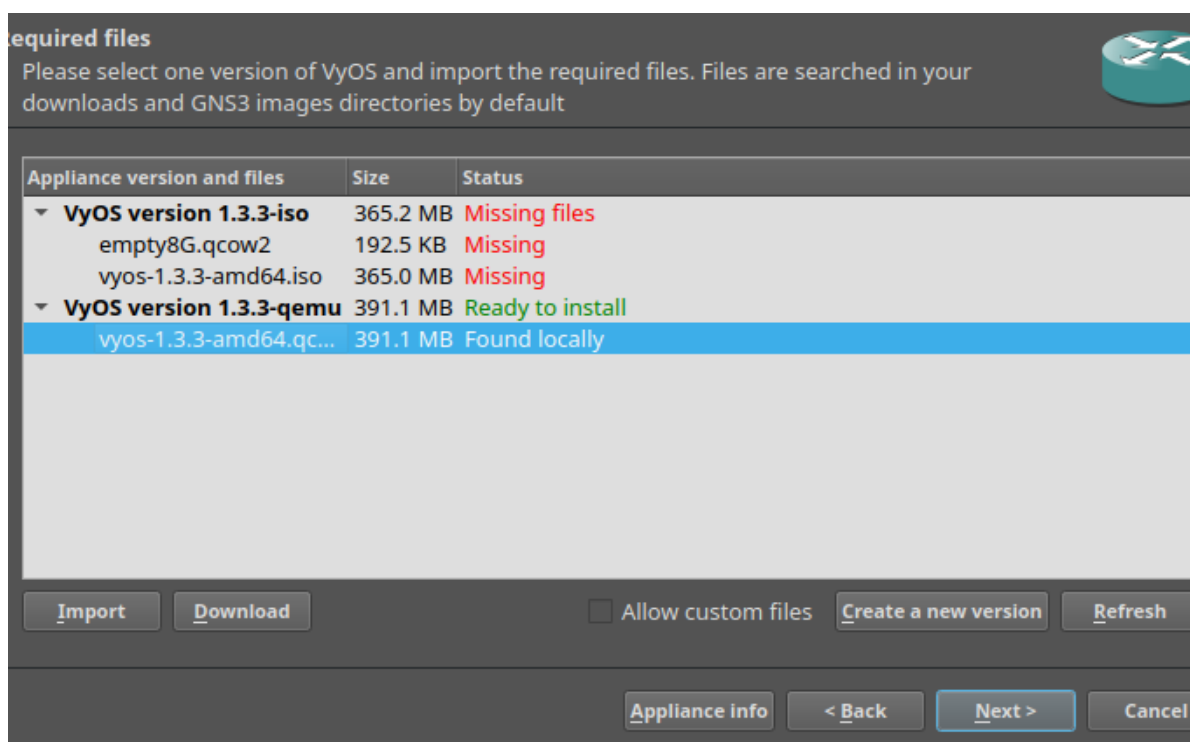


Рис. 3.8: gns

После этого нужно выполнить всю настройку на новом устройстве.

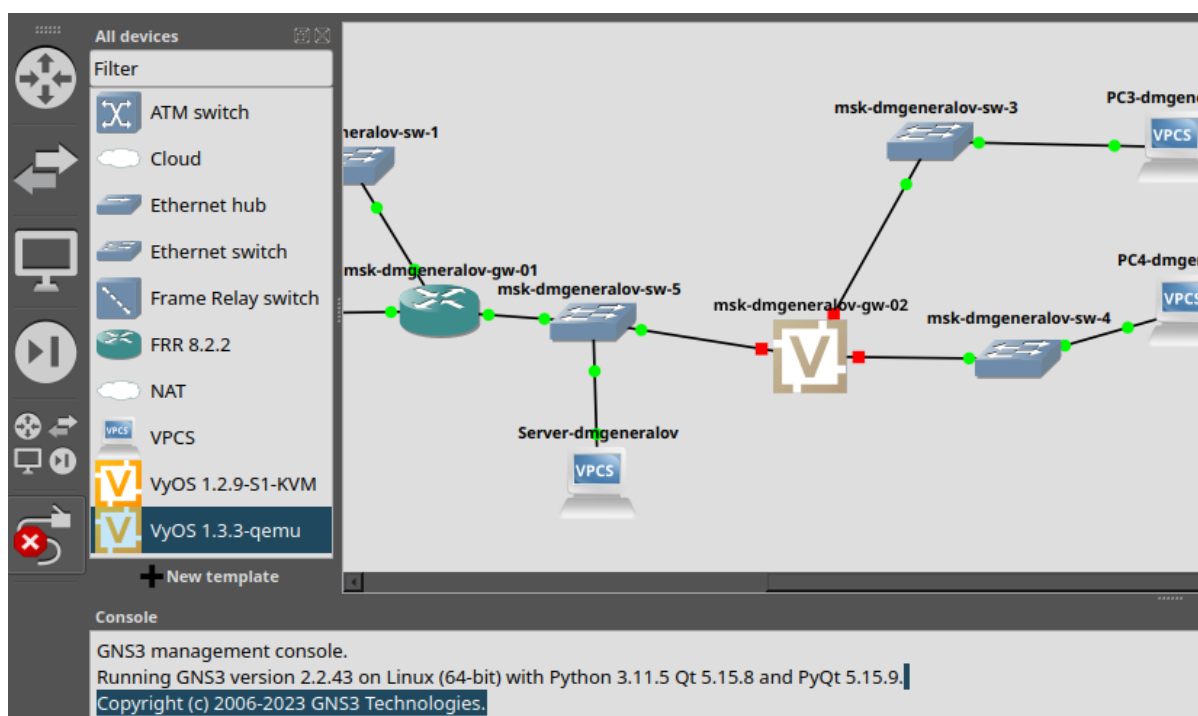


Рис. 3.9: gns

Теперь настройка router advertisements работает, и мы применяем её по указаниям.



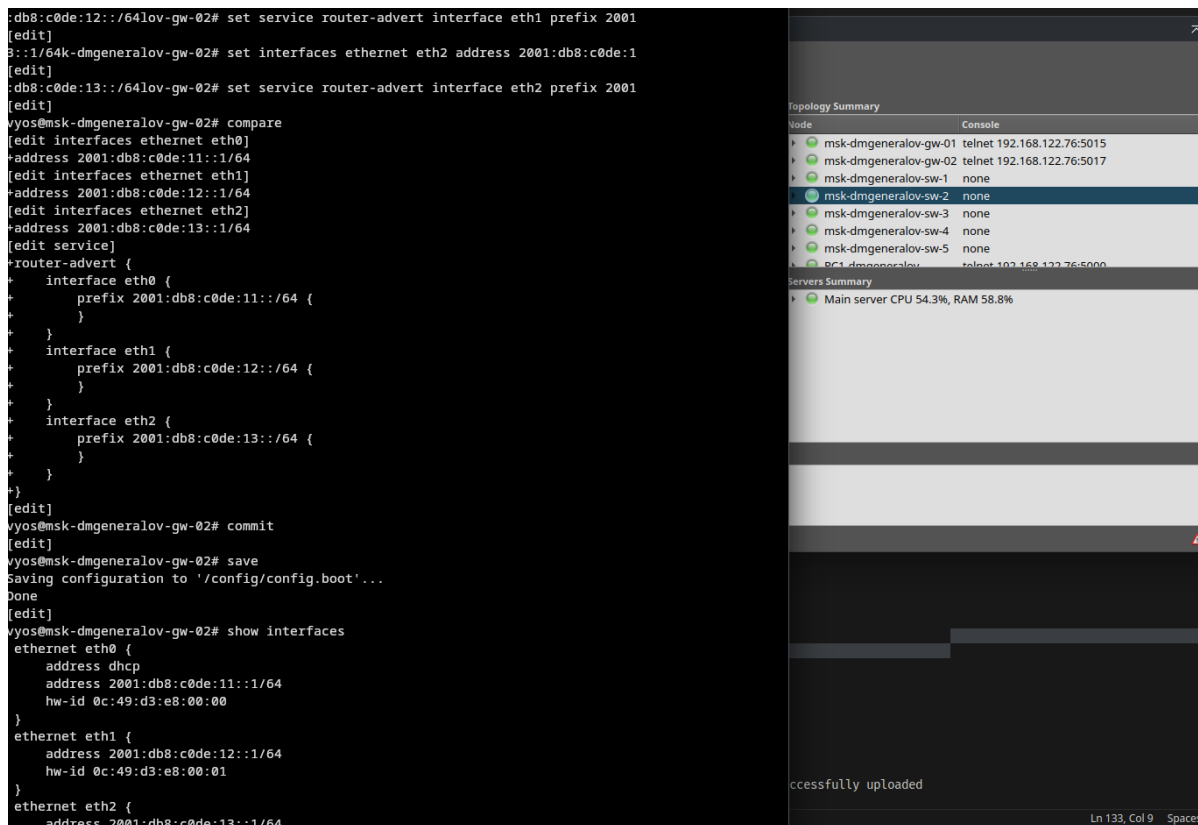


Рис. 3.10: gns

Можно теперь увидеть на отслеживаемом соединении, что роутер начал отправлять router advertisement, рекламируя свой маршрут. Также, либо роутер, либо сервер начали запрашивать DHCPv4-адрес (я не совсем понимаю, почему это происходит).

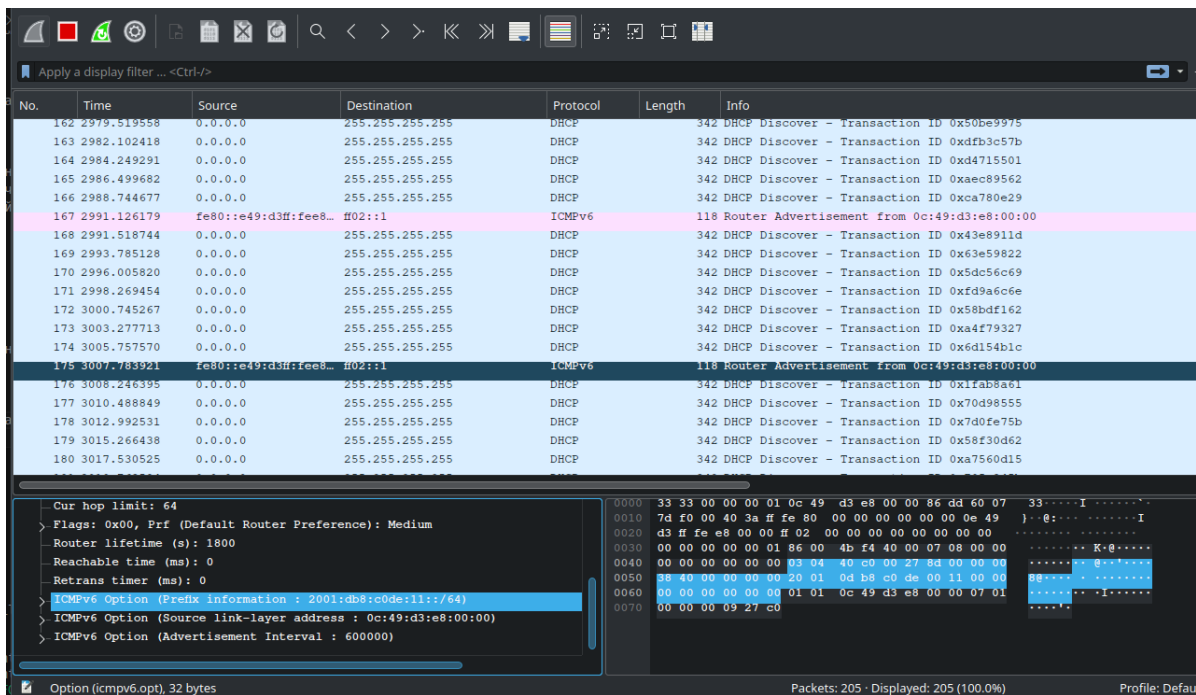


Рис. 3.11: wireshark

Теперь сеть полностью работоспособна: компьютеры в IPv4-сети могут связываться друг с другом, и компьютеры в IPv6-сети – друг с другом, и сервер может связываться с обоими, но одни не могут связываться с другими.

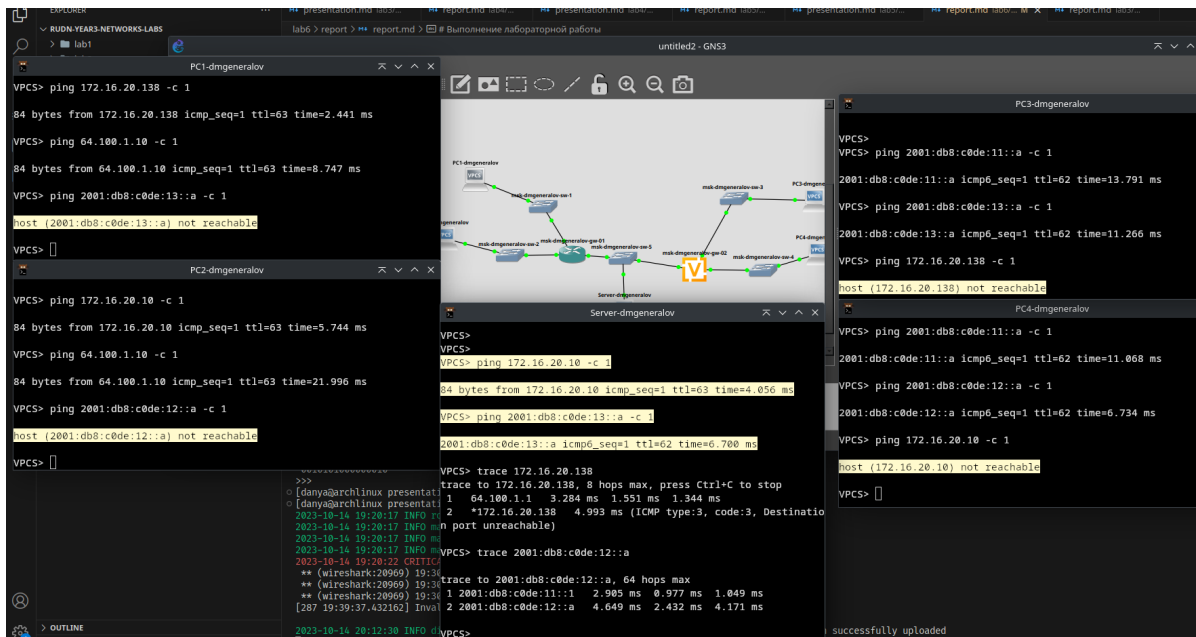


Рис. 3.12: gns

Все ping, которые взаимодействовали с сервером, можно увидеть на записи пакетов, которые направлены к серверу. На этом скриншоте видно, как, сначала, роутер отправляет серверу пакет ICMP – ping-запрос – затем сервер спрашивает про MAC-адрес роутера через ARP, получает ответ, затем отправляет ICMP-ответ на ping. После этого, через некоторое время, сервер сам отправляет ICMP-запрос и получает ответ, а затем ICMPv6-запрос и ответ: это когда я делал ping от сервера. Наконец, начинаются UDP-пакеты, которые получают ICMP-ответы, сообщающие о проблемах с TTL: это то, как работает программа traceroute. Ниже есть еще сообщения, как отправляются пакеты UDP по IPv6, и приходят ответы ICMPv6 – это тот же самый traceroute, но в IPv6-сеть. В каждом из этих пакетов будет содержаться IP-адреса связанных устройств и сетей, и из этого можно определить топологию всей сети.

No.	Time	Source	Destination	Protocol	Length	Info
249	3562.043526	172.16.20.10	64.100.1.10	ICMP	98	b \v \v16\017\020\021\022\023\024\025\026\027\030\031\032\033\0
250	3562.043834	64.100.1.10	172.16.20.10	ICMP	98	b \v \v16\017\020\021\022\023\024\025\026\027\030\031\032\033\0
251	3567.043129	0c:e3:52:d6:00:00	Private_66:68:04	ARP	60	Who has 64.100.1.10? Tell 64.100.1.1
252	3567.043310	Private_66:68:04	0c:e3:52:d6:00:00	ARP	60	64.100.1.10 is at 00:50:79:66:68:04
253	3570.379728	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x3d0c045f
254	3573.130060	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x3d0c045f
255	3576.817764	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x3d0c045f
256	3586.738571	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x3d0c045f
257	3593.689785	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x3d0c045f
258	3603.174103	64.100.1.10	172.16.20.10	ICMP	98	b \v \v16\017\020\021\022\023\024\025\026\027\030\031\032\033\0
259	3603.177942	172.16.20.10	64.100.1.10	ICMP	98	b \v \v16\017\020\021\022\023\024\025\026\027\030\031\032\033\0
260	3608.671589	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x3d0c045f
261	3613.490079	2001:db8:c0de:11::a	2001:db8:c0de:13::a	ICMPv6	118	0000001002003004005006a\b \v \v16\017\020\021\022\023\024
262	3613.494660	fe80::e49:d3ff:fe8::	fe02::1:000a:	ICMPv6	86	Neighbor Solicitation for 2001:db8:c0de:11::a from 0c:49:d3:fe:800:
263	3613.494894	2001:db8:c0de:11::a	fe80::e49:d3ff:fe80:	ICMPv6	86	Neighbor Advertisement 2001:db8:c0de:11::a (sol, ovr) is at 00:50:
264	3613.496577	2001:db8:c0de:13::a	2001:db8:c0de:11::a	ICMPv6	118	0000001002003004005006a\b \v \v16\017\020\021\022\023\024
265	3627.059137	64.100.1.10	172.16.20.138	UDP	106	000Pyrh004\016\017\020\021\022\023\024\025\026\027\030\031\032\0
266	3627.062119	64.100.1.10	64.100.1.10	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
267	3627.062321	64.100.1.10	172.16.20.138	UDP	106	000Pyrh004\016\017\020\021\022\023\024\025\026\027\030\031\032\0
268	3627.063726	64.100.1.10	64.100.1.10	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
269	3627.064535	64.100.1.10	172.16.20.138	UDP	106	000Pyrh004\016\017\020\021\022\023\024\025\026\027\030\031\032\0
270	3627.065725	64.100.1.10	64.100.1.10	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
271	3627.067514	64.100.1.10	172.16.20.138	UDP	106	000Pyrh004\016\017\020\021\022\023\024\025\026\027\030\031\032\0
272	3627.072250	172.16.20.138	64.100.1.10	ICMP	86	Destination unreachable (Port unreachable)
273	3627.073377	64.100.1.10	172.16.20.138	UDP	106	000Pyrh004\016\017\020\021\022\023\024\025\026\027\030\031\032\0
274	3627.076738	172.16.20.138	64.100.1.10	ICMP	86	Destination unreachable (Port unreachable)
275	3627.077936	64.100.1.10	172.16.20.138	UDP	106	000Pyrh004\016\017\020\021\022\023\024\025\026\027\030\031\032\0
276	3627.081353	172.16.20.138	64.100.1.10	ICMP	86	Destination unreachable (Port unreachable)
277	3630.004465	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x3d0c045f
278	3632.083018	0c:e3:52:d6:00:00	Private_66:68:04	ARP	60	Who has 64.100.1.10? Tell 64.100.1.1
279	3632.083322	Private_66:68:04	0c:e3:52:d6:00:00	ARP	60	64.100.1.10 is at 00:50:79:66:68:04
280	3636.530753	2001:db8:c0de:11::a	2001:db8:c0de:12::a	UDP	126	000Pyrh004\016\017\020\021\022\023\024\025\026\027\030\031\032\0
281	3636.533454	2001:db8:c0de:11::a	2001:db8:c0de:11::a	ICMPv6	174	Time Exceeded (hop limit exceeded in transit)
282	3636.534007	2001:db8:c0de:11::a	2001:db8:c0de:12::a	UDP	126	000Pyrh004\016\017\020\021\022\023\024\025\0

Рис. 3.13: wireshark

Наконец, требуется создать новую сеть, которая состоит из двух компьютеров и одного роутера на VyOS. Эти два компьютера будут находиться в разных подсетях IPv4 и IPv6: компьютер 1 – в подсети 10.10.1.96/27, где могут быть адреса от 10.10.1.97 до 10.10.1.127 включительно, и 2001:db8:1:1::/64, где могут быть адреса от 2001:db8:1:1:: до 2001:db8:1:1:ffff:ffff:ffff:ffff; а компьютер 2 – в подсети 10.10.1.16/28, где IP-адреса от 10.10.1.17 до 10.10.1.31, и 2001:DB8:1:4::/64, где от 2001:DB8:1:4:: до 2001:DB8:1:4:ffff:ffff:ffff:ffff.

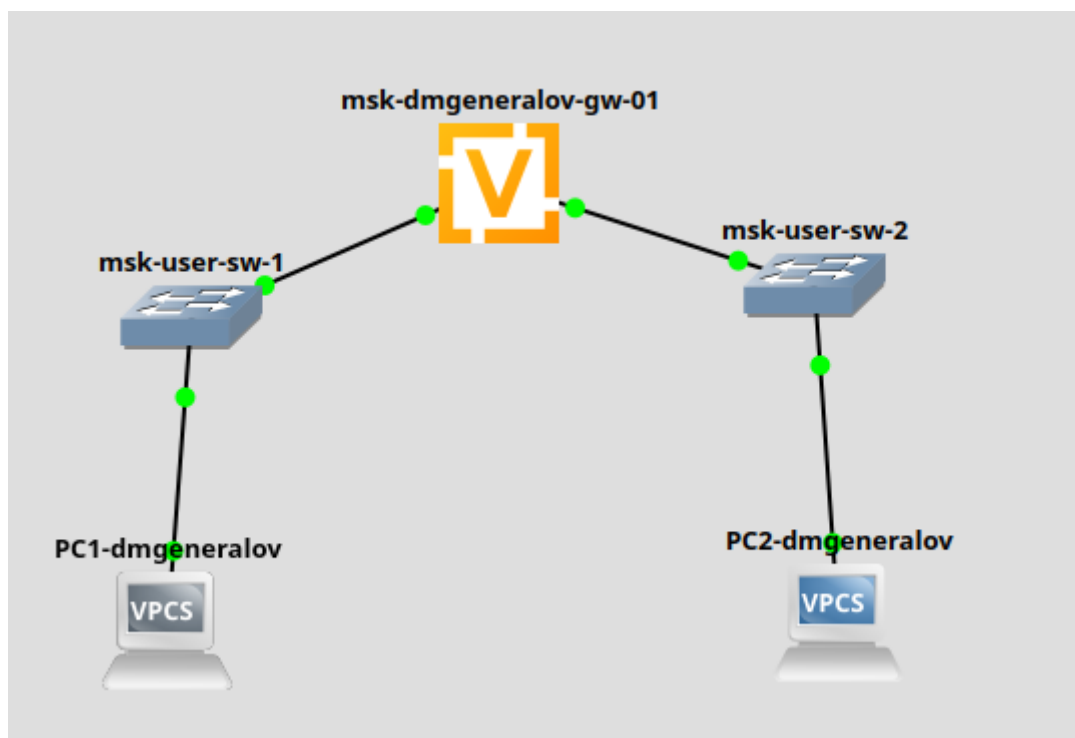


Рис. 3.14: gns

Традиционно роутер имеет наименьший адрес в подсети, поэтому на левой стороне он будет иметь адрес 10.10.1.97, а на правой – 10.10.1.17. Локальная часть IPv6-адреса у него в обоих случаях будет заканчиваться на единицу. Компьютеры будут иметь следующий по порядку IP-адрес (10.10.1.98 и 10.10.1.18 и заканчивающийся на двойку).

<pre> Executing the startup file Hostname is too long. (Maximum 12 characters) VPCS&gt; ip 10.10.1.98/27 10.10.1.97 Checking for duplicate address... VPCS : 10.10.1.98 255.255.255.224 gateway 10.10.1.97  VPCS&gt; ip 2001:db8:1:1::1/64 PC1 : 2001:db8:1:1::1/64  VPCS&gt; save Saving startup configuration to startup.vpc . done  VPCS&gt; ip 2001:db8:1:1::2/64 PC1 : 2001:db8:1:1::2/64  VPCS&gt; save Saving startup configuration to startup.vpc . done  VPCS&gt; </pre>	<pre> unted on lev run ys/firmware/efi/efivars  lev/shm imp root root/efi run/user/1000 </pre>	<pre> Hostname is too long. (Maximum 12 characters) VPCS&gt; ip 10.10.1.18/28 10.10.1.17 Checking for duplicate address... VPCS : 10.10.1.18 255.255.255.240 gateway 10.10.1.17  VPCS&gt; ip 2001:db8:1:4::1/64 PC1 : 2001:db8:1:4::1/64  VPCS&gt; save Saving startup configuration to startup.vpc . done  VPCS&gt; ip 2001:db8:1:4::2/64 PC1 : 2001:db8:1:4::2/64  VPCS&gt; VPCS&gt; save Saving startup configuration to startup.vpc . done  VPCS&gt; </pre>
---	--	---

Рис. 3.15: gns

Роутер затем настраивается с router advertisement и IPv4-адресами.

```
ryos@msk-dmgeneralov-gw-01# show
interfaces {
    ethernet eth0 {
        address 10.10.1.97/27
        hw-id 0c:42:3e:4d:00:00
    }
    ethernet eth1 {
        address 10.10.1.17/28
        address 2001:db8:1:4::1/64
        hw-id 0c:42:3e:4d:00:01
    }
    ethernet eth2 {
        hw-id 0c:42:3e:4d:00:02
    }
    loopback lo {
    }
}
service {
    router-advert {
        interface eth0 {
            prefix 2001:db8:1:1::/64 {
            }
        }
        interface eth1 {
            prefix 2001:db8:1:4::/64 {
            }
        }
    }
}
```

Рис. 3.16: gns

После этого, ping работает в обе стороны, по обоим протоколам.

```

VPCS>
VPCS>
VPCS> ping 10.10.1.18
84 bytes from 10.10.1.18 icmp_seq=1 ttl=63 time=2.615 ms
^C
VPCS> ping 2001:db8:1:4::2 -c 1
2001:db8:1:4::2 icmp6_seq=1 ttl=62 time=13.158 ms
^C
VPCS> trace 10.10.1.18
trace to 10.10.1.18, 8 hops max, press Ctrl+C to stop
 1 10.10.1.97 1.776 ms 1.316 ms 0.938 ms
 2 *10.10.1.18 4.863 ms (ICMP type:3, code:3, Destination port unreachable)
VPCS> trace 2001:db8:1:4::2
trace to 2001:db8:1:4::2, 64 hops max
 1 2001:db8:1:1::1 2.286 ms 3.377 ms 3.143 ms
 2 2001:db8:1:4::2 5.051 ms 3.990 ms 2.538 ms
VPCS>

```

```

VPCS>
VPCS>
VPCS> ping 10.10.1.98 -c 1
84 bytes from 10.10.1.98 icmp_seq=1 ttl=63 time=7.391 ms
VPCS> ping 2001:db8:1:1::2 -c 1
2001:db8:1:1::2 icmp6_seq=1 ttl=62 time=8.695 ms
VPCS> trace 10.10.1.98
trace to 10.10.1.98, 8 hops max, press Ctrl+C to stop
 1 10.10.1.17 2.056 ms 0.905 ms 1.005 ms
 2 *10.10.1.98 1.684 ms (ICMP type:3, code:3, Destination port unreachable)
VPCS> trace 2001:db8:1:1::2
trace to 2001:db8:1:1::2, 64 hops max
 1 2001:db8:1:4::1 2.137 ms 0.851 ms 0.744 ms
 2 2001:db8:1:1::2 1.987 ms 2.452 ms 2.890 ms
VPCS>

```

Рис. 3.17: gns

## 4 Выводы

Я получил опыт настройки сетей с протоколом IPv4 и IPv6.