

Министерство цифрового развития, связи и
массовых коммуникаций Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего
образования «Сибирский государственный университет телекоммуникаций и
информатики» (СибГУТИ)

Кафедра вычислительных систем

ОТЧЕТ
по практической работе 4

по дисциплине «Сети ЭВМ и телекоммуникации»

Выполнил:
студент гр. ИС-142
«__» июня 2023 г.

/Григорьев Ю.В./

Проверил:
«__» июня 2023 г.

/Перышкова Е.Н./

Оценка « _____ »

Новосибирск 2023

ОГЛАВЛЕНИЕ

ПОСТАНОВКА ЗАДАЧИ	3
ВЫПОЛНЕНИЕ РАБОТЫ	5

ПОСТАНОВКА ЗАДАЧИ

1. Соберите конфигурацию сети, представленной на рисунке 1. Коммутаторы на рисунке – это виртуальные коммутаторы VirtualBox, работающие в режиме Host-only network.

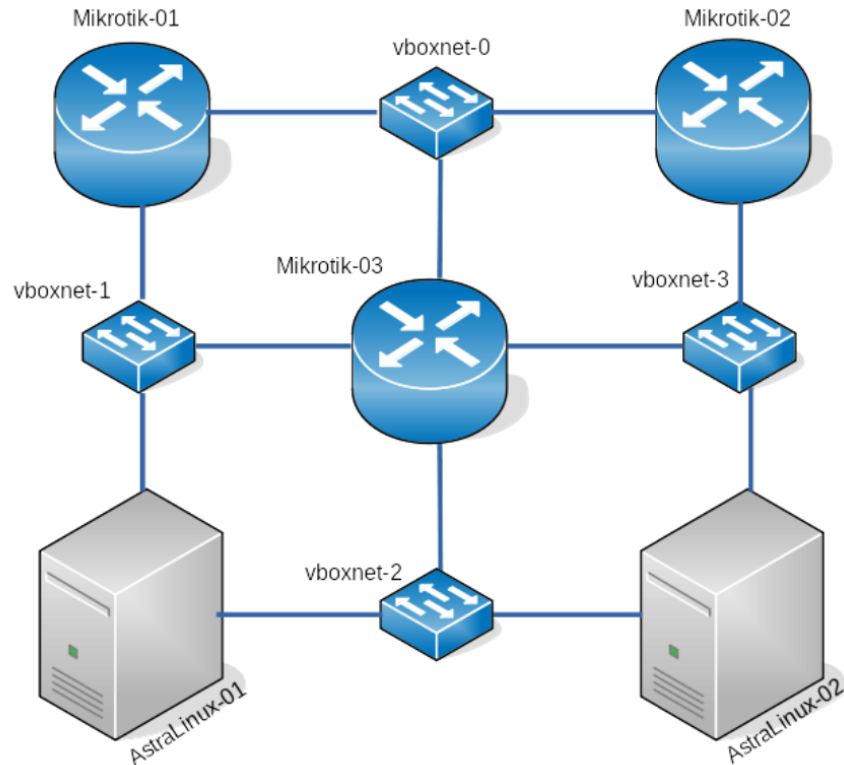


Рисунок 1 – Конфигурация сети для практического занятия

2. Вам представлена подсеть 10.10.N.0/24, где N — это Ваш порядковый номер в списке журнала преподавателя. Разделите полученный диапазон адресов на 4 равные подсети. Определите какой из полученных диапазонов будет использоваться в какой сети. Настройте все сетевые интерфейсы в соответствии с выбранной схемой адресации.

3. Используя статическую маршрутизацию настройте передачу пакетов таким образом, чтобы они в предложенной конфигурации передавали между сетями по часовой стрелке (т. е. Astralinux-01 → Vboxnet1 → mikrotik-01 → vboxnet0 → mikrotik-02 → vboxnet3 → Astralinux-02 → vboxnet2 → Astralinux-01). Проверьте, что любой узел пингует любой адрес из назначенных в сети.

4. Удалите всю конфигурацию статической маршрутизации. Настройте на маршрутизаторах Mikrotik динамическую маршрутизацию по протоколу RIP. Покажите информация о каких сетях стала известна маршрутизаторам? С использованием пакетного сниффера Wireshark покажите содержимое

пакетов, распространяемых по сети по протоколу RIP. Покажите, как в полученной конфигурации сети работает отказоустойчивость сети.

5. Удалите всю конфигурацию динамической маршрутизации по протоколу RIP. Настройте на маршрутизаторах Mikrotik динамическую маршрутизацию по протоколу OSPFv2. Покажите информация о каких сетях стала известна маршрутизаторам? С использованием пакетного сниффера Wireshark покажите содержимое пакетов, распространяемых по сети по протоколу OSPF. Покажите, как в полученной конфигурации сети работает отказоустойчивость сети.

6. Вам выделен префикс IPv6 fd00:{YEAR}:{MONTH}::/64, где YEAR — год Вашего рождения, MONTH — месяц Вашего рождения. Определите префиксы для 4 подсетей. Настройте интерфейсы маршрутизаторов mikrotik так, чтобы они распространяли префиксы соответствующих подсетей.

7. Настройте на маршрутизаторах Mikrotik динамическую маршрутизацию по протоколу OSPFv3. Покажите информация о каких сетях стала известна маршрутизаторам? С использованием пакетного сниффера Wireshark покажите содержимое пакетов, распространяемых по сети по протоколу OSPF. Покажите, как в полученной конфигурации сети работает отказоустойчивость сети.

ВЫПОЛНЕНИЕ РАБОТЫ

При выполнении работы было сделано следующее:

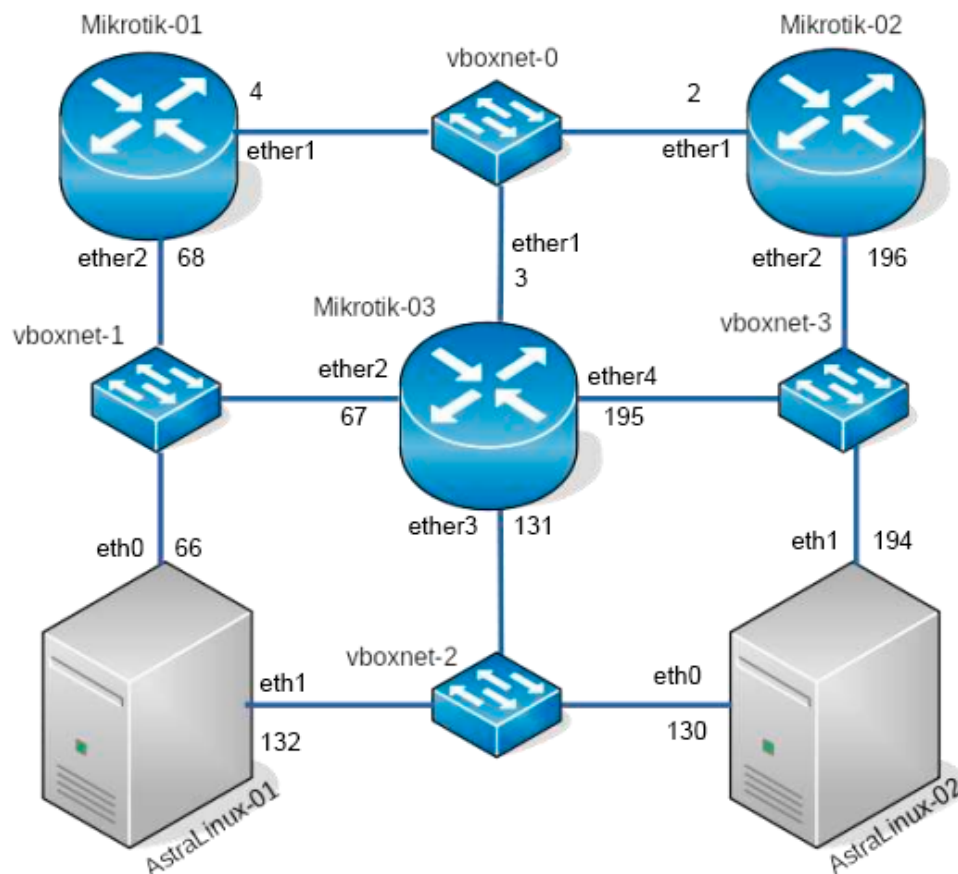
1. Созданы и сконфигурированы виртуальные машины `astra1`, `astra2`, `router1`, `router2`, `router3` и виртуальные адаптеры сети по схеме задания.

2. Мной были выделены 4 подсети из диапазона `10.10.3.0/24`, зафиксировав 2 первых бита четвёртого октета адреса IPv4:

`vboxnet0`: `10.10.3.0` - `10.10.3.63`
`vboxnet1`: `10.10.3.64` - `10.10.3.127`
`vboxnet2`: `10.10.3.128` - `10.10.3.191`
`vboxnet3`: `10.10.3.192` - `10.10.3.255`

Чтобы данные подсети имели свои собственные адреса сети (т.е. `vboxnet0` соответствует network `10.10.3.0`, `vboxnet1` - `10.10.3.64` и т.д.), маска намеренно изменена на `/26` - добавленные 2 бита отвечают за подсети, описанные ранее.

Конфигурация с изображёнными интерфейсами и последними октетами IPv4-адресов



Демонстрация настроенных статических IPv4-адресов:

router1			router2		
[admin@mt-01] > ip address print			[admin@mt-02] > ip address print		
Columns: ADDRESS, NETWORK, INTERFACE			Columns: ADDRESS, NETWORK, INTERFACE		
#	ADDRESS	NETWORK	#	ADDRESS	NETWORK
0	10.10.3.4/26	10.10.3.0	0	10.10.3.2/26	10.10.3.0
1	10.10.3.68/26	10.10.3.64	1	10.10.3.196/26	10.10.3.192
		ether1			ether1
		ether2			ether2

router3

```
[admin@mt-03] > ip address print
Columns: ADDRESS, NETWORK, INTERFACE
# ADDRESS      NETWORK      INTERFACE
0 10.10.3.3/26   10.10.3.0    ether1
1 10.10.3.67/26  10.10.3.64   ether2
2 10.10.3.131/26 10.10.3.128  ether3
3 10.10.3.195/26 10.10.3.192  ether4
```

astral

```
eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
link/ether 08:00:27:f1:47:41 brd ff:ff:ff:ff:ff:ff
inet 10.10.3.66/26 brd 10.10.3.127 scope global eth0
    valid_lft forever preferred_lft forever
inet6 fe80::a00:27ff:fef1:4741/64 scope link
    valid_lft forever preferred_lft forever
eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
link/ether 08:00:27:f9:d1:28 brd ff:ff:ff:ff:ff:ff
inet 10.10.3.132/26 brd 10.10.3.191 scope global eth1
```

astra2

```
eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
link/ether 08:00:27:72:06:7d brd ff:ff:ff:ff:ff:ff
inet 10.10.3.130/26 brd 10.10.3.191 scope global eth0
    valid_lft forever preferred_lft forever
inet6 fe80::a00:27ff:fe72:67d/64 scope link
    valid_lft forever preferred_lft forever
eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
link/ether 08:00:27:fd:ed:19 brd ff:ff:ff:ff:ff:ff
inet 10.10.3.194/26 brd 10.10.3.255 scope global eth1
```

3. Для настройки маршрутизации посмотрим, что уже есть в таблицах маршрутизации:

```
owner@astra1:~$ sudo -i
root@astra1:~# route -n
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
10.10.3.64       0.0.0.0         255.255.255.192 U        0      0      0 eth0
10.10.3.128      0.0.0.0         255.255.255.192 U        0      0      0 eth1
```

```
[admin@mt-01] > ip route print
Flags: D - DYNAMIC; A - ACTIVE; c, y - COPY
Columns: DST-ADDRESS, GATEWAY, DISTANCE
DAc 10.10.3.0/26   ether1         0
DAc 10.10.3.64/26 ether2         0
[admin@mt-03] > ip route print
Flags: D - DYNAMIC; A - ACTIVE; c, y - COPY
Columns: DST-ADDRESS, GATEWAY, DISTANCE
DAc 10.10.3.0/26   ether1         0
DAc 10.10.3.64/26 ether2         0
DAc 10.10.3.128/26 ether3         0
DAc 10.10.3.192/26 ether4         0
```

На виртуальных машинах с ОС AstraLinux в файле /etc/sysctl.conf убираем комментарий со строки включения перенаправления пакетов переназначением переменной ядра net.ipv4.ip_forward на значение 1. Также меняем значение переменной

net.ipv4.conf.*.rp_filter на 2, чтобы принимать пакеты со всех интерфейсов устройства, а не только с того, с которого мы бы отправили его обратно (reverse path).

Командой `sysctl -p` применяем настройки этой конфигурации для нынешней сессии.

Также включим захват пакетов на всех машинах командами “`vboxmanage modifyvm <vmname> --nictrace1 on --nictracefile1 <path>`”, чтобы не искать пакеты в live-режиме.

```
C:\Program Files\Oracle\VirtualBox>VBoxManage.exe modifyvm router1 --nictrace1 on --nictracefile1
C:\Users\Yuriy\Documents\router1-1.pcap

C:\Program Files\Oracle\VirtualBox>VBoxManage.exe modifyvm router1 --nictrace2 on --nictracefile2
C:\Users\Yuriy\Documents\router1-2.pcap

C:\Program Files\Oracle\VirtualBox>VBoxManage.exe modifyvm router2 --nictrace1 on --nictracefile1
C:\Users\Yuriy\Documents\router2-1.pcap

C:\Program Files\Oracle\VirtualBox>VBoxManage.exe modifyvm router2 --nictrace2 on --nictracefile2
C:\Users\Yuriy\Documents\router2-2.pcap
```

```
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
```

```
root@astra2:~# sysctl -p
net.ipv4.conf.default.rp_filter = 2
net.ipv4.conf.all.rp_filter = 2
net.ipv4.ip_forward = 1
```

Настроим таблицы маршрутизации у каждого устройства в сети, чтобы пакеты шли по часовой стрелке командой `route add -net <net_address>/26 gw <gateway_address>` в машинах `astralinux` и командой `ip route add dst-address=<net_address>/26 gateway=<gateway_address>` на роутерах MikroTik.

```
root@astra1:~# route add -net 10.10.3.0/26 gw 10.10.3.68
root@astra1:~# route add -net 10.10.3.192/26 gw 10.10.3.68
root@astra1:~# route -n
Kernel IP routing table
Destination    Gateway         Genmask         Flags Metric Ref    Use Iface
10.10.3.0      10.10.3.68     255.255.255.192 UG      0      0      0 eth0
10.10.3.64     0.0.0.0        255.255.255.192 U        0      0      0 eth0
10.10.3.128    0.0.0.0        255.255.255.192 U        0      0      0 eth1
10.10.3.192    10.10.3.68     255.255.255.192 UG      0      0      0 eth0
```

```
root@astra2:~# route add -net 10.10.3.0/26 gw 10.10.3.132
root@astra2:~# route add -net 10.10.3.64/26 gw 10.10.3.132
root@astra2:~# route -n
Kernel IP routing table
Destination    Gateway         Genmask         Flags Metric Ref    Use Iface
10.10.3.0      10.10.3.132    255.255.255.192 UG      0      0      0 eth0
10.10.3.64     10.10.3.132    255.255.255.192 UG      0      0      0 eth0
10.10.3.128    0.0.0.0        255.255.255.192 U        0      0      0 eth0
10.10.3.192    0.0.0.0        255.255.255.192 U        0      0      0 eth1
```

Результат:

router1

```
[admin@mt-01] > ip route print
Flags: D - DYNAMIC; A - ACTIVE; c, s, y - COPY
Columns: DST-ADDRESS, GATEWAY, DISTANCE
#      DST-ADDRESS      GATEWAY      DISTANCE
  DAc  10.10.3.0/26      ether1        0
  DAc  10.10.3.64/26      ether2        0
0  As  10.10.3.128/26     10.10.3.2     1
1  As  10.10.3.192/26     10.10.3.2     1
```

router2

```
[admin@mt-02] > ip route print
Flags: D - DYNAMIC; A - ACTIVE; c, s, y - COPY
Columns: DST-ADDRESS, GATEWAY, DISTANCE
#      DST-ADDRESS      GATEWAY      DISTANCE
  DAc  10.10.3.0/26      ether1        0
0  As  10.10.3.64/26      10.10.3.194   1
1  As  10.10.3.128/26     10.10.3.194   1
  DAc  10.10.3.192/26     ether2        0
```

Так как router3 одновременно подключен ко всем подсетям, его пакеты будут идти сразу же в нужную сеть и приниматься из любой сети на предназначенный для неё интерфейс. Его таблица маршрутизации в дополнении не нуждается.

```
[admin@mt-03] > ip route print
Flags: D - DYNAMIC; A - ACTIVE; c, y - COPY
Columns: DST-ADDRESS, GATEWAY, DISTANCE
  DAc  10.10.3.0/26      ether1        0
  DAc  10.10.3.64/26      ether2        0
  DAc  10.10.3.128/26     ether3        0
  DAc  10.10.3.192/26     ether4        0
```

astral

```
root@astra1:~# route -n
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
10.10.3.0        10.10.3.68      255.255.255.192 UG      0      0      0 eth0
10.10.3.64       0.0.0.0         255.255.255.192 U        0      0      0 eth0
10.10.3.128      0.0.0.0         255.255.255.192 U        0      0      0 eth1
10.10.3.192      10.10.3.68      255.255.255.192 UG      0      0      0 eth0
```

astra2

```
root@astra2:~# route -n
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
10.10.3.0        10.10.3.132     255.255.255.192 UG      0      0      0 eth0
10.10.3.64       10.10.3.132     255.255.255.192 UG      0      0      0 eth0
10.10.3.128      0.0.0.0         255.255.255.192 U        0      0      0 eth0
10.10.3.192      0.0.0.0         255.255.255.192 U        0      0      0 eth1
```

Чтобы проверить связанность всех устройств, запустим ping с машины astral до адреса 10.10.3.196 (интерфейс ether2 машины router2). Это заставит наш пакет пройти “полный круг” устройств по часовой стрелке. Всё работает!

```
root@astra1:~# ping 10.10.3.196
PING 10.10.3.196 (10.10.3.196) 56(84) bytes of data.
64 bytes from 10.10.3.196: icmp_seq=1 ttl=63 time=1.30 ms
64 bytes from 10.10.3.196: icmp_seq=2 ttl=63 time=1.14 ms
```


Параллельно запустим Wireshark и проанализируем отправляемые пакеты по всем устройствам. В данном случае в промежуточных узлах router1, router2 намеренно отслеживались только интерфейсы, перенаправляющие пакеты echo request, а в astra2 - перенаправляющий пакет echo reply (с целью экономии памяти на диске).

От eth0 astra1 к ether2 router1 (начало):

140	613.151371	PcsCompu_f1:47:41	Broadcast	ARP	60 Who has 10.10.3.68? Tell 10.10.3.66
141	613.151733	PcsCompu_be:f2:d5	PcsCompu_f1:47:41	ARP	60 10.10.3.68 is at 08:00:27:be:f2:d5
142	613.151972	10.10.3.66	10.10.3.196	ICMP	98 Echo (ping) request id=0x0269, seq=1/256,
143	614.181386	10.10.3.66	10.10.3.196	ICMP	98 Echo (ping) request id=0x0269, seq=2/512,
144	615.204964	10.10.3.66	10.10.3.196	ICMP	98 Echo (ping) request id=0x0269, seq=3/768,

От ether1 router1 к ether1 router2:

191	633.133472	PcsCompu_1f:9c:5f	Broadcast	ARP	60 Who has 10.10.3.2? Tell 10.10.3.4
192	633.133913	PcsCompu_1c:ba:6b	PcsCompu_1f:9c:5f	ARP	60 10.10.3.2 is at 08:00:27:1c:ba:6b
193	633.134191	10.10.3.66	10.10.3.196	ICMP	98 Echo (ping) request id=0x0269, seq=1/256,
194	634.162900	10.10.3.66	10.10.3.196	ICMP	98 Echo (ping) request id=0x0269, seq=2/512,
195	635.186469	10.10.3.66	10.10.3.196	ICMP	98 Echo (ping) request id=0x0269, seq=3/768,

От ether2 router2 к eth1 astra2:

145	618.712626	PcsCompu_37:30:03	Broadcast	ARP	60 Who has 10.10.3.194? Tell 10.10.3.196
146	618.712905	PcsCompu_fd:ed:19	PcsCompu_37:30:03	ARP	60 10.10.3.194 is at 08:00:27:fd:ed:19
147	618.713188	10.10.3.196	10.10.3.66	ICMP	98 Echo (ping) reply id=0x0269, seq=1/256, ttl=64
148	619.741264	10.10.3.196	10.10.3.66	ICMP	98 Echo (ping) reply id=0x0269, seq=2/512, ttl=64
149	620.764850	10.10.3.196	10.10.3.66	ICMP	98 Echo (ping) reply id=0x0269, seq=3/768, ttl=64

От eth0 astra2 к eth1 astra1 (конец):

28	128.732285	PcsCompu_72:06:7d	Broadcast	ARP	60 Who has 10.10.3.132? Tell 10.10.3.130
29	128.732584	PcsCompu_f9:d1:28	PcsCompu_72:06:7d	ARP	60 10.10.3.132 is at 08:00:27:f9:d1:28
30	128.732667	10.10.3.196	10.10.3.66	ICMP	98 Echo (ping) reply id=0x02bb, seq=1/256, ttl=63
31	129.733572	10.10.3.196	10.10.3.66	ICMP	98 Echo (ping) reply id=0x02bb, seq=2/512, ttl=63
32	130.735502	10.10.3.196	10.10.3.66	ICMP	98 Echo (ping) reply id=0x02bb, seq=3/768, ttl=63

Полученные пакеты на eth1 astra1:

172	1035.921714	PcsCompu_72:06:7d	Broadcast	ARP	60 Who has 10.10.3.132? Tell 10.10.3.130
173	1035.921949	PcsCompu_f9:d1:28	PcsCompu_72:06:7d	ARP	60 10.10.3.132 is at 08:00:27:f9:d1:28
174	1035.922077	10.10.3.196	10.10.3.66	ICMP	98 Echo (ping) reply id=0x02bb, seq=1/256, ttl=63
175	1036.923001	10.10.3.196	10.10.3.66	ICMP	98 Echo (ping) reply id=0x02bb, seq=2/512, ttl=63
176	1037.924921	10.10.3.196	10.10.3.66	ICMP	98 Echo (ping) reply id=0x02bb, seq=3/768, ttl=63

Удалим добавленные нами из таблиц маршрутизации IPv4, больше они не пригодятся.

4. Настроим динамическую маршрутизацию по протоколу RIP.

Зайдём в WebFig всех роутеров и добавим новый RIP-instance и interface template (для всех интерфейсов устройства) в меню Routing. В параметре Redistribute укажем connected, static и rip для получения информации о напрямую подключенных устройствах (маршрутах), статических записей в таблицах маршрутизации и записей, полученных другим устройством также через протокол RIP.

Update Interval установлю в значение 10 секунд, чтобы не ждать долго пакетов RIP Response. Смотрим таблицы маршрутизации на роутерах: появились абсолютно все маршруты до всех устройств в сети с выставленными метриками (расстояниями).

Проверим получаемые/отправляемые пакеты через Wireshark на примере router1: видим пакет RIPv2 Request и далее поочередно получаем таблицы маршрутизации соседних роутеров. Динамическая маршрутизация создана успешно.

Настройка динамической маршрутизации IPv4 по протоколу RIPv2 в WebFig'e MikroTik.

Enabled ☒

Name

VRF

▼

AFI

▼

Input Filter

▼

Output Filter

▼

Select Output Filter

▼

☒connected

☒static

☒rip

☐ospf

☐bgp

☐vpn

☐dhcp

☐fantasy

☐modem

☐copy

Originate Default

▼

Routing Table

▼

Route Timeout

▼

Route GC Timeout

▼

Update Interval

▲

RouterOS v7.8 (stable)

OK Cancel Apply Remove

Enabled ☒

Name

Instance

rip-instance-1 ▼

Interfaces

all ▼ ▲

Source Addresses

▼

Обновленные таблицы маршрутизации с помощью протокола RIPv2:

router1

```
[admin@mt-01] > ip route print
Flags: D - DYNAMIC; A - ACTIVE; c, r, y - COPY; + - ECMP
Columns: DST-ADDRESS, GATEWAY, DISTANCE
  DST-ADDRESS  GATEWAY      DISTANCE
D r  10.10.3.0/26  10.10.3.67%ether2  120
DAc  10.10.3.0/26  ether1         0
D r  10.10.3.64/26  10.10.3.3%ether1  120
DAc  10.10.3.64/26  ether2         0
DAr+ 10.10.3.128/26  10.10.3.67%ether2  120
DAr+ 10.10.3.128/26  10.10.3.3%ether1  120
DAr+ 10.10.3.192/26  10.10.3.67%ether2  120
DAr+ 10.10.3.192/26  10.10.3.2%ether1  120
DAr+ 10.10.3.192/26  10.10.3.3%ether1  120
```

router2

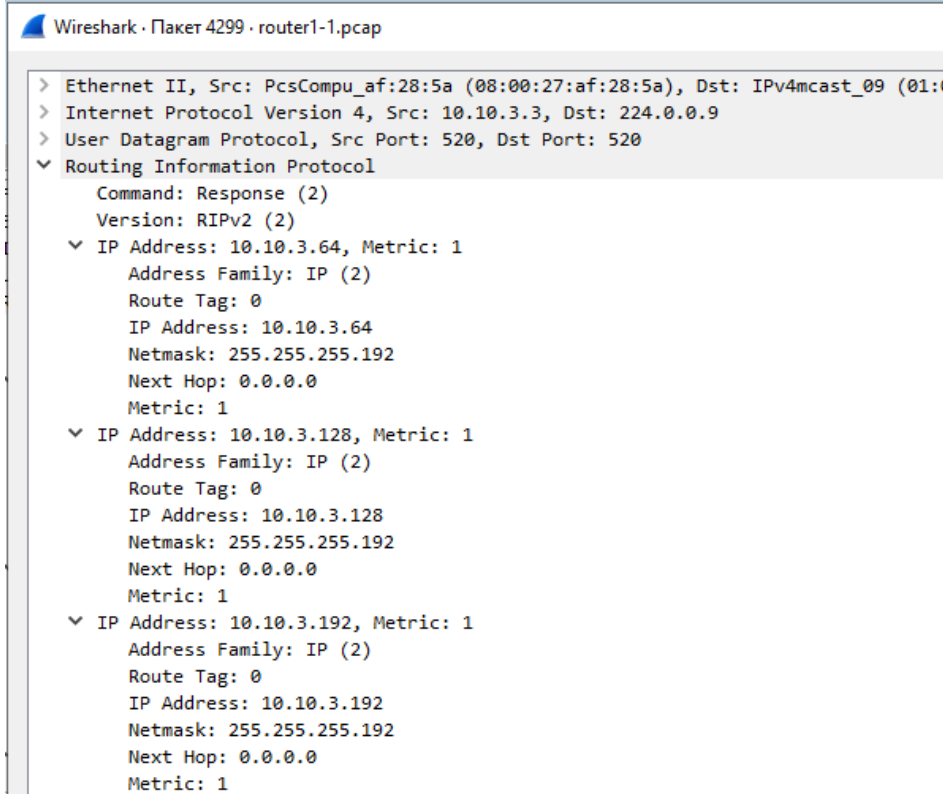
```
[admin@mt-02] > ip route print
Flags: D - DYNAMIC; A - ACTIVE; c, r, y - COPY; + - ECMP
Columns: DST-ADDRESS, GATEWAY, DISTANCE
  DST-ADDRESS  GATEWAY      DISTANCE
D r  10.10.3.0/26  10.10.3.195%ether2  120
DAc  10.10.3.0/26  ether1         0
DAr+ 10.10.3.64/26  10.10.3.4%ether1  120
DAr+ 10.10.3.64/26  10.10.3.195%ether2  120
DAr+ 10.10.3.64/26  10.10.3.3%ether1  120
DAr+ 10.10.3.128/26  10.10.3.195%ether2  120
DAr+ 10.10.3.128/26  10.10.3.3%ether1  120
D r  10.10.3.192/26  10.10.3.3%ether1  120
DAc  10.10.3.192/26  ether2         0
```

router3

```
[admin@mt-03] > ip route print
Flags: D - DYNAMIC; A - ACTIVE; c, r, y - COPY
Columns: DST-ADDRESS, GATEWAY, DISTANCE
DST-ADDRESS    GATEWAY        DISTANCE
D r 10.10.3.0/26 10.10.3.68%ether2 120
D r 10.10.3.0/26 10.10.3.196%ether4 120
DAc 10.10.3.0/26 ether1          0
D r 10.10.3.64/26 10.10.3.4%ether1 120
DAc 10.10.3.64/26 ether2          0
D r 10.10.3.128/26 10.10.3.4%ether1 120
D r 10.10.3.128/26 10.10.3.2%ether1 120
DAc 10.10.3.128/26 ether3          0
D r 10.10.3.192/26 10.10.3.2%ether1 120
DAc 10.10.3.192/26 ether4          0
```

Демонстрация пакетов RIPv2 с eth0 router1 и содержания одного из них (в данном случае это таблица маршрутизации, которой поделился router3):

4280	1771.256768	10.10.3.4	224.0.0.9	RIPv2	66 Request
4285	1771.263339	10.10.3.3	10.10.3.4	RIPv2	106 Response
4286	1771.263692	10.10.3.2	10.10.3.4	RIPv2	106 Response
4288	1771.272035	10.10.3.4	224.0.0.9	RIPv2	86 Response
4298	1772.803755	10.10.3.2	224.0.0.9	RIPv2	106 Response
4299	1779.556263	10.10.3.3	224.0.0.9	RIPv2	106 Response



Демонстрация отказоустойчивости сети:

Попробуем пропинговать адрес 10.10.3.196 (ether2 router2) с машины astra1. В поле маршрутизации к сети 10.10.3.192 указан ether2 router1 (10.10.3.68).

Проверяем командой “**tracert -n**” маршрут следования нашего пакета.

Пока все интерфейсы работают, пинг идёт по маршруту astra1->router1->router2.

Теперь пробуем отключить интерфейс ether1 у router1, через который идёт прямое подключение к router2. Видим: соединение потеряно и router1 не знает, куда отправить пакет, потому что информация о маршрутах в RIP ещё не обновилась.

Включаем-выключаем ether2 у router1, чтобы принудительно обновить конфигурацию RIPv2 пакетом Request. Проверяем путь пакета - всё снова работает, но в этот раз router1 пересылает пакет на router3, который далее отправляет его получателю - ether2 router2.

```

root@astra1:~# traceroute -n 10.10.3.196
traceroute to 10.10.3.196 (10.10.3.196), 30 hops max, 60 byte packets
 1  10.10.3.68  1.398 ms  1.046 ms  1.040 ms
 2  10.10.3.196  3.689 ms  4.128 ms  4.124 ms
root@astra1:~# traceroute -n 10.10.3.196
traceroute to 10.10.3.196 (10.10.3.196), 30 hops max, 60 byte packets
 1  10.10.3.68  5.117 ms  4.809 ms  4.554 ms^C
root@astra1:~# traceroute -n 10.10.3.196
traceroute to 10.10.3.196 (10.10.3.196), 30 hops max, 60 byte packets
 1  10.10.3.68  6.551 ms  6.242 ms  5.987 ms
 2  10.10.3.67  5.211 ms  4.958 ms  4.705 ms
 3  10.10.3.196  4.448 ms  3.388 ms  3.114 ms

```

Отключаем конфигурацию динамической маршрутизации IPv4 по протоколу RIPv2, более она нам не понадобится.

5. Настраиваем динамическую маршрутизацию IPv4 с помощью OSPFv2: создаём instance (экземпляр), interface template (шаблон) и area (область) OSPF в WebFig MikroTik.

Enabled	<input checked="" type="checkbox"/>
Name	ospf-instance-1
Version	2
VRF	main
Router ID	main
Routing Table	▼
Originate Default	▼
Redistribute	<input checked="" type="checkbox"/> connected <input checked="" type="checkbox"/> static <input type="checkbox"/> rip <input type="checkbox"/> ospf <input type="checkbox"/> bgp <input type="checkbox"/> vpn <input type="checkbox"/> dhcp <input type="checkbox"/> fantasy <input type="checkbox"/> modem <input type="checkbox"/> copy

Enabled	<input checked="" type="checkbox"/>
Name	ospf-area-1
Instance	ospf-instance-1
Area ID	0.0.0.0
Type	default

Interface Template (all interfaces)

Enabled	<input checked="" type="checkbox"/>
Interfaces	all
Area	ospf-area-1
Networks	▼
Network Type	broadcast

Для примера проверяем вкладку Neighbors в WebFig router1: OSPF-instance, назначенный на интерфейсы, обнаружил всех своих соседей, которые отправили пакет OSPF Hello. В данном случае это интерфейс ether1 router2 и интерфейсы ether1 ether2 router3.

RouterOS v7.8 (stable)

Instances Interface Templates Interfaces Areas Area Ranges Static Neighbors **Neighbors** LSA

3 items

		▲ Instance	Area	Address	State	State Changes	
	D	ospf-instan	ospf-area-1	10.10.3.3	Full	7	
	D	ospf-instan	ospf-area-1	10.10.3.2	Full	6	
	D	ospf-instan	ospf-area-1	10.10.3.67	Full	6	

Посмотрим пойманные пакеты с ether1 router1 в Wireshark: видим пакеты OSPF Hello, которыми обмениваются роутеры для “встречи” друг друга (с информацией о своём идентификаторе, зоне и прочем) и OSPF Update + Acknowledge для обмена сетями и описанием устройств.

14534	11642.683668	10.10.3.3	224.0.0.5	OSPF	82 Hello Packet
14536	11642.688215	10.10.3.2	224.0.0.5	OSPF	82 Hello Packet
14537	11642.693390	10.10.3.3	224.0.0.5	OSPF	198 LS Update
14539	11643.196111	10.10.3.2	224.0.0.5	OSPF	98 LS Acknowledge
14853	11746.769924	10.10.3.4	224.0.0.5	OSPF	86 Hello Packet
14859	11747.754930	10.10.3.4	224.0.0.6	OSPF	134 LS Update
14860	11747.787362	10.10.3.2	10.10.3.4	OSPF	98 LS Update
14861	11747.820185	10.10.3.4	224.0.0.6	OSPF	110 LS Update
14862	11747.821074	10.10.3.3	224.0.0.5	OSPF	110 LS Update
14863	11747.851425	10.10.3.3	10.10.3.4	OSPF	98 LS Update
14867	11748.286435	10.10.3.4	224.0.0.6	OSPF	78 LS Acknowledge
14868	11748.328667	10.10.3.2	224.0.0.5	OSPF	78 LS Acknowledge

Эти сведения сохранились в базе LSA (Link State Advertisements), которую можно увидеть в следующей вкладке WebFig.

Instances Interface Templates Interfaces Areas Area Ranges Static Neighbors **LSA**

14 items

		▲ Instance	Area	Type	Originator	ID	Link	Link Insta... Id	Sequence	Age
	D	ospf-instan	ospf-area-1	network	10.10.3.195	10.10.3.195		0	80000001	516
	D	ospf-instan	ospf-area-1	network	10.10.3.195	10.10.3.67		0	80000001	415
	D	ospf-instan	ospf-area-1	network	10.10.3.195	10.10.3.3		0	80000002	416
	D	ospf-instan	ospf-area-1	router	10.10.3.196	10.10.3.196		0	80000004	503
	D	ospf-instan	ospf-area-1	router	10.10.3.195	10.10.3.195		0	80000007	415
	SD	ospf-instan	ospf-area-1	router	10.10.3.68	10.10.3.68		0	80000004	403
	D	ospf-instan		external	10.10.3.196	10.10.3.192		0	80000001	531
	D	ospf-instan		external	10.10.3.195	10.10.3.192		0	80000001	1185
	D	ospf-instan		external	10.10.3.195	10.10.3.128		0	80000001	1185
	D	ospf-instan		external	10.10.3.195	10.10.3.64		0	80000001	1185
	SD	ospf-instan		external	10.10.3.68	10.10.3.64		0	80000001	421
	D	ospf-instan		external	10.10.3.196	10.10.3.0		0	80000001	531
	D	ospf-instan		external	10.10.3.195	10.10.3.0		0	80000001	1185
	SD	ospf-instan		external	10.10.3.68	10.10.3.0		0	80000001	421

В IP->Routes также можно увидеть все просчитанные алгоритмом OSPF пути до всех сетей.

7 items				
		▲ Dst. Address	Gateway	Distance
-	DAC	10.10.3.0/26	%ether1	
-	DAC	10.10.3.64/26	%ether2	
-	DAo+	10.10.3.128/26	10.10.3.67%ether2	110
-	DAo+	10.10.3.128/26	10.10.3.3%ether1	110
-	DAo+	10.10.3.192/26	10.10.3.67%ether2	110
-	DAo+	10.10.3.192/26	10.10.3.2%ether1	110
-	DAo+	10.10.3.192/26	10.10.3.3%ether1	110

Демонстрация отказоустойчивости сети: производим ping от astra1 до router2 как в пункте 4. Изначальный маршрут: astra1 -> router1-> router2. Отключаем интерфейс ether1 на router1 и без дополнительной перезагрузки OSPF понимает, что расстояние до этого интерфейса стало = 16 (бесконечное в терминологии OSPF). Поэтому router1 переадресовывает пакет ближайшему маршруту до router2: интерфейсу ether2 router3. Всё происходит практически мгновенно и traceroute отображает информацию сразу же, без задержек, как, например, уже было в этой работе с протоколом RIPv2.

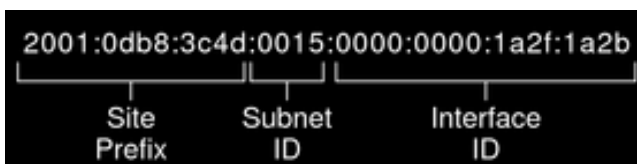
```

root@astra1:~# traceroute -n 10.10.3.196
traceroute to 10.10.3.196 (10.10.3.196), 30 hops max, 60 byte packets
 1  10.10.3.68  6.619 ms  6.322 ms  6.073 ms
 2  10.10.3.196  1.544 ms  2.343 ms  2.837 ms
root@astra1:~# traceroute -n 10.10.3.196
traceroute to 10.10.3.196 (10.10.3.196), 30 hops max, 60 byte packets
 1  10.10.3.68  5.944 ms  5.221 ms  4.936 ms
 2  10.10.3.67  4.654 ms  4.309 ms  4.038 ms
 3  10.10.3.196  3.518 ms  3.135 ms  3.461 ms

```

Отключаем конфигурацию динамической маршрутизации IPv4 с помощью OSPFv2, более она нам не понадобится.

6. В задании сказано, что выделен префикс с маской /64, однако скорее всего это не так и имелось в виду, что доступна для нашего вмешательства маска /64 (MikroTik не поддерживает поддержку префиксов с маской более /64, а для выделения 4 подсетей из этой маски понадобилась бы минимум маска /66). Мне выделен префикс **fd00:2003:4::/48**,



(Oracle - IPv6 Addressing Overview)

Согласно структуре IPv6-адреса, обозначим четвёртый октет (fd00:2003:0004:**0000**:0000:0000:0000:0000) как целевой, так как все последующие будут отвечать за адрес устройств (интерфейсов). Фиксируем в нём номера наших подсетей (0/1/2/3). Префиксы для каждой подсети видоизменяются до fd00:2003:4:0::/64, fd00:2003:4:1::/64, fd00:2003:4:2::/64 и fd00:2003:4:3::/64.

Таким образом, выделенные мной префиксы для построения 4 подсетей:

Networks (on nibble-boundary) (4 total)

```
fd00:2003:0004:0000:0000:0000:0000:0000/64
fd00:2003:0004:0001:0000:0000:0000:0000/64
fd00:2003:0004:0002:0000:0000:0000:0000/64
fd00:2003:0004:0003:0000:0000:0000:0000/64
```

Добавим соответствующие префиксы на роутерах MikroTik и включим их распространение по своим подсетям vboxnet0/1/2/3). В подсети vboxnet0 префикс назначен на router1 (ether1), в vboxnet1 - также router1 (ether2), vboxnet2 - router3 (ether3), vboxnet3 - router2 (ether2).

router1

```
[admin@mt-01] > ipv6 address add address=fd00:2003:4:0::/64 interface=ether1 advertise=yes eui-64=yes
[admin@mt-01] > ipv6 address add address=fd00:2003:4:1::/64 interface=ether2 advertise=yes eui-64=yes
[admin@mt-01] > ipv6 address print
Flags: D - DYNAMIC; G, L - LINK-LOCAL
Columns: ADDRESS, INTERFACE, ADVERTISE
# ADDRESS INTERFACE ADVERTISE
0 DL fe80::a00:27ff:fe1f:9c5f/64 ether1 no
1 DL fe80::a00:27ff:febe:f2d5/64 ether2 no
2 G fd00:2003:4:0:a00:27ff:fe1f:9c5f/64 ether1 yes
3 G fd00:2003:4:1:a00:27ff:febe:f2d5/64 ether2 yes
```

router2

```
[admin@mt-02] > ipv6 address add address=fd00:2003:4:3::/64 interface=ether2 advertise=yes eui-64=yes
[admin@mt-02] > ipv6 address print
Flags: D - DYNAMIC; G, L - LINK-LOCAL
Columns: ADDRESS, INTERFACE, ADVERTISE
# ADDRESS INTERFACE ADVERTISE
0 DL fe80::a00:27ff:fe1c:ba6b/64 ether1 no
1 DL fe80::a00:27ff:fe37:3003/64 ether2 no
2 G fd00:2003:4:3:a00:27ff:fe37:3003/64 ether2 yes
```

router3

```
[admin@mt-03] > ipv6 address add address=fd00:2003:4:2::/64 interface=ether3 advertise=yes eui-64=yes
[admin@mt-03] > ipv6 address print
Flags: D - DYNAMIC; G, L - LINK-LOCAL
Columns: ADDRESS, INTERFACE, ADVERTISE
# ADDRESS INTERFACE ADVERTISE
0 DL fe80::a00:27ff:feaf:285a/64 ether1 no
1 DL fe80::a00:27ff:fe09:e653/64 ether2 no
2 DL fe80::a00:27ff:fef6:6902/64 ether3 no
3 DL fe80::a00:27ff:fef2:49ab/64 ether4 no
4 G fd00:2003:4:2:a00:27ff:fef6:6902/64 ether3 yes
```

astral

```

eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
    link/ether 08:00:27:f1:47:41 brd ff:ff:ff:ff:ff:ff
    inet 10.10.3.66/26 brd 10.10.3.127 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fd00:2003:4:1:a00:27ff:fe1:4741/64 scope global mngtmpaddr dynamic
        valid_lft 2591771sec preferred_lft 604571sec
    inet6 fe80::a00:27ff:fe1:4741/64 scope link
        valid_lft forever preferred_lft forever
eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
    link/ether 08:00:27:f9:d1:28 brd ff:ff:ff:ff:ff:ff
    inet 10.10.3.132/26 brd 10.10.3.191 scope global eth1
        valid_lft forever preferred_lft forever
    inet6 fd00:2003:4:2:a00:27ff:fe9:d128/64 scope global mngtmpaddr dynamic
        valid_lft 2591868sec preferred_lft 604668sec

```

astra2

```

eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
    link/ether 08:00:27:72:06:7d brd ff:ff:ff:ff:ff:ff
    inet 10.10.3.130/26 brd 10.10.3.191 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fd00:2003:4:2:a00:27ff:fe72:67d/64 scope global mngtmpaddr dynamic
        valid_lft 2591759sec preferred_lft 604559sec
    inet6 fe80::a00:27ff:fe72:67d/64 scope link
        valid_lft forever preferred_lft forever
eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
    link/ether 08:00:27:fd:ed:19 brd ff:ff:ff:ff:ff:ff
    inet 10.10.3.194/26 brd 10.10.3.255 scope global eth1
        valid_lft forever preferred_lft forever
    inet6 fd00:2003:4:3:a00:27ff:fed:ed19/64 scope global mngtmpaddr dynamic
        valid_lft 2591786sec preferred_lft 604586sec
    inet6 fe80::a00:27ff:fed:ed19/64 scope link
        valid_lft forever preferred_lft forever

```

Как видно на скриншотах, astra1/2 успешно получили префиксы IPv6 от своих роутеров, однако роутеры между собой не желают принимать чужие префиксы. Исправим это, установив настройку “ipv6/settings/set/accept-router-advertisements” в роутерах 2 и 3 в значение “yes”. Router1 в нашем случае является уникальным распространителем с уже назначенными префиксами на свои интерфейсы. Перезагрузим все роутеры для получения новых пакетов IPv6 Advertisement. Ниже видны новые префиксные адреса router2 router3.

router2

```

[admin@mt-02] > ipv6 address print
Flags: D - DYNAMIC; G, L - LINK-LOCAL
Columns: ADDRESS, INTERFACE, ADVERTISE
#   ADDRESS                                INTERFACE  ADVERTISE
0   G fd00:2003:4:3:a00:27ff:fe37:3003/64  ether2     yes
1   DL fe80::a00:27ff:fe1c:ba6b/64         ether1     no
2   DL fe80::a00:27ff:fe37:3003/64         ether2     no
3   DG fd00:2003:4:0:a00:27ff:fe1c:ba6b/64  ether1     no

```

router3


```
[admin@mt-03] > ipv6 address print
Flags: D - DYNAMIC; G, L - LINK-LOCAL
Columns: ADDRESS, INTERFACE, ADVERTISE
# ADDRESS INTERFACE ADVERTISE
0 G fd00:2003:4:2:a00:27ff:fe1f:9c5f/64 ether3 yes
1 DL fe80::a00:27ff:feaf:285a/64 ether1 no
2 DL fe80::a00:27ff:fe09:e653/64 ether2 no
3 DL fe80::a00:27ff:fe1f:9c5f/64 ether3 no
4 DL fe80::a00:27ff:fe1f:9c5f/64 ether4 no
5 DG fd00:2003:4:1:a00:27ff:fe09:e653/64 ether2 no
6 DG fd00:2003:4:0:a00:27ff:feaf:285a/64 ether1 no
7 DG fd00:2003:4:3:a00:27ff:fe1f:9c5f/64 ether4 no
```

Пробуем пинговать устройства в одной подсети: всё работает.

```
root@astra1:~# ping -6 fd00:2003:4:0:a00:27ff:fe1f:9c5f
PING fd00:2003:4:0:a00:27ff:fe1f:9c5f(fd00:2003:4:0:a00:27ff:fe1f:9c5f) 56 data bytes
64 bytes from fd00:2003:4:0:a00:27ff:fe1f:9c5f: icmp_seq=1 ttl=64 time=1.37 ms
64 bytes from fd00:2003:4:0:a00:27ff:fe1f:9c5f: icmp_seq=2 ttl=64 time=0.716 ms
64 bytes from fd00:2003:4:0:a00:27ff:fe1f:9c5f: icmp_seq=3 ttl=64 time=0.893 ms
```

НЕ ПО ЗАДАНИЮ РАБОТЫ // Настроим статическую маршрутизацию по IPv6: разница между маршрутизацией IPv4 отличается в команде **route -6** (а не route) на astralinux и **ipv6 route** (а не ip route) на MikroTik. Для начала на машинах astra1 и astra2 необходимо включить ipv6-forwarding, убрав комментарий в файле /etc/sysctl.conf. Note: включение этого параметра сделает недоступным получение/обновление IPv6 адреса по SLAAC, поэтому к этому времени он уже должен быть сконфигурирован/получен.

```
GNU nano 2.7.4 Файл: /etc/sysctl.conf

# prevent some spoofing attacks
net.ipv4.conf.default.rp_filter=2
net.ipv4.conf.all.rp_filter=2

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
# Enabling this option disables Stateless Address Autoconfiguration
# based on Router Advertisements for this host
net.ipv6.conf.all.forwarding=1
```

```
root@astra2:~# sysctl -p
net.ipv4.conf.default.rp_filter = 2
net.ipv4.conf.all.rp_filter = 2
net.ipv4.ip_forward = 1
net.ipv6.conf.all.forwarding = 1
```

Демонстрация сконфигурированных таблиц IPv6-маршрутизации на каждой машине:

astra1

```

root@astral1:~# route -6 add fd00:2003:4::/64 gw fd00:2003:4:1:a00:27ff:febe:f2d5
root@astral1:~# route -6 add fd00:2003:4:3::/64 gw fd00:2003:4:1:a00:27ff:febe:f2d5
root@astral1:~# route -6
Kernel IPv6 routing table

```

Destination	Next Hop	Flag	Met	Ref	Use	If
fd00:2003:4::/64	fd00:2003:4:1:a00:27ff:febe:f2d5	UG	1	1	0	eth0
fd00:2003:4:1::/64	:::	UAe	256	4	0	eth0
fd00:2003:4:2::/64	:::	UAe	256	3	0	eth1
fd00:2003:4:3::/64	fd00:2003:4:1:a00:27ff:febe:f2d5	UG	1	1	0	eth0

astral2

```

root@astral2:~# route -6
Kernel IPv6 routing table

```

Destination	Next Hop	Flag	Met	Ref	Use	If
fd00:2003:4::/64	fd00:2003:4:2:a00:27ff:fe9:d128	UG	1	3	0	eth0
fd00:2003:4:1::/64	fd00:2003:4:2:a00:27ff:fe9:d128	UG	1	2	0	eth0
fd00:2003:4:2::/64	:::	UAe	256	4	0	eth0
fd00:2003:4:3::/64	:::	UAe	256	2	0	eth1

router1

```

[admin@mt-01] > ipv6 route add dst-address=fd00:2003:4:2::/64 gateway=fd00:2003:
4:0:a00:27ff:fe1c:ba6b
[admin@mt-01] > ipv6 route add dst-address=fd00:2003:4:3::/64 gateway=fd00:2003:
4:0:a00:27ff:fe1c:ba6b
[admin@mt-01] > ipv6 route print
Flags: D - DYNAMIC; A - ACTIVE; c, s, y - COPY
Columns: DST-ADDRESS, GATEWAY, DISTANCE

```

#	DST-ADDRESS	GATEWAY	DISTANCE
	Dac fd00:2003:4::/64	ether1	0
	Dac fd00:2003:4:1::/64	ether2	0
0	As fd00:2003:4:2::/64	fd00:2003:4:0:a00:27ff:fe1c:ba6b	1
1	As fd00:2003:4:3::/64	fd00:2003:4:0:a00:27ff:fe1c:ba6b	1

router2

```

[admin@mt-02] > ipv6 route add dst-address=fd00:2003:4:1::/64 gateway=fd00:2003:
4:3:a00:27ff:fed:ed19
[admin@mt-02] > ipv6 route add dst-address=fd00:2003:4:2::/64 gateway=fd00:2003:
4:3:a00:27ff:fed:ed19
[admin@mt-02] > ipv6 route print
Flags: D - DYNAMIC; A - ACTIVE; c, s, y - COPY
Columns: DST-ADDRESS, GATEWAY, DISTANCE

```

#	DST-ADDRESS	GATEWAY	DISTANCE
	Dac fd00:2003:4::/64	ether1	0
0	As fd00:2003:4:1::/64	fd00:2003:4:3:a00:27ff:fed:ed19	1
1	As fd00:2003:4:2::/64	fd00:2003:4:3:a00:27ff:fed:ed19	1
	Dac fd00:2003:4:3::/64	ether2	0

Попробуем пинговать astral2 к router1, чтобы пройти полный круг по часовой стрелке:

```

root@astral2:~# ping fd00:2003:4:0:a00:27ff:fe1f:9c5f
PING fd00:2003:4:0:a00:27ff:fe1f:9c5f(fd00:2003:4:0:a00:27ff:fe1f:9c5f) 56 data bytes
64 bytes from fd00:2003:4:0:a00:27ff:fe1f:9c5f: icmp_seq=1 ttl=63 time=2.82 ms
64 bytes from fd00:2003:4:0:a00:27ff:fe1f:9c5f: icmp_seq=2 ttl=63 time=1.71 ms
64 bytes from fd00:2003:4:0:a00:27ff:fe1f:9c5f: icmp_seq=3 ttl=63 time=1.81 ms
64 bytes from fd00:2003:4:0:a00:27ff:fe1f:9c5f: icmp_seq=4 ttl=63 time=1.69 ms

```

Смотрим через Wireshark захваченные пакеты с каждого устройства:

От eth0 astral2 к eth1 astral1 (начало):

4623	6333.716422	fe80::a00:27ff:fe72...	fe80::7a82:9f13:12d...	ICMPv6	86 Neighbor Solicitation for fe80::7a82:9f13:12de:9b58 from 08:0...
4624	6333.716546	fe80::7a82:9f13:12d...	fe80::a00:27ff:fe72...	ICMPv6	86 Neighbor Advertisement fe80::7a82:9f13:12de:9b58 (sol, ovr) i...
4625	6338.941404	fd00:2003:4:2:a00:2...	fd00:2003:4:0:a00:2...	ICMPv6	118 Echo (ping) request id=0x03c9, seq=1, hop limit=64 (no respon...
4626	6339.942750	fd00:2003:4:2:a00:2...	fd00:2003:4:0:a00:2...	ICMPv6	118 Echo (ping) request id=0x03c9, seq=2, hop limit=64 (no respon...
4627	6340.944917	fd00:2003:4:2:a00:2...	fd00:2003:4:0:a00:2...	ICMPv6	118 Echo (ping) request id=0x03c9, seq=3, hop limit=64 (no respon...
4628	6341.946851	fd00:2003:4:2:a00:2...	fd00:2003:4:0:a00:2...	ICMPv6	118 Echo (ping) request id=0x03c9, seq=4, hop limit=64 (no respon...

От eth0 astral1 к ether2 router1:

1103	6340.213723	fd00:2003:4:2:a00:2...	fd00:2003:4:0:a00:2...	ICMPv6	118 Echo (ping) request id=0x03c9, seq=1, hop limit=64
1104	6341.215070	fd00:2003:4:2:a00:2...	fd00:2003:4:0:a00:2...	ICMPv6	118 Echo (ping) request id=0x03c9, seq=2, hop limit=64
1105	6342.217237	fd00:2003:4:2:a00:2...	fd00:2003:4:0:a00:2...	ICMPv6	118 Echo (ping) request id=0x03c9, seq=3, hop limit=64
1106	6343.219177	fd00:2003:4:2:a00:2...	fd00:2003:4:0:a00:2...	ICMPv6	118 Echo (ping) request id=0x03c9, seq=4, hop limit=64

От ether1 router1 к ether1 router2:

245	91.405116	fe80::a00:27ff:fe1f...	fe80::a00:27ff:fe1c...	ICMPv6	86 Neighbor Solicitation for fe80::a00:27ff:fe1c:ba6b from 08:00
246	91.405521	fe80::a00:27ff:fe1c...	fe80::a00:27ff:fe1f...	ICMPv6	78 Neighbor Advertisement fe80::a00:27ff:fe1c:ba6b (rtr, sol)
247	94.485758	fd00:2003:4:0:a00:2...	fd00:2003:4:2:a00:2...	ICMPv6	118 Echo (ping) reply id=0x03cb, seq=1, hop limit=64
248	95.487666	fd00:2003:4:0:a00:2...	fd00:2003:4:2:a00:2...	ICMPv6	118 Echo (ping) reply id=0x03cb, seq=2, hop limit=64
249	96.489426	fd00:2003:4:0:a00:2...	fd00:2003:4:2:a00:2...	ICMPv6	118 Echo (ping) reply id=0x03cb, seq=3, hop limit=64

От ether2 router2 к eth1 astra2 (конец):

143	90.243401	fe80::a00:27ff:fe1f...	fe80::a00:27ff:fe1c...	ICMPv6	86 Neighbor Solicitation for fe80::a00:27ff:fe1c:ba6b from 08:00
144	90.243737	fe80::a00:27ff:fe1c...	fe80::a00:27ff:fe1f...	ICMPv6	78 Neighbor Advertisement fe80::a00:27ff:fe1c:ba6b (rtr, sol)
145	93.324032	fd00:2003:4:0:a00:2...	fd00:2003:4:2:a00:2...	ICMPv6	118 Echo (ping) reply id=0x03cb, seq=1, hop limit=64
146	94.325938	fd00:2003:4:0:a00:2...	fd00:2003:4:2:a00:2...	ICMPv6	118 Echo (ping) reply id=0x03cb, seq=2, hop limit=64
147	95.327692	fd00:2003:4:0:a00:2...	fd00:2003:4:2:a00:2...	ICMPv6	118 Echo (ping) reply id=0x03cb, seq=3, hop limit=64

//

7. Настраиваем динамическую маршрутизацию IPv6 с помощью OSPFv3: создаём новые instance, area и interface template на всех роутерах как в пункте 4 или же для простоты просто меняем версию OSPF в instance на 3 для использования OSPFv3 (для IPv6).

Enabled	<input checked="" type="checkbox"/>
Name	ospf-instance-1
Version	3
VRF	main
Router ID	main
Routing Table	▼
Originate Default	▼
Redistribute	<input checked="" type="checkbox"/> connected <input checked="" type="checkbox"/> static <input type="checkbox"/> rip <input type="checkbox"/> ospf <input type="checkbox"/> bgp <input type="checkbox"/> vpn <input type="checkbox"/> dhcp <input type="checkbox"/> fantasy <input type="checkbox"/> modem <input type="checkbox"/> copy

Enabled	<input checked="" type="checkbox"/>
Name	ospf-area-1
Instance	ospf-instance-1
Area ID	0.0.0.0
Type	default

Enabled	<input checked="" type="checkbox"/>
Interfaces	all
Area	ospf-area-1
Networks	▼
Network Type	broadcast
Prefix List	▼
Instance ID	0
Cost	1
Priority	128
Passive	<input type="checkbox"/>

В Neighbors также видим соседей, в LSA - данные всех подключений, которыми с нами поделились. Пробуем пинговать router2 с astra1 - всё прекрасно работает. Смотрим маршрут получения наших пакетов: сначала они доходят до адреса fd00:2003:4:1:a00:27ff:febe:f2d5 (router1), и далее переходят к router2.

Проверяем отказоустойчивость сети: выключаем интерфейс ether1 на router2 и посмотрим, каким путём пойдёт наш пакет. Ожидаемо, с router1 он был

3 items						
		▲ Instance	Area	Address	State	State Changes
	D	ospf-instan	ospf-area-1	fe80::a00:27ff:fe0	Full	6
	D	ospf-instan	ospf-area-1	fe80::a00:27ff:fe1	Full	6
	D	ospf-instan	ospf-area-1	fe80::a00:27ff:fea	Full	6

отправлен на router3 и оттуда нашёл прямой путь до ether2 router2.

ping и traceroute от astral1 до ether2 router2 до проверки отказоустойчивости (пакеты приходят на ether1 router2 и переадресовываются на ether2 через IPv6 forwarding):

```
root@astral1:~# ping -6 fd00:2003:4:3:a00:27ff:fe37:3003
PING fd00:2003:4:3:a00:27ff:fe37:3003(fd00:2003:4:3:a00:27ff:fe37:3003) 56 data bytes
64 bytes from fd00:2003:4:3:a00:27ff:fe37:3003: icmp_seq=1 ttl=63 time=1.99 ms
64 bytes from fd00:2003:4:3:a00:27ff:fe37:3003: icmp_seq=2 ttl=63 time=1.35 ms
^C
--- fd00:2003:4:3:a00:27ff:fe37:3003 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 1.356/1.674/1.992/0.318 ms
root@astral1:~# traceroute -6 fd00:2003:4:3:a00:27ff:fe37:3003
traceroute to fd00:2003:4:3:a00:27ff:fe37:3003 (fd00:2003:4:3:a00:27ff:fe37:3003), 30 hops max, 80 byte packets
 1 fd00:2003:4:1:a00:27ff:febe:f2d5 (fd00:2003:4:1:a00:27ff:febe:f2d5) 7.452 ms 7.154 ms 6.883 ms
 2 fd00:2003:4:3:a00:27ff:fe37:3003 (fd00:2003:4:3:a00:27ff:fe37:3003) 6.052 ms 2.319 ms 2.060 ms
```

traceroute от astral1 до ether2 router2 после отключения ether1 router2:

```
root@astral1:~# traceroute -6 fd00:2003:4:3:a00:27ff:fe37:3003
traceroute to fd00:2003:4:3:a00:27ff:fe37:3003 (fd00:2003:4:3:a00:27ff:fe37:3003), 30 hops max, 80 byte packets
 1 fd00:2003:4:1:a00:27ff:febe:f2d5 (fd00:2003:4:1:a00:27ff:febe:f2d5) 5.894 ms 5.569 ms 0.736 ms
 2 fd00:2003:4:1:a00:27ff:fe09:e653 (fd00:2003:4:1:a00:27ff:fe09:e653) 1.827 ms fd00:2003:4:0:a00:27ff:feaf:285a (fd00:2003:4:0:a00:27ff:feaf:285a) 1.832 ms 1.541 ms
 3 fd00:2003:4:3:a00:27ff:fe37:3003 (fd00:2003:4:3:a00:27ff:fe37:3003) 3.239 ms 2.976 ms 4.313 ms
```

Пакеты на ether2 router1 (в данном случае - отправка и получение ping request/reply к/от router3):

31766	41001.936935	fd00:2003:4:1:a00:2...	fd00:2003:4:3:a00:2...	ICMPv6	118 Echo (ping) request id=0x07b6, seq=1, hop limit=64 (reply in
31767	41001.937962	fe80::a00:27ff:febe...	ff02::1:fff1:4741	ICMPv6	86 Neighbor Solicitation for fd00:2003:4:1:a00:27ff:fe1:4741 fr
31768	41001.938215	fd00:2003:4:1:a00:2...	fe80::a00:27ff:febe...	ICMPv6	86 Neighbor Advertisement fd00:2003:4:1:a00:27ff:fe1:4741 (rtr,
31769	41001.938215	fd00:2003:4:3:a00:2...	fd00:2003:4:1:a00:2...	ICMPv6	118 Echo (ping) reply id=0x07b6, seq=1, hop limit=63 (request in
31770	41002.937676	fd00:2003:4:1:a00:2...	fd00:2003:4:3:a00:2...	ICMPv6	118 Echo (ping) request id=0x07b6, seq=2, hop limit=64 (reply in
31771	41002.938678	fd00:2003:4:3:a00:2...	fd00:2003:4:1:a00:2...	ICMPv6	118 Echo (ping) reply id=0x07b6, seq=2, hop limit=63 (request in

Пакеты на ether2 router2 (получение и отправка ping request/reply к/от router3):

33945	40581.212309	fe80::a00:27ff:fe1c...	fe80::a00:27ff:fe1f...	ICMPv6	86 Neighbor Solicitation for fe80::a00:27ff:fe1f:9c5f from 08:00.
33946	40581.212728	fe80::a00:27ff:fe1f...	fe80::a00:27ff:fe1c...	ICMPv6	78 Neighbor Advertisement fe80::a00:27ff:fe1f:9c5f (rtr, sol)
34456	41000.864363	fd00:2003:4:1:a00:2...	fd00:2003:4:3:a00:2...	ICMPv6	118 Echo (ping) request id=0x07b6, seq=1, hop limit=63 (reply in
34457	41000.864705	fd00:2003:4:3:a00:2...	fd00:2003:4:1:a00:2...	ICMPv6	118 Echo (ping) reply id=0x07b6, seq=1, hop limit=64 (request in
34460	41001.865123	fd00:2003:4:1:a00:2...	fd00:2003:4:3:a00:2...	ICMPv6	118 Echo (ping) request id=0x07b6, seq=2, hop limit=63 (reply in
34461	41001.865453	fd00:2003:4:3:a00:2...	fd00:2003:4:1:a00:2...	ICMPv6	118 Echo (ping) reply id=0x07b6, seq=2, hop limit=64 (request in

Пакеты, отправляемые по протоколу OSPF, используя OSPFv3:

29836	38973.310864	fe80::a00:27ff:febe...	ff02::5	OSPF	94 Hello Packet
29837	38973.311431	fe80::a00:27ff:fe09...	ff02::5	OSPF	94 Hello Packet
29838	38973.392873	fe80::a00:27ff:febe...	ff02::5	OSPF	90 LS Acknowledge
29839	38980.887751	fe80::a00:27ff:febe...	ff02::5	OSPF	130 LS Update
29840	38980.888197	fe80::a00:27ff:fe09...	ff02::5	OSPF	130 LS Update
29841	38981.389961	fe80::a00:27ff:febe...	ff02::5	OSPF	90 LS Acknowledge

Все задания практической работы выполнены успешно.