

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ  
КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

**ПРАКТИЧЕСКАЯ РАБОТА №2**

по дисциплине «Сети ЭВМ и телекоммуникации»

Тема: «Адресация узлов в сетях. MAC-адрес. Сетевые адреса IPv4.

Протокол ARP. Статическая и динамическая конфигурация узлов»

Выполнил: студент группы ИС-142

Наумов А. А.

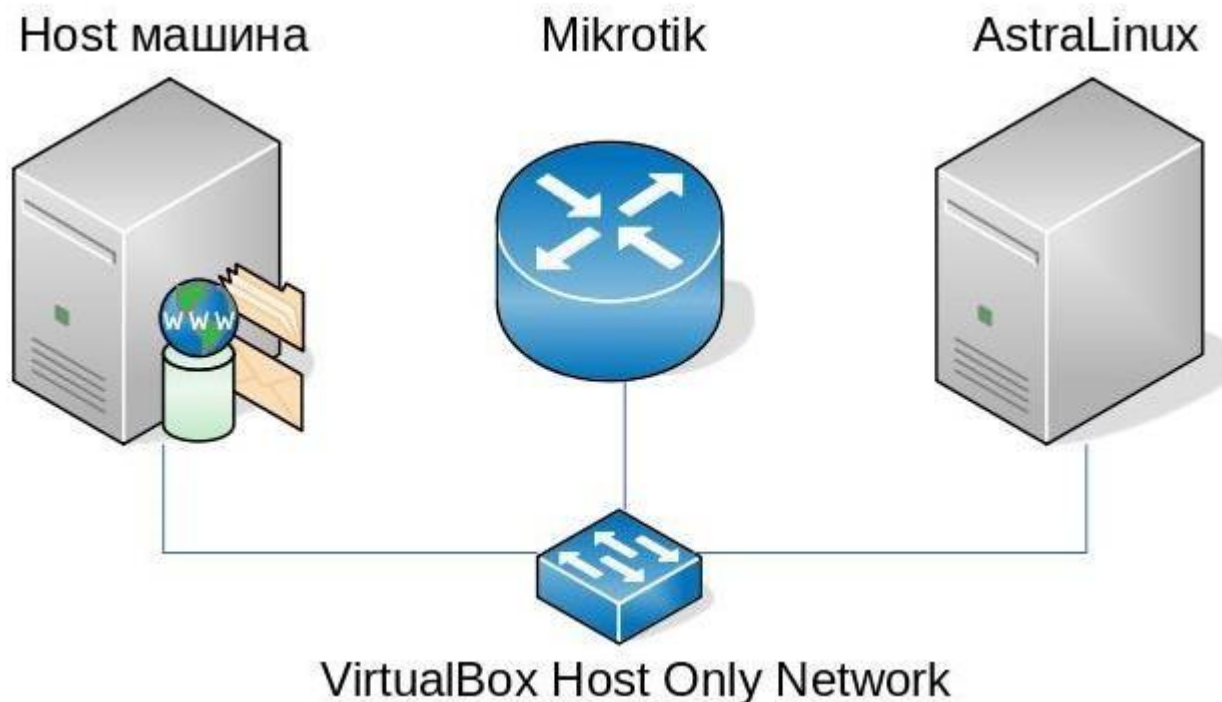
Проверил: доцент

кафедры ВС Перышкова Е.Н.

Новосибирск 2023

## Задание

1. Собрать конфигурацию сети, представленной на рисунке.

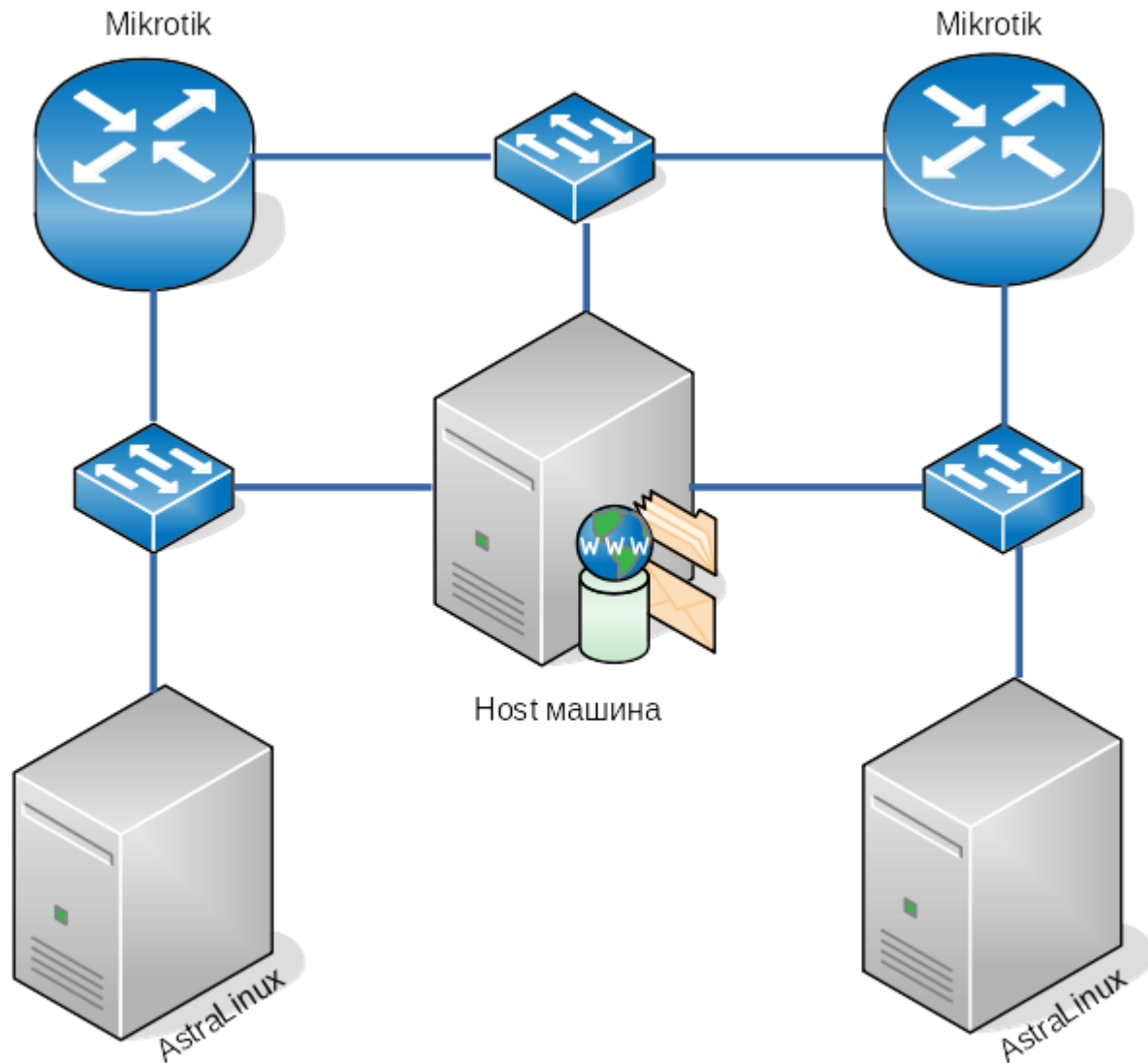


2. Определить MAC адреса сетевых интерфейсов маршрутизатора Mikrotik, AstraLinux, хост-машины.
3. Выделено адресное пространство 10.N.0.0/16, где N - это Ваш порядковый номер в журнале. Спроектировать его таким образом, чтобы он был разделён на 4 равные по количеству адресов подсети (количество адресов в каждой подсети - максимально возможное). Записать диапазоны адресов для каждой полученной подсети. Указать адрес сети и адрес для широковещательной передачи данных.
4. Выбрать один из рассчитанных в п. 3 диапазонов адресов и сконфигурировать соответствующим образом сетевые интерфейсы хост-машины, Astra Linux и Mikrotik. Адреса для узлов из назначенного диапазона выбираются произвольно.
5. Запустить на host-машине Wireshark и проверить связность хоста с Astra Linux и Mikrotik с использованием протокола ICMP и режима ping-pong. Показать этапы работы протокола разрешения сетевых адресов (ARP) и протокола контроля сетевого подключения (ICMP). Сравнить MAC-адрес адаптера хост-машины с адресом в захваченных пакетах.
6. Запустить на host-машине Wireshark. Запустить ping с Astra Linux до Mikrotik. Проанализировать полученный поток пакетов.
7. Перезапустить виртуальную машину Astra Linux в режиме записи потока сетевых пакетов. Повторить действия пункта 6. Показать, что в захваченном

потоке пакетов присутствуют все пакеты, отправляемые и получаемые виртуальной машиной. Объяснить, почему поток захваченных пакетов отличается от потока в п.6.

8. Запустить на host-машине Wireshark. С хост-машины из назначенного диапазона адресов "пропинговать" адрес сети и широковещательный адрес. Что поменяется в последовательности пакетов в сравнении с пингом простых адресов? В каком случае будет использоваться протокол ARP и почему? Могут ли в сети оказаться узлы, на интерфейсах которых назначены такие адреса и будет работать протокол ARP? Если могут, продемонстрировать.
9. На машине Astra Linux установить пакет для конфигурирования сетевых интерфейсов с использованием APIPA.
10. Запустить Astra Linux в режиме захвата пакетов. Настроить сетевой интерфейс так, чтобы для его для конфигурирования использовалась APIPA. Запустить процесс конфигурации интерфейса. В захваченном потоке пакетов показать, какие действия выполнялись при конфигурации интерфейса.
11. Используя web-интерфейс Mikrotik, установить на нем DHCP сервер. Адреса в DHCP должны выдаваться из иного диапазона, чем тот, который был выбран в п. 4. В настройках DHCP указать, что все пакеты протокола должны передаваться в широковещательном режиме. Если в Mikrotik запущен DHCP-client, остановить его.
12. Запустить на host-машине Wireshark. В Astra Linux создать виртуальный интерфейс (например eth0:1), настроить его на получение сетевых настроек в автоматическом режиме. "Поднять" интерфейс, определить какой адрес был получен для назначения на сетевой интерфейс. На хост-машине отфильтровать захваченный поток пакетов так, чтобы отображались только пакеты протокола DHCP. Какие типы пакетов были захвачены? Подождать время, на которое была выдана аренда адреса (допускается в настройках DHCP сервера это время уменьшить) и показать, какие пакеты были отправлены клиентом и сервером. Остановить DHCP сервер и показать, что будет делать клиент в этом случае. Будут ли все эти пакеты видны на host-машине? Если необходимо, перезапустить Astra Linux в режиме захвата пакетов и показать полный перечень пакетов, которые появляются в этом случае.

13. Изменить конфигурацию сети в соответствии с рисунком.



14. На второй виртуальной машине Astra Linux сконфигурировать интерфейс так, чтобы он получал настройки автоматически. На маршрутизаторе Mikrotik установить второй DHCP сервер, чтобы он выдавал адреса из свободного диапазона, рассчитанного в п. 3. Попробовать получить сетевые настройки. Объяснить, почему не получилось.
15. Настроить сеть между маршрутизаторами Mikrotik в соответствии со свободным диапазоном из п. 3. Настроить на втором Mikrotik dhcp-relay агент. Запустить на host-машине Wireshark. Получить сетевые настройки для интерфейса второй виртуальной машины Astra Linux. Какие пакеты "видны" на хост машине и почему? Перезапустить виртуальную машину Astra Linux в режиме захвата пакетов и получить сетевые настройки. Указать, какие типы пакетов DHCP были отправлены.
16. Перезапустить маршрутизаторы Mikrotik в режиме захвата пакетов. Получить сетевые настройки на второй виртуальной машине. Показать, какие DHCP пакеты передавались между маршрутизаторами.

# Выполнение работы

## 1. Конфигурация сети

Конфигурация, как на рисунке, уже была настроена в предыдущей практической работе. Для выполнения этой работы я отключил DHCP-сервер на адаптере VirtualBox.

Имя	IPv4 префикс	IPv6 префикс	DHCP сервер
vboxnet0	192.168.56.1/24		Включен

Адаптер

DHCP сервер

☐ Включить сервер

Адрес сервера: 192.168.56.100

Маска сети сервера: 255.255.255.0

Нижняя граница адресов: 192.168.56.101

Верхняя граница адресов: 192.168.56.254

Сбросить

Применить

## 2. MAC адреса сетевых интерфейсов

На хост-машине:

```
alexeynaumov@Lenovo-Legion-5-15ARH05H-267a6435:~/ArcPC/simplecomputer$ ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eno1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN mode DEFAULT group default ql
   en 1000
   link/ether 8c:8c:aa:27:45:d6 brd ff:ff:ff:ff:ff:ff
   altname enp3s0
3: wlp4s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode DORMANT group default qlen
   1000
   link/ether b0:7d:64:84:14:e9 brd ff:ff:ff:ff:ff:ff
4: vboxnet0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 1000
   link/ether 0a:00:27:00:00:00 brd ff:ff:ff:ff:ff:ff
```

0A:00:27:00:00:00

На Astra Linux:

```
owner@astra:~$ ip -c link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT group default qlen 1000
   link/ether 08:00:27:68:cd:e0 brd ff:ff:ff:ff:ff:ff
```

08:00:27:68:cd:e0

На Mikrotik:

```
[admin@mt-01] > interface ethernet print
Flags: R - RUNNING
Columns: NAME, MTU, MAC-ADDRESS, ARP
# NAME MTU MAC-ADDRESS ARP
0 R ether1 1500 08:00:27:5E:D2:7C enabled
```

```
08:00:27:5E:D2:7C
```

### 3. Адресное пространство

Выделено адресное пространство 10.10.0.0/16. Маска - 255.255.0.0.

Чтобы разделить на 4 подсети, фиксирую 2 старших бита в третьей части адреса.

Таким образом:

1. 00001010.00000110.00000000.00000000: от 10.6.0.0 до 10.6.63.255;
2. 00001010.00000110.01000000.00000000: от 10.6.64.0 до 10.6.127.255;
3. 00001010.00000110.10000000.00000000: от 10.6.128.0 до 10.6.191.255;
4. 00001010.00000110.11000000.00000000: от 10.6.192.0 до 10.6.255.255.

При этом 10.10.\*.0 будут адресами сети, а 10.10.\*.255 - адресами broadcast (свой для каждой подсети). Маска станет 255.255.192.0 или /18.

### 4. Конфигурация сетевых интерфейсов ВМ

Установлю адаптеру адрес 10.10.0.1 с маской подсети 255.255.192.0.

```
VBoxManage hostonlyif ipconfig vboxnet0 --ip=10.10.0.0 --netmask=255.255.192.0
```

vboxnet0	10.10.0.1/18	fe80::800:27ff:fe00:0/64	Выключен

Адаптер

DHCP сервер

☐ Настроить адаптер автоматически

☒ Настроить адаптер вручную

IPv4 адрес:

IPv4 маска сети:

```
alexeynaumov@Lenovo-Legion-5-15ARH05H-267a6435:~$ VBoxManage hostonlyif ipconfig vboxnet0 --ip=10.10.0.0 --netmask=255.255.0.0
```

На BM Astra Linux отредактирую файл /etc/network/interfaces.d/eth0:

```
owner@astra:~$ cat /etc/network/interfaces.d/eth0
auto eth0
iface eth0 inet static
    address 10.10.0.2
    netmask 255.255.192.0
```



Перезапускаю интерфейс eth0 и убеждаюсь, что подключение появилось:

```
owner@astra:~$ sudo ifup eth0
owner@astra:~$ ip -c a show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
    link/ether 08:00:27:68:cb:e0 brd ff:ff:ff:ff:ff:ff
    inet 10.10.0.2/18 brd 10.10.63.255 scope global eth0
        valid_lft forever preferred_lft forever
```

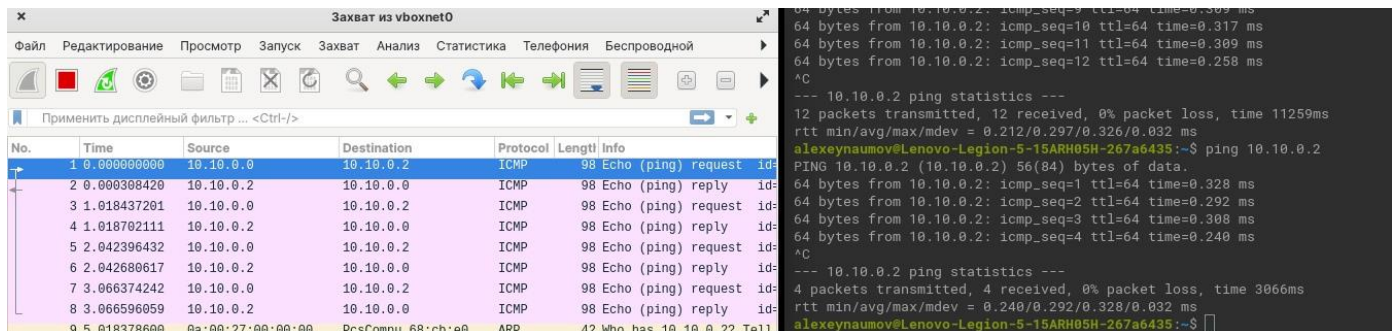
На BM Mikrotik выполняю:

```
[admin@mt-01] > ip address add address=10.10.0.3/18 interface=ether1
[admin@mt-01] > ip address print
Columns: ADDRESS, NETWORK, INTERFACE
# ADDRESS      NETWORK      INTERFACE
0 10.10.0.3/18  10.10.0.0    ether1
```

Интерфейсы сконфигурированы успешно.

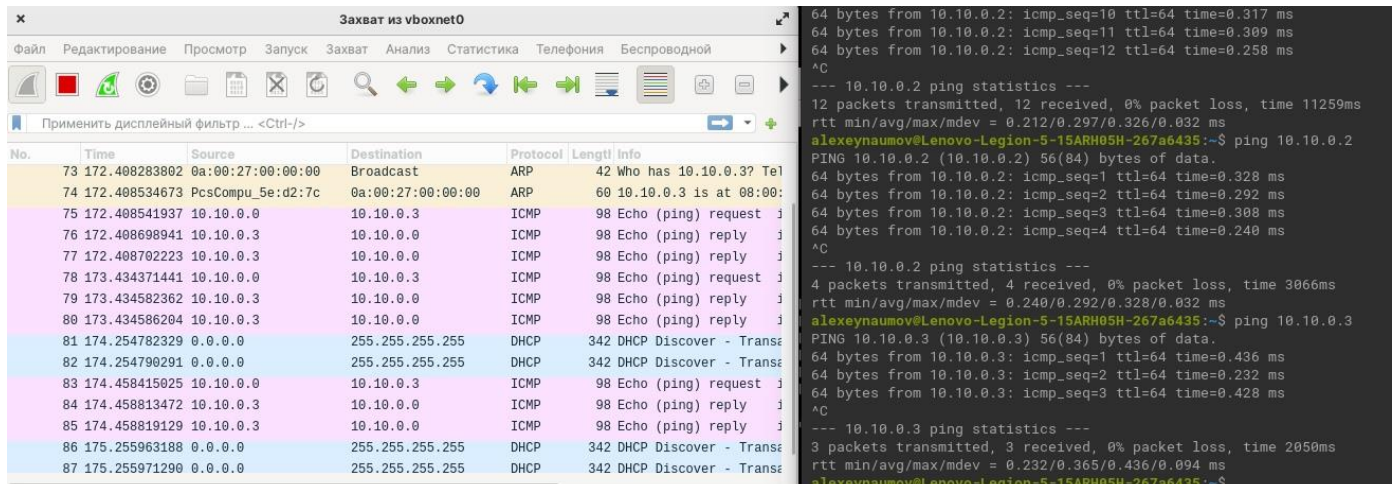
## 5. Связность хоста с BM

Настрою Wireshark на прослушивание интерфейса vboxnet0. Запущу ping с хоста до Astra Linux.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.10.0.0	10.10.0.2	ICMP	98	Echo (ping) request id=...
2	0.000398420	10.10.0.2	10.10.0.0	ICMP	98	Echo (ping) reply id=...
3	1.018437201	10.10.0.0	10.10.0.2	ICMP	98	Echo (ping) request id=...
4	1.018702111	10.10.0.2	10.10.0.0	ICMP	98	Echo (ping) reply id=...
5	2.042396432	10.10.0.0	10.10.0.2	ICMP	98	Echo (ping) request id=...
6	2.042680617	10.10.0.2	10.10.0.0	ICMP	98	Echo (ping) reply id=...
7	3.066374242	10.10.0.0	10.10.0.2	ICMP	98	Echo (ping) request id=...
8	3.066596959	10.10.0.2	10.10.0.0	ICMP	98	Echo (ping) reply id=...
9	5.018378600	0a:00:27:00:00:00	PcsCompu 68:cb:e0	ARP	42	Who has 10.10.0.2? Tell...

Сначала хост отправляет всем машинам в подсети запрос "Кто имеет адрес 10.10.0.2?", хосту отвечает BM Astra Linux **напрямую** и сообщает свой MAC адрес. Далее так же напрямую идут request/reply ICMP пакеты между хостом и BM. Затем ping до Mikrotik:

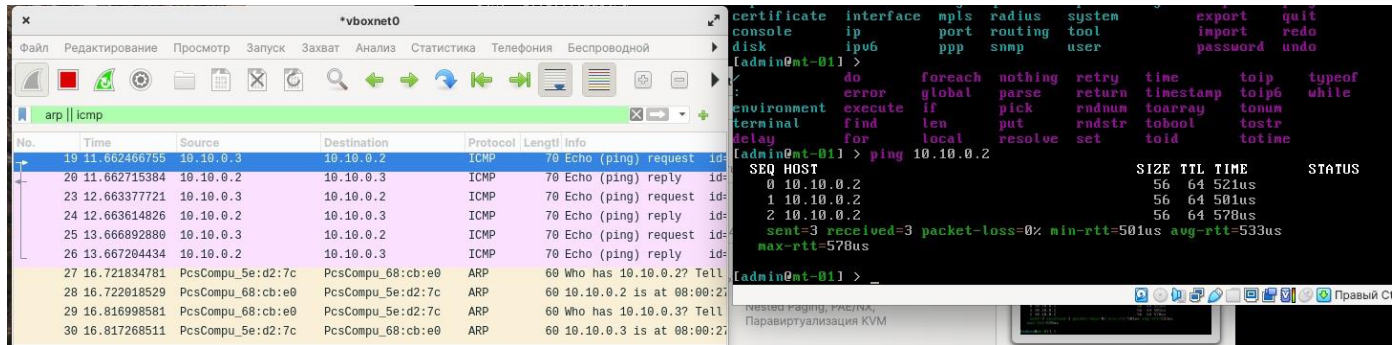


No.	Time	Source	Destination	Protocol	Length	Info
73	172.408283802	0a:00:27:00:00:00	Broadcast	ARP	42	Who has 10.10.0.3? Tell...
74	172.408534673	PcsCompu_5e:d2:7c	0a:00:27:00:00:00	ARP	60	10.10.0.3 is at 08:00:...
75	172.408541937	10.10.0.0	10.10.0.3	ICMP	98	Echo (ping) request id=...
76	172.408698941	10.10.0.3	10.10.0.0	ICMP	98	Echo (ping) reply id=...
77	172.408702223	10.10.0.0	10.10.0.3	ICMP	98	Echo (ping) request id=...
78	173.434371441	10.10.0.0	10.10.0.3	ICMP	98	Echo (ping) request id=...
79	173.434582362	10.10.0.3	10.10.0.0	ICMP	98	Echo (ping) reply id=...
80	173.434586204	10.10.0.3	10.10.0.0	ICMP	98	Echo (ping) reply id=...
81	174.254782329	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transm...
82	174.254790291	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transm...
83	174.458415025	10.10.0.0	10.10.0.3	ICMP	98	Echo (ping) request id=...
84	174.458813472	10.10.0.0	10.10.0.3	ICMP	98	Echo (ping) request id=...
85	174.458819129	10.10.0.3	10.10.0.0	ICMP	98	Echo (ping) reply id=...
86	175.255963188	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transm...
87	175.255971290	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transm...

Принцип такой же, как был описан выше.

## 6. Связность между VM

Настрою Wireshark на прослушивание интерфейса vboxnet0. Запущу ping с Mikrotik до Astra Linux.



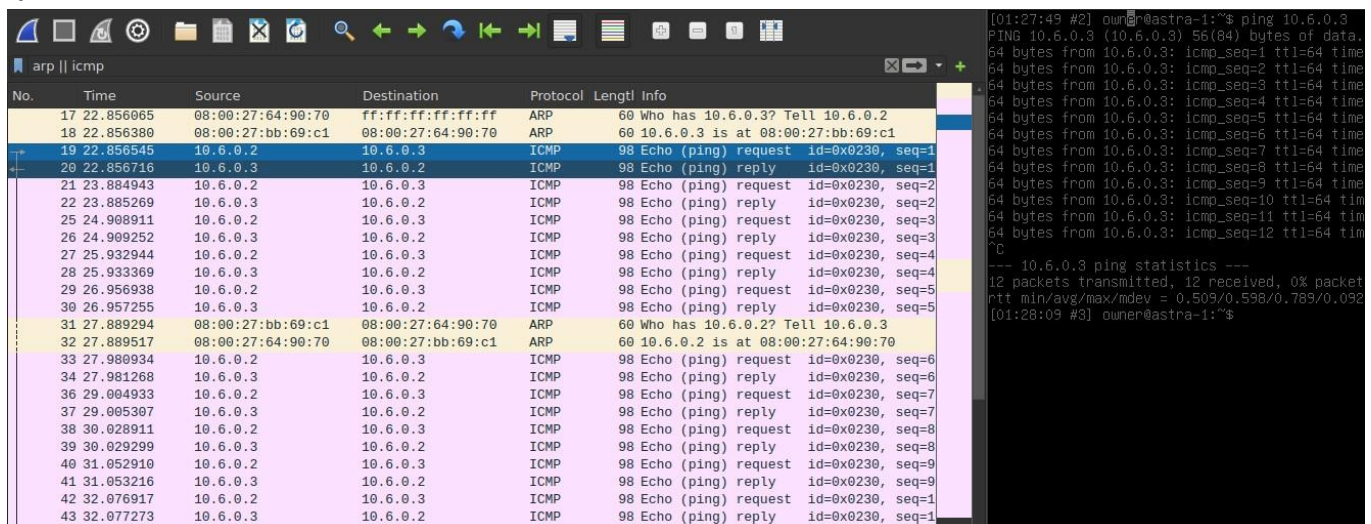
Сначала Mikrotik отправляет всем машинам в подсети запрос "Кто имеет адрес 10.6.0.2?". Поскольку VM Astra Linux сообщает свой MAC адрес ему **напрямую**, я не могу увидеть это при захвате пакетов, как и echo request/reply пакеты. Но по логу видно, что пакеты между VM принимаются и отправляются корректно.

## 7. Связность между VM, попытка номер 2

Выключу VM Astra Linux и включу логгирование пакетов на стороне гостевой машины:

```
VBoxManage modifyvm "Astra-1" --nictracel on --nictracefile1  
~/Desktop/astra-1.pcap
```

Запущу Astra Linux и пропирую Mikrotik. Затем приостановлю VM и открою .pcap файл в Wireshark.



Те самые ARP и ICMP пакеты присутствуют. Почему поток отличается от захваченного с хоста, объяснено в п.6.

Запись потока с VM может понадобится далее, поэтому запоминаю, что в конце её обязательно нужно выключить, чтобы не заполнить диск.



## 8. Адрес сети и Broadcast

Напомню, что адрес сети - 10.10.0.0. Если пропинговать его с хоста, можно увидеть следующие ARP пакеты:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	0a:00:27:00:00:00	ff:ff:ff:ff:ff:ff	ARP	42	Who has 10.6.0.0? Tell 10.6.0.1
2	1.032530058	0a:00:27:00:00:00	ff:ff:ff:ff:ff:ff	ARP	42	Who has 10.6.0.0? Tell 10.6.0.1
3	2.056450756	0a:00:27:00:00:00	ff:ff:ff:ff:ff:ff	ARP	42	Who has 10.6.0.0? Tell 10.6.0.1
4	3.080552895	0a:00:27:00:00:00	ff:ff:ff:ff:ff:ff	ARP	42	Who has 10.6.0.0? Tell 10.6.0.1
5	4.104530084	0a:00:27:00:00:00	ff:ff:ff:ff:ff:ff	ARP	42	Who has 10.6.0.0? Tell 10.6.0.1
6	5.128447994	0a:00:27:00:00:00	ff:ff:ff:ff:ff:ff	ARP	42	Who has 10.6.0.0? Tell 10.6.0.1
7	6.152565885	0a:00:27:00:00:00	ff:ff:ff:ff:ff:ff	ARP	42	Who has 10.6.0.0? Tell 10.6.0.1
8	7.176524826	0a:00:27:00:00:00	ff:ff:ff:ff:ff:ff	ARP	42	Who has 10.6.0.0? Tell 10.6.0.1
9	8.200424557	0a:00:27:00:00:00	ff:ff:ff:ff:ff:ff	ARP	42	Who has 10.6.0.0? Tell 10.6.0.1

Всем машинам в подсети с хоста отправляется запрос "Кто имеет адрес 10.10.0.0?", но так как ни одна машина не имеет такого адреса и не отвечает, то делается вывод, что конечный хост недоступен. Но что, если назначить этот адрес для интерфейса eth0 в Astra Linux?

```
[01:29:27 #10] owner@astra-1:~$ ip a show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
    link/ether 08:00:27:64:90:70 brd ff:ff:ff:ff:ff:ff
    inet 10.6.0.0/18 brd 10.6.63.255 scope global
        valid_lft forever preferred_lft forever
```

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	0a:00:27:00:00:00	ff:ff:ff:ff:ff:ff	ARP	42	Who has 10.6.0.0? Tell 10.6.0.1
2	0.000375149	08:00:27:64:90:70	0a:00:27:00:00:00	ARP	60	10.6.0.0 is at 08:00:27:64:90:70
3	0.000380589	10.6.0.1	10.6.0.0	ICMP	98	Echo (ping) request id=0x0006, seq=1/256,
4	0.000515509	10.6.0.0	10.6.0.1	ICMP	98	Echo (ping) reply id=0x0006, seq=1/256,
5	1.031860305	10.6.0.1	10.6.0.0	ICMP	98	Echo (ping) request id=0x0006, seq=2/512,
6	1.032159114	10.6.0.0	10.6.0.1	ICMP	98	Echo (ping) reply id=0x0006, seq=2/512,
7	2.055867808	10.6.0.1	10.6.0.0	ICMP	98	Echo (ping) request id=0x0006, seq=3/768,
8	2.056161188	10.6.0.0	10.6.0.1	ICMP	98	Echo (ping) reply id=0x0006, seq=3/768,
9	3.080825011	10.6.0.1	10.6.0.0	ICMP	98	Echo (ping) request id=0x0006, seq=4/1024,
10	3.081116791	10.6.0.0	10.6.0.1	ICMP	98	Echo (ping) reply id=0x0006, seq=4/1024,
11	4.103885696	10.6.0.1	10.6.0.0	ICMP	98	Echo (ping) request id=0x0006, seq=5/1280,
12	4.104169586	10.6.0.0	10.6.0.1	ICMP	98	Echo (ping) reply id=0x0006, seq=5/1280,
13	5.059580561	08:00:27:64:90:70	0a:00:27:00:00:00	ARP	60	Who has 10.6.0.1? Tell 10.6.0.0
14	5.059592511	0a:00:27:00:00:00	08:00:27:64:90:70	ARP	42	10.6.0.1 is at 0a:00:27:00:00:00
15	5.127862150	10.6.0.1	10.6.0.0	ICMP	98	Echo (ping) request id=0x0006, seq=6/1536,
16	5.128152000	10.6.0.0	10.6.0.1	ICMP	98	Echo (ping) reply id=0x0006, seq=6/1536,

Обмен пакетами происходит без каких либо проблем по стандартной схеме.

Адрес Broadcast для данной подсети - 10.10.63.255. Пропингую его с хоста:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.6.0.1	10.6.63.255	ICMP	98	Echo (ping) request id=0x0009, seq=1/256,
2	1.056891729	10.6.0.1	10.6.63.255	ICMP	98	Echo (ping) request id=0x0009, seq=2/512,
3	2.080861632	10.6.0.1	10.6.63.255	ICMP	98	Echo (ping) request id=0x0009, seq=3/768,
4	3.104867654	10.6.0.1	10.6.63.255	ICMP	98	Echo (ping) request id=0x0009, seq=4/1024,
5	4.128893617	10.6.0.1	10.6.63.255	ICMP	98	Echo (ping) request id=0x0009, seq=5/1280,
6	5.152876150	10.6.0.1	10.6.63.255	ICMP	98	Echo (ping) request id=0x0009, seq=6/1536,
7	6.177866523	10.6.0.1	10.6.63.255	ICMP	98	Echo (ping) request id=0x0009, seq=7/1792,
8	7.201888088	10.6.0.1	10.6.63.255	ICMP	98	Echo (ping) request id=0x0009, seq=8/2048,
9	8.224886412	10.6.0.1	10.6.63.255	ICMP	98	Echo (ping) request id=0x0009, seq=9/2304,
10	9.248857505	10.6.0.1	10.6.63.255	ICMP	98	Echo (ping) request id=0x0009, seq=10/2560,
11	10.272903060	10.6.0.1	10.6.63.255	ICMP	98	Echo (ping) request id=0x0009, seq=11/2816,

Во-первых, ARP запрос не посылается, т.к. адрес пингуемого хоста совпадает с широковещательным адресом. Во-вторых, на echo запросы не приходят ответные пакеты, и утилита ping интерпретирует это как 100% потерю пакетов. Назначу этот адрес для интерфейса eth0 в Astra Linux:

```
[01:32:17 #14] owner@astra-1:~$ ip a show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
    link/ether 08:00:27:64:90:70 brd ff:ff:ff:ff:ff:ff
    inet 10.6.63.255/18 brd 10.6.63.255 scope globa
        valid_lft forever preferred_lft forever
```

arp    icmp							(2): lostie@chasm:~	
No.	Time	Source	Destination	Protocol	Length	Info		
1	0.000000000	10.6.0.1	10.6.63.255	ICMP	98	Echo (ping) request id=0x000a, seq=1	lostie@chasm:~\$ ping 10.6.63.255 -b	
2	0.000396327	08:00:27:64:90:70	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.6.0.1? Tell 10.6.63.255	WARNING: pinging broadcast address	
3	0.000402297	08:00:27:64:90:70	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.6.0.1? Tell 10.6.63.255	PING 10.6.63.255 (10.6.63.255) 56(84) bytes of data.	
4	1.012771852	08:00:27:64:90:70	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.6.0.1? Tell 10.6.63.255	^C	
5	1.012777572	08:00:27:64:90:70	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.6.0.1? Tell 10.6.63.255	--- 10.6.63.255 ping statistics ---	
6	1.026577457	10.6.0.1	10.6.63.255	ICMP	98	Echo (ping) request id=0x000a, seq=2	13 packets transmitted, 0 received, 100% packet loss	
7	2.036762272	08:00:27:64:90:70	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.6.0.1? Tell 10.6.63.255	lostie@chasm:~\$	
8	2.036770232	08:00:27:64:90:70	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.6.0.1? Tell 10.6.63.255		
9	2.050563306	10.6.0.1	10.6.63.255	ICMP	98	Echo (ping) request id=0x000a, seq=3		
10	3.074573713	10.6.0.1	10.6.63.255	ICMP	98	Echo (ping) request id=0x000a, seq=4		
11	3.074901672	08:00:27:64:90:70	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.6.0.1? Tell 10.6.63.255		
12	3.074905922	08:00:27:64:90:70	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.6.0.1? Tell 10.6.63.255		
13	4.084774885	08:00:27:64:90:70	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.6.0.1? Tell 10.6.63.255		
14	4.084782375	08:00:27:64:90:70	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.6.0.1? Tell 10.6.63.255		
15	4.098558229	10.6.0.1	10.6.63.255	ICMP	98	Echo (ping) request id=0x000a, seq=5		
16	5.108784109	08:00:27:64:90:70	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.6.0.1? Tell 10.6.63.255		
17	5.108790699	08:00:27:64:90:70	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.6.0.1? Tell 10.6.63.255		

Теперь Astra посылает ARP запрос с вопросом "Кто имеет адрес 10.6.0.1 (хост)?", но не получает ответа от хоста из-за настроек безопасности и поэтому не может отправить echo reply. Поэтому результат аналогичный - 100% потеря пакетов. Вывод - узлу можно без проблем присвоить адрес сети, но присваивание ему адреса бродкаста приведёт к трудностям и проблемам.

## 9. Automatic Private IP Addressing (APIPA)

На Astra Linux временно поменяю сетевой адаптер на NAT (для доступа в WAN) и конфигурацию eth0 на получение адреса по DHCP:

```
sudo vi /etc/network/interfaces.d/eth0:
```

```
auto eth0
iface eth0 inet dhcp
    #address 10.10.0.2
    #netmask 255.255.192.0
```

```
sudo apt update
sudo apt install avahi-autoipd
```

Пакет установлен, верну виртуальный адаптер хоста.

## 10. Automatic Private IP Addressing (APIPA), продолжение

На Astra Linux временно поменяю конфигурацию eth0 на получение Zeroconf адреса:

```
sudo vi /etc/network/interfaces.d/eth0:
```

```
#auto eth0
iface eth0 inet ipv4ll
    #address 10.10.0.2
    #netmask 255.255.192.0
```

Удалю файл ~/Desktop/astra-1.pcap и перезапущу виртуальную машину. Подниму интерфейс eth0, затем приостановлю VM и открою поток в Wireshark.

```
[03:12:02 #4] owner@astra-1:~$ ip -c a show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qd
    link/ether 08:00:27:64:90:70 brd ff:ff:ff:ff:ff:ff
    inet 169.254.6.235/16 brd 169.254.255.255 scope link
    valid_lft forever preferred_lft forever
```

arp						
No.	Time	Source	Destination	Protocol	Length	Info
5	22.333762	08:00:27:64:90:70	ff:ff:ff:ff:ff:ff	ARP	60	Who has 169.254.6.235? (ARP Probe)
8	23.637766	08:00:27:64:90:70	ff:ff:ff:ff:ff:ff	ARP	60	Who has 169.254.6.235? (ARP Probe)
10	25.113754	08:00:27:64:90:70	ff:ff:ff:ff:ff:ff	ARP	60	Who has 169.254.6.235? (ARP Probe)
11	27.114212	08:00:27:64:90:70	ff:ff:ff:ff:ff:ff	ARP	60	ARP Announcement for 169.254.6.235
13	29.113761	08:00:27:64:90:70	ff:ff:ff:ff:ff:ff	ARP	60	ARP Announcement for 169.254.6.235

Сначала avahi генерирует случайный адрес 169.254.\*.\*, затем с помощью ARP опрашивает сеть, занят этот адрес или нет. Если адрес не занят, avahi занимает его и сообщает об этом другим узлам в сети (делает анонс).

На этом взаимодействие с APIPA в данной практической работе завершено, поэтому верну конфигурацию eth0 обратно на статический IP.

```
sudo vi /etc/network/interfaces.d/eth0:
```

```
auto eth0
iface eth0 inet static
    address 10.10.0.2
    netmask 255.255.192.0
```

## 11. Mikrotik DHCP Server

Зайду в веб-панель Mikrotik WebFig по адресу 10.10.0.3.

В IP \* Pool добавлю следующий пул адресов от 10.10.64.1 до 10.10.127.254 (второй из п.4). То есть пространство 10.10.0.1-10.6.63.254 будет отводиться под статические подключения, а 10.10.64.1-10.6.127.254 - под динамические.

OK	Cancel	Apply	Remove
Name	subnet2-pool		
Addresses	10.6.64.1-10.6.127.255		
Next Pool	none		

В IP \* DHCP Client выключу клиент для интерфейса ether1.

		▲ Interface	Use Peer DNS	Add Default Route	IP Address	Expires After
- E	X	ether1	yes	yes		

В IP \* DHCP Server \* Option Sets создам пустое множество настроек **subnet2-set**.

В IP \* DHCP Server \* Networks создам новую сеть 10.10.0.0/18 с маской 255.255.192.0, которую сервер должен будет сообщать клиенту.

OK
Cancel
Apply

Address
10.6.0.0/18

Gateway
▼

Netmask
▲
255.255.192.0

No DNS
☐

DNS Servers
▼

Domain
▼

WINS Servers
▼

NTP Servers
▼

CAPS Managers
▼

Next Server
▼

Boot File Name
▼

DHCP Options
▼

DHCP Option Set
▲
subnet2-set ▼

В IP \* DHCP Server создам новый сервер с именем **subnet2-server**. Укажу время аренды 2 минуты (позже увеличу), пул адресов **subnet2-pool**, множество настроек

**subnet2-set.** Также включу широковещательный режим (**Always Broadcast**).

Name	subnet2-server
Interface	ether1 ▾
Relay	▾
Lease Time	00:02:00
Bootp Lease Time	forever ▾
Address Pool	subnet2-pool ▾
DHCP Option Set	▲ subnet2-set ▾
Server Address	▾
Delay Threshold	▾
Authoritative	yes ▾
Bootp Support	static ▾
Client MAC Limit	▾
Use RADIUS	no ▾
Always Broadcast	<input checked="" type="checkbox"/>

## 12. Astra Linux DHCP Client

Запущу Astra Linux и добавлю виртуальный интерфейс eth0:1, который будет выступать DHCP клиентом.

```
sudo vi /etc/network/interfaces.d/eth0:
```

```
auto eth0
iface eth0 inet static
    address 10.10.0.2
    netmask 255.255.192.0

auto eth0:1
iface eth0:1 inet dhcp
```

Перезапущу VM. Дождусь, пока машина получит IP по DHCP. В Wireshark открою файл astra-1.pcap, в который записывается поток из VM.



dhcp						
No.	Time	Source	Destination	Protocol	Length	Info
8	15.721116	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover
10	16.228867	10.6.0.3	255.255.255.255	DHCP	342	DHCP Offer
11	16.229285	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request
12	16.229578	10.6.0.3	255.255.255.255	DHCP	342	DHCP ACK

```

Client IP address: 0.0.0.0
Your (client) IP address: 10.6.127.254
Next server IP address: 10.6.0.3

```

Как видно, подключение произошло успешно, машина получила адрес 10.10.127.254.

Через минуту Astra отправляет DHCPREQUEST на продление аренды, и сервер обновляет её, посылая DHCPACK пакет.

dhcp						
No.	Time	Source	Destination	Protocol	Length	Info
8	15.721116	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x55f96e20
10	16.228867	10.6.0.3	255.255.255.255	DHCP	342	DHCP Offer - Transaction ID 0x55f96e20
11	16.229285	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x55f96e20
12	16.229578	10.6.0.3	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 0x55f96e20
22	69.937572	10.6.0.2	10.6.0.3	DHCP	342	DHCP Request - Transaction ID 0x55f96e20
23	69.941582	10.6.0.3	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 0x55f96e20

```

▼ Dynamic Host Configuration Protocol (Request)
  Message type: Boot Request (1)
  Hardware type: Ethernet (0x01)
  Hardware address length: 6
  Hops: 0
  Transaction ID: 0x55f96e20
  Seconds elapsed: 0
  ► Bootp flags: 0x0000 (Unicast)
  Client IP address: 10.6.127.254
  Your (client) IP address: 0.0.0.0

```

В этот раз Client IP address уже не 0.0.0.0, потому что он был выдан узлу ранее.

Затем выключу Mikrotik DHCP Server и подожду 2 минуты (полное время аренды).

46	242.308348	10.6.0.2	10.6.0.3	DHCP	342	DHCP Request - Transaction ID 0x55f96e20
47	245.500245	10.6.0.2	10.6.0.3	DHCP	342	DHCP Request - Transaction ID 0x55f96e20
50	253.036258	10.6.0.2	10.6.0.3	DHCP	342	DHCP Request - Transaction ID 0x55f96e20
51	260.788170	10.6.0.2	10.6.0.3	DHCP	342	DHCP Request - Transaction ID 0x55f96e20
53	270.252227	10.6.0.2	10.6.0.3	DHCP	342	DHCP Request - Transaction ID 0x55f96e20
54	270.252585	10.6.0.3	10.6.0.2	ICMP	370	Destination unreachable (Port unreachable)
57	285.548108	10.6.0.2	10.6.0.3	DHCP	342	DHCP Request - Transaction ID 0x55f96e20
58	285.548506	10.6.0.3	10.6.0.2	ICMP	370	Destination unreachable (Port unreachable)
63	301.772069	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x53cbd776
64	308.795979	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x53cbd776
65	321.711884	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x53cbd776

Astra посылает запросы на продление, но сервер не отвечает. Поэтому текущее подключение пропадает, и Astra начинает посылать запросы к другим DHCP серверам, которые **ВОЗМОЖНО** находятся в сети.

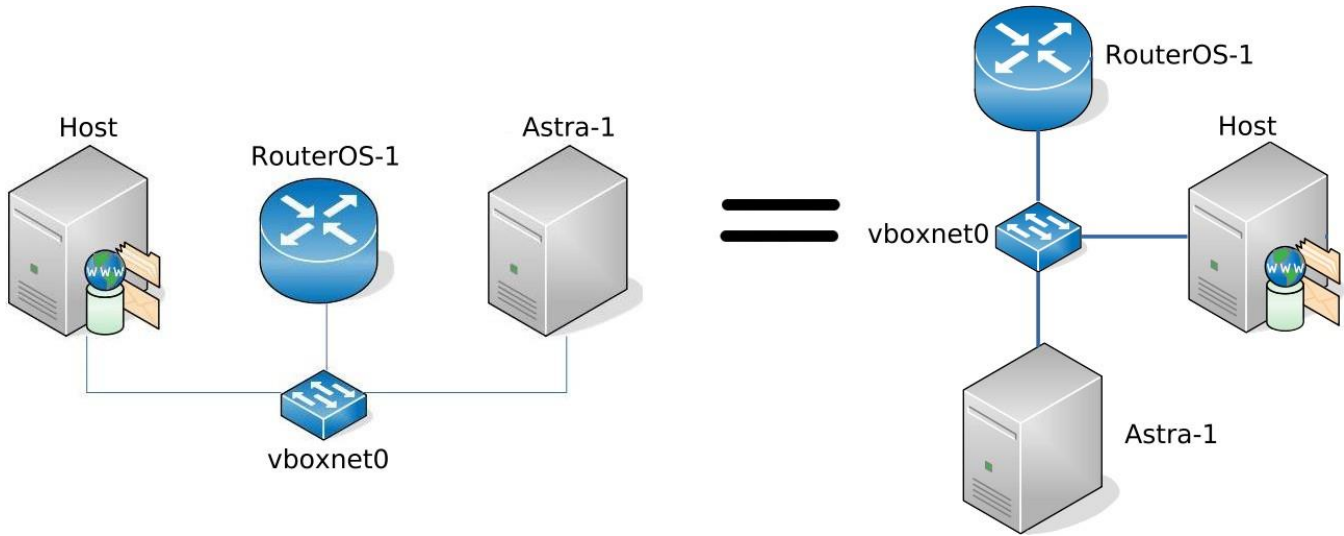
Настройка завершена, выключаю запись потока VM Astra Linux, т.к. больше она не понадобится.

```
VBoxManage modifyvm "Astra-1" --nictracel off
```

Изменяю время аренды 60 минут для DHCP сервера Mikrotik.

### 13. Обновление конфигурации сетевых интерфейсов VM

Для начала уточню текущую конфигурацию сети:



Создам дополнительные 2 виртуальных адаптера хоста: vboxnet1 (будет верхним) и vboxnet2 (будет правым).

Соответственно, для vboxnet1 отводится адресное пространство 10.10.128.0/18.

```
VBoxManage hostonlyif ipconfig vboxnet1 --ip=10.6.128.1 --  
netmask=255.255.192.0
```

Для vboxnet2 - 10.6.192.0/18.

```
VBoxManage hostonlyif ipconfig vboxnet2 --ip=10.6.192.1 --  
netmask=255.255.192.0
```

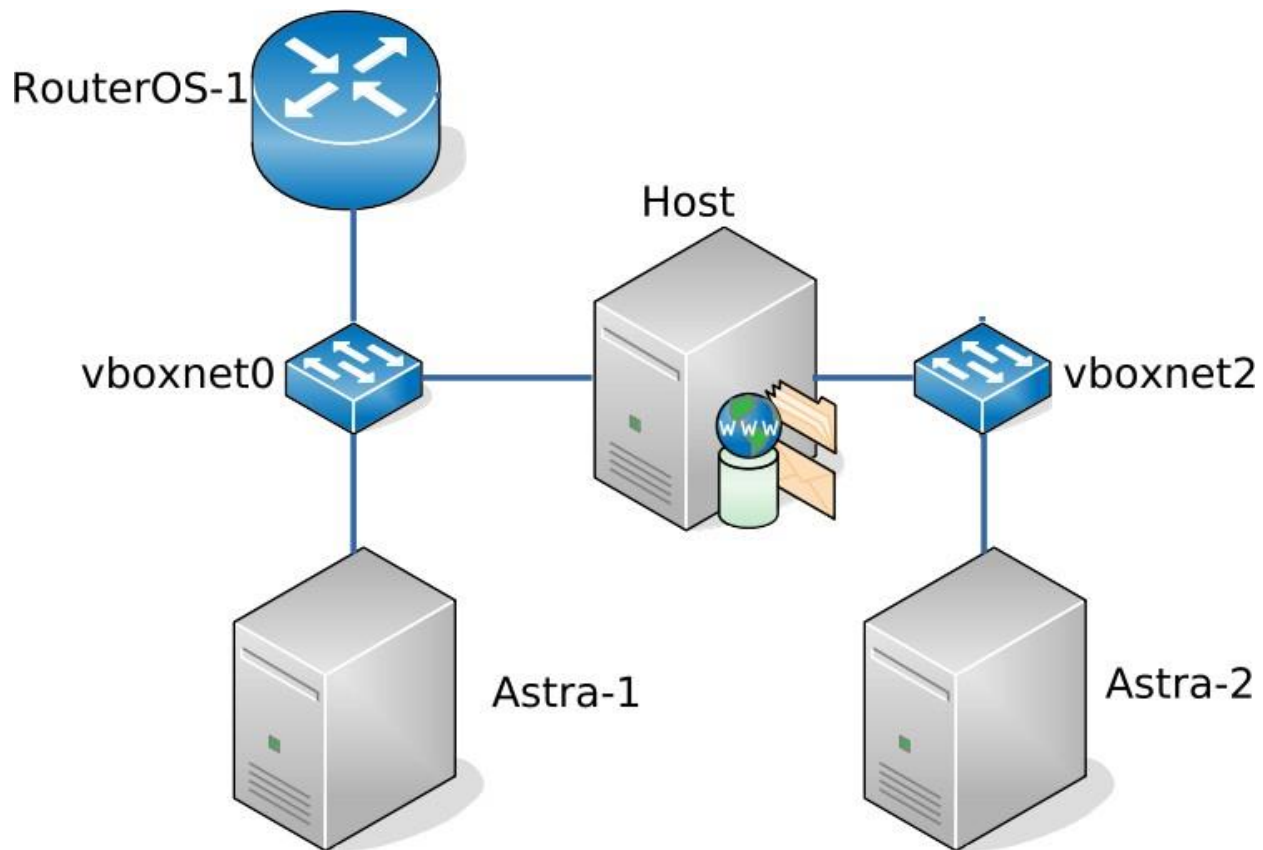
Имя	IPv4 префикс
vboxnet0	10.6.0.1/18
vboxnet1	10.6.128.1/18
vboxnet2	10.6.192.1/18

Создам VM Astra-2 (клон Astra-1) и подключу её к адаптеру vboxnet2. Сменю ей hostname на astra-2, чтобы не возникало путаницы.

```
hostnamectl set-hostname astra-2  
  
sudo vi /etc/hosts:  
...  
127.0.1.1 astra-2
```

```
...  
sudo systemctl reboot
```

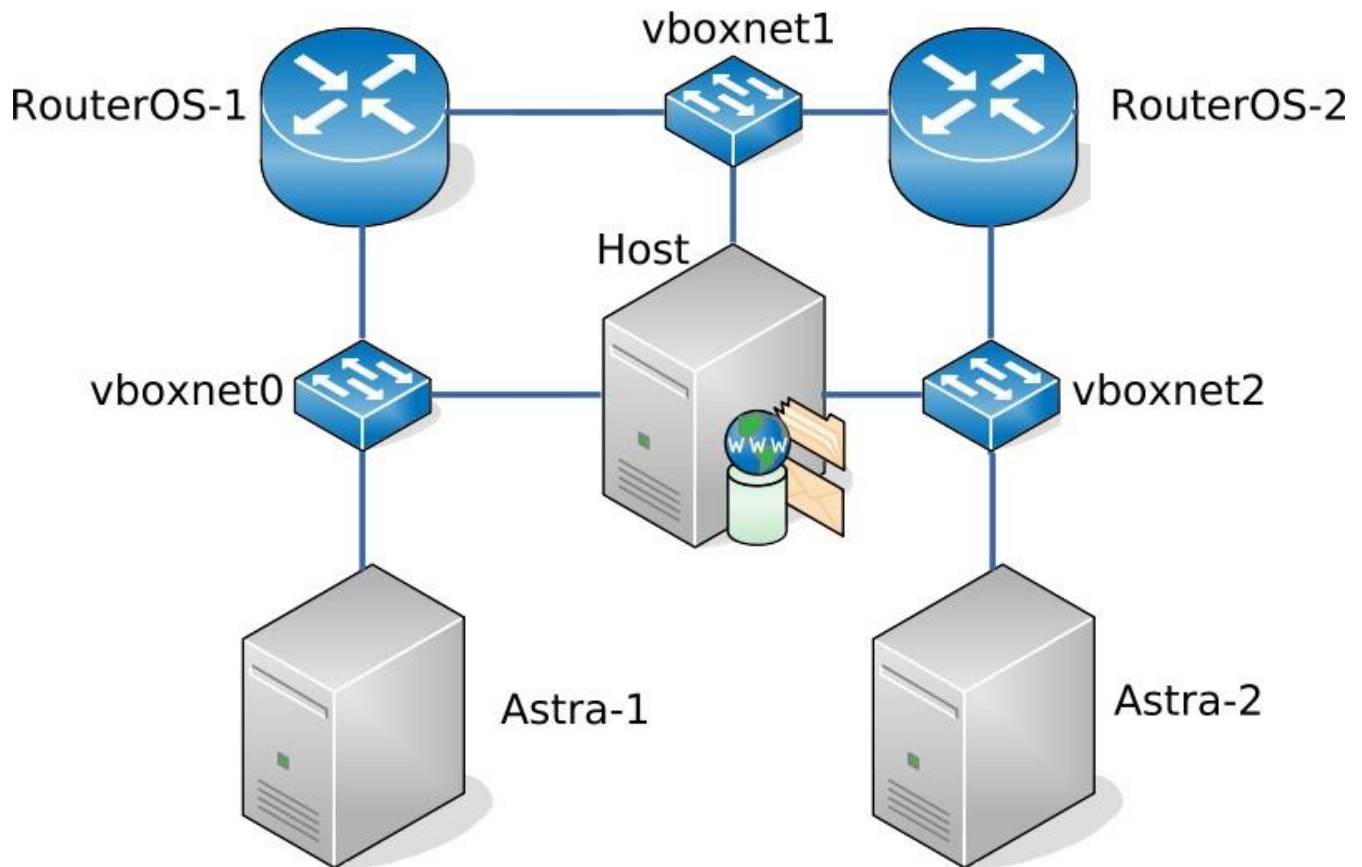
Текущая конфигурация:



Для RouterOS-1 добавлю vboxnet1 в качестве второго по счёту адаптера.  
Создам новую VM RouterOS-2 с чистой конфигурацией и подключу её к адаптерам  
vboxnet1 (первый по счёту) и vboxnet2 (второй по счёту). Сменю имя на mt-02,  
чтобы не возникало путаницы:

```
[admin@MikroTik] > system identity set name=mt-02  
[admin@mt-02] > _
```

Финальная конфигурация:



Задание выполнено.

## 14. Astra-2 DHCP

На VM Astra-2 укажу, чтобы интерфейс eth0 получал IP и настройки сети по DHCP.

```
sudo ifdown eth0

sudo vi /etc/network/interfaces.d/eth0:

auto eth0
iface eth0 inet dhcp
```

В данный момент нет смысла пытаться поднять интерфейс и получить сетевые настройки от DHCP сервера. VM никак не может их получить, т.к. не "видит" сервер. Это логично, ведь он находится в другой подсети. Чтобы исправить эту проблему, в п.15 будет создан DHCP Relay.

## 15. Связность роутеров Mikrotik

На VM RouterOS-1 обеспечу связность с RouterOS-2. Для этого выдам интерфейсу ether2 адрес 10.10.128.2/18.

```
[admin@mt-01] > ip address add interface=ether2 address=10.6.128.2/18
[admin@mt-01] > ip address print
Columns: ADDRESS, NETWORK, INTERFACE
# ADDRESS      NETWORK      INTERFACE
0 10.6.0.3/18   10.6.0.0     ether1
1 10.6.128.2/18 10.6.128.0   ether2
```

На VM RouterOS-2 обеспечу связность с RouterOS-1. Для этого выдам интерфейсу ether1 адрес 10.10.128.3/18.

```
[admin@mt-02] > ip address add interface=ether1 address=10.6.128.3/18
[admin@mt-02] > ip address print
Columns: ADDRESS, NETWORK, INTERFACE
# ADDRESS      NETWORK      INTERFACE
0 10.6.128.3/18 10.6.128.0   ether1
```

Машины пингуют друг друга, значит все сделано правильно:

```
[admin@mt-01] > ping 10.6.128.3
SEQ HOST
 0 10.6.128.3
 1 10.6.128.3
 2 10.6.128.3
 3 10.6.128.3
 4 10.6.128.3
sent=5 received=5 packet-loss=0%
max-rtt=734us
```

На VM RouterOS-2 обеспечу связность с узлами сети vboxnet2. Для этого выдам интерфейсу ether2 адрес 10.10.192.2/18.

```
[admin@mt-02] > ip address add interface=ether2 address=10.6.192.2/18
[admin@mt-02] > ip address print
Columns: ADDRESS, NETWORK, INTERFACE
# ADDRESS      NETWORK      INTERFACE
0 10.6.128.3/18 10.6.128.0   ether1
1 10.6.192.2/18 10.6.192.0   ether2
```

Также удалю все DHCP клиенты.

```
[admin@mt-02] > ip dhcp-client remove 0
[admin@mt-02] > ip dhcp-client print
[admin@mt-02] >
```

Зайду в веб-панель по адресу 10.10.192.2 и настрою DHCP Relay.

Enabled	<input checked="" type="checkbox"/>
Name	<input type="text" value="subnet4-relay"/>
Interface	<input type="text" value="ether2"/>
DHCP Server	<input type="text" value="10.6.128.2"/>



То есть, агент слушает DHCP запросы на интерфейсе ether2 (в подсети vboxnet2) и пересылает их серверу, который находится на RouterOS-1 в подсети vboxnet1.

Чтобы эти запросы от агента обрабатывались сервером, на RouterOS-1 в IP \* Pool создаю новый пул адресов **subnet4-pool** от 10.10.192.1 до 10.10.255.254.

Name	subnet4-pool
Addresses	10.6.192.1-10.6.255.254
Next Pool	none

В IP \* DHCP Server \* Networks создам новую сеть 10.10.192.0/18 с маской 255.255.192.0, которую сервер должен будет сообщать клиенту. Там же создам новый DHCP сервер:

Name	subnet4-server
Interface	ether2
Relay	10.6.192.2
Lease Time	01:00:00
Bootp Lease Time	forever
Address Pool	subnet4-pool
DHCP Option Set	subnet2-set
Server Address	
Delay Threshold	
Authoritative	yes
Bootp Support	static
Client MAC Limit	
Use RADIUS	no
Always Broadcast	<input type="checkbox"/>

Он будет слушать запросы агента с адресом 10.10.192.2 на интерфейсе ether2.

Запущу Wireshark для адаптера vboxnet2 и попробую поднять интерфейс eth0 на машине Astra-2.

dhcp						
No.	Time	Source	Destination	Protocol	Length	Info
21	17.768734786	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x5e673d28
22	17.768743416	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x5e673d28
25	18.288879868	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x5e673d28
26	18.288876188	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x5e673d28

Видно только пакеты DHCP Discover и DHCP Request к агенту. Причина - узел запрашивает конфигурацию ширококестельно, а relay отвечает конкретному узлу. Перезапузу Astra-2 в режиме захвата пакетов и повторю процедуру получения IP.

```
VBoxManage modifyvm "Astra-2" --nictrace1 on --nictracefile1
~/Desktop/astra-2.pcap
```

dhcp						
No.	Time	Source	Destination	Protocol	Length	Info
8	11.785354	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover
9	11.790520	10.6.192.2	10.6.255.254	DHCP	342	DHCP Offer
10	11.790859	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request
11	11.791644	10.6.192.2	10.6.255.254	DHCP	342	DHCP ACK

DHCP Discover от Astra-2 к RouterOS-2, затем ответ с предложением от RouterOS-2 к Astra-2.

## 16. Пакеты между роутерами

Перезапузу RouterOS-1 в режиме захвата пакетов и повторю процедуру получения IP на Astra-2.

```
VBoxManage modifyvm "RouterOS-1" --nictrace2 on --nictracefile2
~/Desktop/ros-1.pcap
```

Поток с RouterOS-1:

dhcp						
No.	Time	Source	Destination	Protocol	Length	Info
18	38.796668	10.6.128.3	10.6.128.2	DHCP	342	DHCP Discover
20	39.306317	10.6.128.2	10.6.192.2	DHCP	342	DHCP Offer
21	39.307993	10.6.128.3	10.6.128.2	DHCP	342	DHCP Request
22	39.308513	10.6.128.2	10.6.192.2	DHCP	342	DHCP ACK

С агента (10.10.128.3) к серверу приходит пакет DHCP Discover с полем:

```
Next server IP address: 0.0.0.0
Relay agent IP address: 10.6.192.2
```

С сервера на агент отправляется пакет DHCP Offer.

С агента (10.10.128.3) к серверу приходит пакет DHCP Request. С сервера на агент отправляется пакет DHCP ACK.

На Astra-2 (п.15) пакеты DHCP Offer и DHCP ACK содержат поле Next server IP address:

```
Your (client) IP address: 10.6.255.254
Next server IP address: 10.6.128.2
```

Выключу захват пакетов со всех ВМ:

```
VBoxManage modifyvm "RouterOS-1" --nictrace2 off
VBoxManage modifyvm "Astra-2" --nictracel1 off
```

Работа выполнена, сеть настроена.

## Вывод

Я познакомился со способами распределения IP адресов для создания подсетей, устройством и назначением MAC адресов, режимом ping-pong, ARP и ICMP пакетами, технологией **Automatic Private IP Addressing (APIPA)**. С их помощью сконфигурировал работоспособную сеть из машин Mikrotik и Astra Linux со статическими и динамическими конфигурациями IP адресов.