

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 №6

по дисциплине «Сети ЭВМ и телекоммуникации»

Тема: «Сетевые мосты. Виртуальные локальные сети. Протокол STP»

Выполнил: студент группы ИС-142

Наумов А.А.

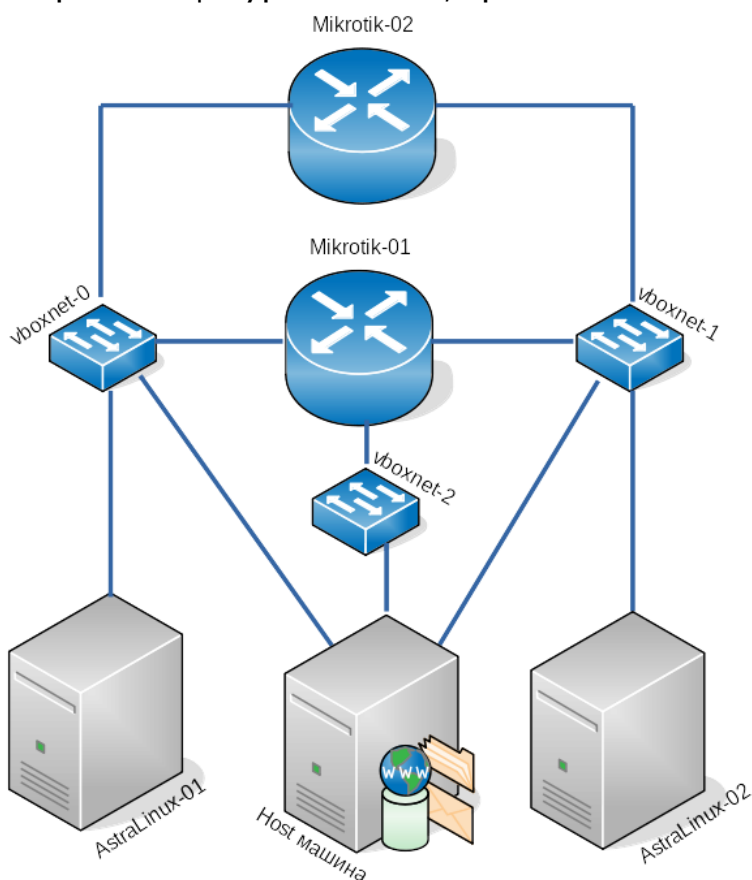
Проверил: доцент

кафедры ВС Перышкова Е.Н.

Новосибирск 2023

Задание

1. Собрать конфигурацию сети, представленной на рисунке



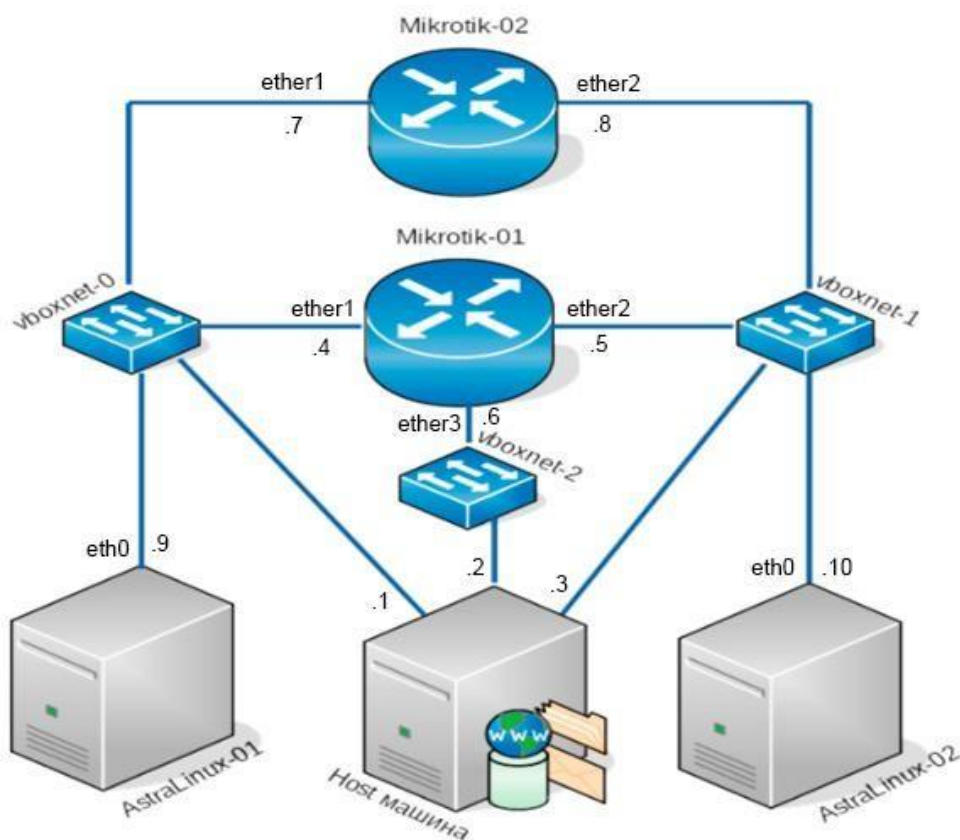
1. Предоставлена подсеть 10.10.N.0/24 (N - номер в журнале). Разделить полученный диапазон адресов на 2 равные подсети. Настроить все сетевые интерфейсы маршрутизаторов и виртуальных машин в соответствии с выбранной схемой адресации так, чтобы они использовали адреса из одной подсети. Какие интерфейсы пингуются?
2. На маршрутизаторе mt-01 объединить интерфейсы в сетевой мост. Какие интерфейсы пингуются теперь?
3. Используя Wireshark, показать, какой трафик доходит до host-машины в сети vboxnet2.
4. В маршрутизаторе mt-01 настроить VLAN с номером 2 для созданного сетевого моста. Изменить конфигурацию интерфейса с vboxnet2 так, чтобы он использовал VLAN порта с номером 2. Включить фильтрацию VLAN на сетевом мосту. Что изменилось в трафике на хост-машине в сети vboxnet2?
5. На маршрутизаторе mt-01 создать виртуальный интерфейс VLAN для созданного моста и виртуальной сети с номером 2. Назначить хост-машине и созданному виртуальному интерфейсу адреса из второй подсети. В виртуальных машинах

astra создать виртуальные интерфейсы для обработки тегированного трафика в VLAN с номером 2. Назначить этим интерфейсам адреса из второй подсети. Продемонстрировать тегированный трафик в сетях vboxnet0 и vboxnet1 и показать, что этот трафик теряет тег в сети vboxnet2.

6. На хост машине запустить Wireshark. На маршрутизаторе mt-02 объединить интерфейсы в сетевой мост с включением протокола STP. Какие порты в каком статусе? Пояснить, почему у портов стали именно такие статусы. Показать в потоке Wireshark пакеты, относящиеся к протоколу STP, и объяснить их порядок.

Выполнение работы

1. Собрана конфигурация в соответствии с заданием, выделенный диапазон разделён на 2 равные подсети: 10.10.10.0/25 и 10.10.10.128/25.



Выделен диапазон адресов 10.10.10.0/24. Чтобы разделить его на 2 подсети, фиксирую дополнительно 1 бит, таким образом пулы **допустимых** адресов для узлов:

10.10.10.1 - 10.10.10.126

10.10.10.129 - 10.10.10.254

Маска: 255.255.255.128 или /25

По заданию для машин нужно использовать адреса из одной подсети, выбираю первый пул. Прописываю в терминале команды:

**VboxManage hostonlyif ipconfig vboxnet0 --ip=10.10.10.1 --
netmask=255.255.255.128**

**VboxManage hostonlyif ipconfig vboxnet1 --ip=10.10.10.2 --
netmask=255.255.255.128**

**VboxManage hostonlyif ipconfig vboxnet2 --ip=10.10.10.3 --
netmask=255.255.255.128**

Виртуальные сети хоста				
Сети NAT				
Облачные сети				
Имя	IPv4 префикс	IPv6 префикс	DHCP сервер	
VirtualBox Host-Only Ethernet Adapter	10.10.10.1/25		Выключен	
VirtualBox Host-Only Ethernet Adapter #3	10.10.10.2/25		Выключен	
VirtualBox Host-Only Ethernet Adapter #2	10.10.10.3/25		Выключен	

Демонстрация выданных IP-адресов:

```
[admin@nt-01] > ip address print
Columns: ADDRESS, NETWORK, INTERFACE
# ADDRESS      NETWORK      INTERFACE
0 10.10.10.4/25  10.10.10.0   ether1
1 10.10.10.5/25  10.10.10.0   ether2
2 10.10.10.6/25  10.10.10.0   ether3

[admin@nt-02] > ip address print
Columns: ADDRESS, NETWORK, INTERFACE
# ADDRESS      NETWORK      INTERFACE
0 10.10.10.7/25  10.10.10.0   ether1
1 10.10.10.8/25  10.10.10.0   ether2
```

```
eth0: <BROADCAST,MULTICAST,UP,LOWER_UP>
1000
link/ether 08:00:27:f1:47:41 brd ff:ff:ff:ff:ff:ff
inet 10.10.10.9/25 brd 10.10.10.127 scope global eth0
    valid_lft forever preferred_lft
inet6 fe80::a00:27ff:fe71:4741/64 scope link
    valid_lft forever preferred_lft
owner@astral1:~$

2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP>
1000
link/ether 08:00:27:72:06:7d brd ff:ff:ff:ff:ff:ff
inet 10.10.10.10/25 brd 10.10.10.127 scope global eth0
    valid_lft forever preferred_lft
inet6 fe80::a00:27ff:fe72:067d/64 scope link
    valid_lft forever preferred_lft
owner@astral2:~$
```

Проверяем связанность между устройствами: пинг происходит только между хостом с роутерами и хостом с машинами astralinux.

Это происходит из-за проблем с таблицами маршрутизации: из-за того, что все интерфейсы находятся в одной подсети, пакеты ходят по одним и тем же маршрутам, не добираясь до получателей.

```

[admin@mt-01] > ping 10.10.10.7
  SEQ HOST                                SIZE TTL TIME                    STATUS
    0 10.10.10.7                          56 128 276us                    timeout
    1 10.10.10.7                          56 128 258us                    timeout
  sent=2 received=0 packet-loss=100%

[admin@mt-01] > ping 10.10.10.8
  SEQ HOST                                SIZE TTL TIME                    STATUS
    0 10.10.10.8                          56 128 276us                    timeout
    1 10.10.10.8                          56 128 258us                    timeout
  sent=2 received=0 packet-loss=100%

[admin@mt-01] > ping 10.10.10.1
  SEQ HOST                                SIZE TTL TIME                    STATUS
    0 10.10.10.1                          56 128 276us
    1 10.10.10.1                          56 128 258us
  sent=2 received=2 packet-loss=0% min-rtt=258us avg-rtt=267us
  max-rtt=276us


root@astral:~# ping 10.10.10.1
PING 10.10.10.1 (10.10.10.1) 56(84) bytes of data.
64 bytes from 10.10.10.1: icmp_seq=1 ttl=128 time=0.355 ms
64 bytes from 10.10.10.1: icmp_seq=2 ttl=128 time=0.139 ms
^C
--- 10.10.10.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1028ms
rtt min/avg/max/mdev = 0.139/0.247/0.355/0.108 ms

[admin@mt-01] > ping 10.10.10.10
  SEQ HOST                                SIZE TTL TIME                    STATUS
    0 10.10.10.10                          56 128 276us                    timeout
    1 10.10.10.10                          56 128 258us                    timeout
  sent=2 received=0 packet-loss=100%

[admin@mt-01] > ping 10.10.10.9
  SEQ HOST                                SIZE TTL TIME                    STATUS
    0 10.10.10.9                          56 128 276us                    timeout
    1 10.10.10.9                          56 128 258us                    timeout
  sent=2 received=0 packet-loss=100%

```

2.Объединим все интерфейсы на mt-01 в сетевой мост: создаём bridge1 во вкладке Bridge, добавляем к нему интерфейсы во вкладке Ports. Теперь роутер, объединил в себе все подключения и с него проходят пакеты ping до любого устройства. Между соседними устройствами также налажен контакт: пинг проходит от каждого до каждого устройства.

1 item				
		▲ Name	Type	L2 MTU
-	D	R  bridge1	Bridge	65535

3 items

		#	Interface	Bridge	Horiz...	Trust...	Priority (hex)	Path Cost	PVID	
-	D	0	ether2	bridge1		no	80	10	1	
-	D	1	ether1	bridge1		no	80	10	1	
-	D	2	ether3	bridge1		no	80	10	1	

[admin@nt-01] > ping 10.10.10.1

SEQ	HOST	SIZE	TTL	TIME	STATUS
0	10.10.10.1	56	128	557us	
1	10.10.10.1	56	128	258us	
sent=2 received=2 packet-loss=0% min-rtt=258us avg-rtt=407us max-rtt=557us					

[admin@nt-01] > ping 10.10.10.2

SEQ	HOST	SIZE	TTL	TIME	STATUS
0	10.10.10.2	56	128	508us	
1	10.10.10.2	56	128	267us	
sent=2 received=2 packet-loss=0% min-rtt=267us avg-rtt=387us max-rtt=508us					

[admin@nt-01] > ping 10.10.10.3

SEQ	HOST	SIZE	TTL	TIME	STATUS
0	10.10.10.3	56	128	443us	
1	10.10.10.3	56	128	259us	
sent=2 received=2 packet-loss=0% min-rtt=259us avg-rtt=351us max-rtt=443us					

[admin@nt-01] > ping 10.10.10.7

SEQ	HOST	SIZE	TTL	TIME	STATUS
0	10.10.10.7	56	64	712us	
1	10.10.10.7	56	64	300us	
sent=2 received=2 packet-loss=0% min-rtt=300us avg-rtt=506us max-rtt=712us					

[admin@nt-01] > ping 10.10.10.8

SEQ	HOST	SIZE	TTL	TIME	STATUS
0	10.10.10.8	56	64	314us	
1	10.10.10.8	56	64	308us	
sent=2 received=2 packet-loss=0% min-rtt=308us avg-rtt=311us max-rtt=314us					

[admin@nt-01] > ping 10.10.10.9

SEQ	HOST	SIZE	TTL	TIME	STATUS
0	10.10.10.9	56	64	499us	
1	10.10.10.9	56	64	363us	
sent=2 received=2 packet-loss=0% min-rtt=363us avg-rtt=431us max-rtt=499us					

```

owner@astra1:~$ ping 10.10.10.4
PING 10.10.10.4 (10.10.10.4) 56(84) bytes of data.
64 bytes from 10.10.10.4: icmp_seq=1 ttl=64 time=0.476 ms
^C
--- 10.10.10.4 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.476/0.476/0.476/0.000 ms
owner@astra1:~$ ping 10.10.10.6
PING 10.10.10.6 (10.10.10.6) 56(84) bytes of data.
64 bytes from 10.10.10.6: icmp_seq=1 ttl=64 time=0.480 ms
^C
--- 10.10.10.6 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.480/0.480/0.480/0.000 ms
owner@astra1:~$ ping 10.10.10.8
PING 10.10.10.8 (10.10.10.8) 56(84) bytes of data.
64 bytes from 10.10.10.8: icmp_seq=1 ttl=64 time=0.964 ms
64 bytes from 10.10.10.8: icmp_seq=2 ttl=64 time=0.401 ms
^C
--- 10.10.10.8 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.401/0.682/0.964/0.282 ms
owner@astra1:~$

```

3. Посмотрим в Wireshark, какие пакеты доходят до хост-машины в сети vboxnet2: это будет весь широковещательный трафик из подсетей, подключенных к сетевому мосту bridge1 на mt-01 (адресованный даже не нашей хост-машине), что не является безопасным решением: если злоумышленник получит доступ к адаптеру подсети, то сможет видеть broadcast-пакеты из других подсетей.

84	111.124460	PcsCompu_be:f2:d5	Broadcast	ARP	42 Who has 10.10.10.10? Tell 10.10.10.4
85	112.189231	PcsCompu_94:b2:6b	Spanning-tree-(for-...	STP	53 RST. Root = 32768/0/08:00:27:be:f2:
86	112.178936	PcsCompu_be:f2:d5	Broadcast	ARP	42 Who has 10.10.10.10? Tell 10.10.10.4
87	113.219058	PcsCompu_be:f2:d5	Broadcast	ARP	42 Who has 10.10.10.10? Tell 10.10.10.4
88	114.111496	PcsCompu_94:b2:6b	Spanning-tree-(for-...	STP	53 RST. Root = 32768/0/08:00:27:be:f2:
89	116.028487	PcsCompu_be:f2:d5	Broadcast	ARP	42 Who has 10.10.10.8? Tell 10.10.10.4
90	116.113605	PcsCompu_94:b2:6b	Spanning-tree-(for-...	STP	53 RST. Root = 32768/0/08:00:27:be:f2:
91	117.059252	PcsCompu_be:f2:d5	Broadcast	ARP	42 Who has 10.10.10.8? Tell 10.10.10.4
92	118.098904	PcsCompu_be:f2:d5	Broadcast	ARP	42 Who has 10.10.10.8? Tell 10.10.10.4

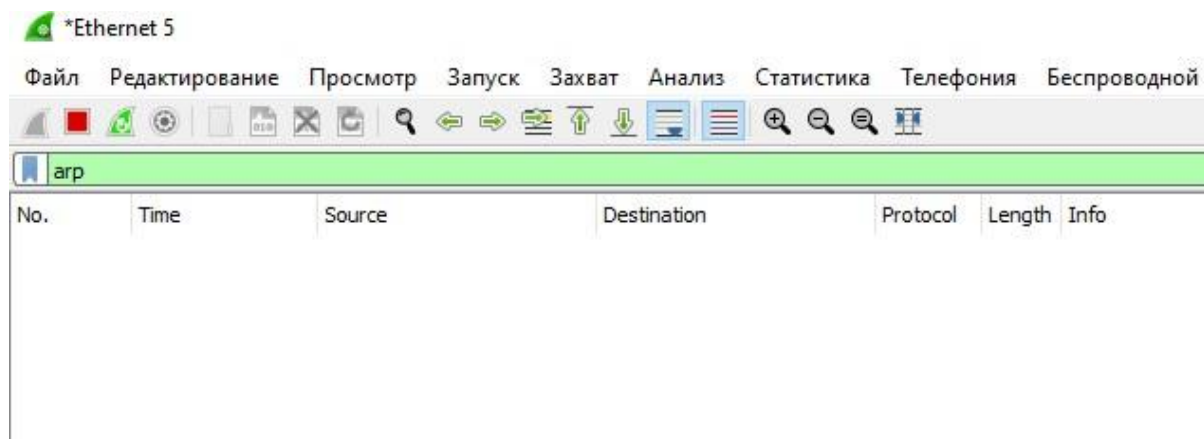
4. Для возвращения безопасности в нашу сеть настроим VLAN: в настройках bridge1 включаем опцию "VLAN Filtering" для фильтрации пакетов в виртуальных локальных сетях и на интерфейсе ether3 в меню Ports ставим значение PVID = 2. Проверяем в подменю "VLANs", создалась ли новая виртуальная локальная сеть: да, создалась

VLAN Filtering	<input checked="" type="checkbox"/>
EtherType	0x8100
PVID	1
Frame Types	admit all
Ingress Filtering	<input checked="" type="checkbox"/>

PVID	2
------	---

		▲ Bridge	VLAN IDs	Current Tagged	Current Untagged
-	D	bridge1	2		ether3
-	D	bridge1	1		bridge1, ether2, ether1

Смотрим пакеты в Wireshark: теперь ping и broadcast-пакеты не доходят до интерфейса ether3, так как он принадлежит VLAN 2, а все остальные интерфейсы - подсети VLAN 1. Для проверки наличия ARP-пакетов были проведены ping с astra1 до mt-01 ether1 и ether2. ARP-пакеты в подсети vboxnet2 на хосте отсутствуют, из-за чего также перестали проходить ping-пакеты от astra1 до хоста.



```
owner@astra1:~$ ping 10.10.10.2
PING 10.10.10.2 (10.10.10.2) 56(84) bytes of data.
^C
--- 10.10.10.2 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1008ms
```

5.Создадим виртуальный интерфейс на mt-01 во вкладке Interfaces: типом интерфейса назначим VLAN, в поле VLAN ID впишем 2, далее добавим его в Bridge->Ports к bridge1. Хосту на адаптере vboxnet2 заменим адрес на подходящий для второй подсети: 10.10.10.129, а новому виртуальному интерфейсу присвоим адрес 10.10.10.130 в меню IP -> Addresses.

Enabled	<input checked="" type="checkbox"/>
Name	<input type="text" value="vlan2"/>
Type	VLAN
MTU	<input type="text" value="1500"/>
Actual MTU	
L2 MTU	
MAC Address	
ARP	<input type="text" value="enabled"/>
ARP Timeout	
VLAN ID	<input type="text" value="2"/>
Interface	<input type="text" value="bridge1"/>

☒ Настроить адаптер вручную

IPv4 адрес:

IPv4 маска сети:

Enabled	<input checked="" type="checkbox"/>
Interface	<input type="text" value="vlan2"/>
Bridge	<input type="text" value="bridge1"/>

Enabled	<input checked="" type="checkbox"/>
Address	<input type="text" value="10.10.10.130/25"/>
Network	
Interface	<input type="text" value="vlan2"/>
Comment	<input type="text"/>

В виртуальных машинах astalinux создадим виртуальные интерфейсы для обработки тегированного трафика в VLAN-2 и назначим этим интерфейсам адреса из второй подсети (10.10.10.131 и 10.10.10.132 соответственно). Чтобы добавить интерфейс eth0.2 с привязкой к VLAN-2, пропишем "ip link add link eth0 name eth0.2 type vlan id 2". Назначим ему IP адрес в файле /etc/network/interfaces.d/eth0:

Astra1:

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_
n 1000
    link/ether 08:00:27:f1:47:41 brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.9/25 brd 10.10.10.127 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe71:4741/64 scope link
        valid_lft forever preferred_lft forever
3: eth0.2@eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
1000
    link/ether 08:00:27:f1:47:41 brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.131/25 brd 10.10.10.255 scope global eth0.2
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe71:4741/64 scope link
        valid_lft forever preferred_lft forever
```

Astra2:

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_
n 1000
    link/ether 08:00:27:72:06:7d brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.10/25 brd 10.10.10.127 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe72:067d/64 scope link
        valid_lft forever preferred_lft forever
3: eth0.2@eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
1000
    link/ether 08:00:27:72:06:7d brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.132/25 brd 10.10.10.255 scope global eth0.2
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe72:067d/64 scope link
```

Для того, чтобы перевести порты (интерфейсы) mt-01, в режим trunk (т.е. передающий пакеты с тегом VLAN) (в сетях vboxnet0 и vboxnet1 имеются устройства и с VLAN-1, и с VLAN-2, из-за чего нужно их разделять в рамках подсети), в меню VLANs добавляем новое правило для VLAN-1: ставим метку tagged на ether1 и ether2 (интерфейсах, соединённых с vboxnet0 и vboxnet1 соответственно).

Enabled	<input checked="" type="checkbox"/>
Bridge	bridge1
VLAN IDs	1
Tagged	ether1 ether2
Untagged	

Далее пробуем пинговать устройства в рамках всей нашей сети: от astra2 (10.10.10.10) к astra1 (10.10.10.131). Оба интерфейса хоста и маршрутизатора находятся в виртуальной сети VLAN-1, и смотря в пакеты, мы видим тег, соответствующий этой VLAN.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	PcsCompu_be:f2:d5	Spanning-tree-(for-	STP	53	RST. Root = 32768/0/08:00:27:be:f2:d5 Co
2	0.147566	PcsCompu_72:06:7d	Broadcast	ARP	64	Who has 10.10.10.131? Tell 10.10.10.10
3	1.162160	PcsCompu_72:06:7d	Broadcast	ARP	64	Who has 10.10.10.131? Tell 10.10.10.10
4	2.002526	PcsCompu_be:f2:d5	Spanning-tree-(for-	STP	53	RST. Root = 32768/0/08:00:27:be:f2:d5 Co
5	2.186108	PcsCompu_72:06:7d	Broadcast	ARP	64	Who has 10.10.10.131? Tell 10.10.10.10

> Frame 3: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface \Device\NPF_{28F6859A-4216-40A2-9A0D-...}

> Ethernet II, Src: PcsCompu_72:06:7d (08:00:27:72:06:7d), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

▼ 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 1

000. = Priority: Best Effort (default) (0)

...0 = DEI: Ineligible

.... 0000 0000 0001 = ID: 1

Type: ARP (0x0806)

Padding: 00000000000000000000000000000000

Trailer: 00000000

▼ Address Resolution Protocol (request)

Hardware type: Ethernet (1)

Protocol type: IPv4 (0x0800)

Смотрим в сеть vboxnet2: ARP-запросы, пришедшие от astra2, не обладают тегом, как и требуется по заданию. Это происходит из-за того, что в подсети VLAN-2 наши интерфейсы обладают меткой untagged.

6	3.739331	PcsCompu_72:06:7d	Broadcast	ARP	42	Who has 10.10.10.131? Tell 10.10.10.10
7	4.242218	PcsCompu_42:96:2d	Spanning-tree-(for-	STP	53	RST. Root = 32768/0/08:00:27:be:f2:d5 Cos
8	4.766886	PcsCompu_72:06:7d	Broadcast	ARP	42	Who has 10.10.10.131? Tell 10.10.10.10
9	5.791105	PcsCompu_72:06:7d	Broadcast	ARP	42	Who has 10.10.10.131? Tell 10.10.10.10

Frame 9: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{70CFAB61-CC0E-433C-BDAC-1...}

Ethernet II, Src: PcsCompu_72:06:7d (08:00:27:72:06:7d), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

Address Resolution Protocol (request)

Hardware type: Ethernet (1)

Protocol type: IPv4 (0x0800)

Дополнительная демонстрация тегированного + нетегированного трафика: пинг с хоста до astra1.

3392	955.714296	0a:00:27:00:00:04	Broadcast	ARP	42	Who has 10.10.10.9? Tell 10.10.10.1
3393	955.714502	0a:00:27:00:00:0c	Broadcast	ARP	64	Who has 10.10.10.9? Tell 10.10.10.3

> Frame 3393: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface \Device\NPF_{28F6859A-4216-4...}

> Ethernet II, Src: 0a:00:27:00:00:0c (0a:00:27:00:00:0c), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 1

> Address Resolution Protocol (request)

Untagged

3392	955.714296	0a:00:27:00:00:04	Broadcast	ARP	42	Who has 10.10.10.9? Tell 10.10.10.1
3393	955.714502	0a:00:27:00:00:0c	Broadcast	ARP	64	Who has 10.10.10.9? Tell 10.10.10.3

> Frame 3392: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{28F6859A-4216-4...}

> Ethernet II, Src: 0a:00:27:00:00:04 (0a:00:27:00:00:04), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

> Address Resolution Protocol (request)

6. На маршрутизаторе mt-02 объединим интерфейсы в сетевой мост с включением протокола STP. Для этого при создании сетевого моста отметим пункт "Protocol Mode" как RSTP (модифицированная версия STP, имеет меньшее время ожидания для построения дерева).

Проверим STP-статусы сетевых мостов на маршрутизаторах: зайдём в подменю Bridge и проверим параметры bridge1 на mt-02: видим, что сетевой мост маршрутизатор стал "корневым" мостом сети. На mt-01 видим, что он не считает себя "корневым", но имеет интерфейс "root-port".

Root port (корневой порт) — это порт, который имеет *минимальную стоимость* до любого порта корневого коммутатора (root bridge);

Designated port (назначенный порт) — это порт, который имеет кратчайшее расстояние от *назначенного коммутатора* до корневого коммутатора.

2 items

	#	Interface	Bridge	Horiz...	Trust...	Priority (hex)	Path Cost	PVID
- D	0	ether1	bridge1		no	80	10	1
- D	1	ether2	bridge1		no	80	10	1

Enabled ☒

Name

Type Bridge

MTU ▼

Actual MTU

L2 MTU

MAC Address

ARP

ARP Timeout ▼

Admin. MAC Address ▼

Ageing Time

IGMP Snooping ☐

DHCP Snooping ☐

Fast Forward ☒

Protocol Mode ☐ none ☐ STP ☒ RSTP ☐ MSTP

Root Bridge ☒

Root Bridge ID 8000.08:00:27:37:30:03

Regional Root Bridge ID 0.00:00:00:00:00:00

Root Path Cost 0

Root Port none

Port Count 2

Designated Port Count 2

Root Bridge ☐

Root Bridge ID 8000.08:00:27:37:30:03

Regional Root Bridge ID 0.00:00:00:00:00:00

Root Path Cost 10

Root Port ether1

Port Count 4

Designated Port Count 2

MST Config Digest

Проверим порты маршрутизаторов:

Видим, что оба порта mt-02 имеют статус designated, т.е. являются портами с наименьшей стоимостью пути до корневого моста и они будут использовать для доступа к нему.

Также у каждого порта есть состояния: у ether1 и ether2 видим отмеченными состояния Learning и Forwarding, т.е. они изучают MAC-адреса получаемых пакетов и пересылают пакеты дальше.

Ether1 mt-02:

Port Number	1
Role	designated port
Edge Port	<input type="checkbox"/>
Edge Port Discovery	<input checked="" type="checkbox"/>
Point To Point Port	<input checked="" type="checkbox"/>
External FDB	<input type="checkbox"/>
Sending RSTP	<input checked="" type="checkbox"/>
Learning	<input checked="" type="checkbox"/>
Forwarding	<input checked="" type="checkbox"/>

Ether2 mt-02:

Port Number	2
Role	designated port
Edge Port	<input type="checkbox"/>
Edge Port Discovery	<input checked="" type="checkbox"/>
Point To Point Port	<input checked="" type="checkbox"/>
External FDB	<input type="checkbox"/>
Sending RSTP	<input checked="" type="checkbox"/>
Learning	<input checked="" type="checkbox"/>
Forwarding	<input checked="" type="checkbox"/>

На mt-01 ситуация следующая: интерфейс ether1 является корневым портом, ether2 - alternate port (альтернативный корневой порт — действует как резервный для корневого (root port). Когда корневой порт заблокирован или утерян, альтернативный становится корневым), ether3 - designated port.

Ether2:

Port Number	1
Role	alternate port
Edge Port	<input type="checkbox"/>
Edge Port Discovery	<input checked="" type="checkbox"/>
Point To Point Port	<input checked="" type="checkbox"/>
External FDB	<input type="checkbox"/>
Sending RSTP	<input checked="" type="checkbox"/>
Learning	<input type="checkbox"/>
Forwarding	<input type="checkbox"/>

Ether3:

Port Number	3
Role	designated port
Edge Port	<input checked="" type="checkbox"/>
Edge Port Discovery	<input checked="" type="checkbox"/>
Point To Point Port	<input checked="" type="checkbox"/>
External FDB	<input type="checkbox"/>
Sending RSTP	<input checked="" type="checkbox"/>
Learning	<input checked="" type="checkbox"/>
Forwarding	<input checked="" type="checkbox"/>

Ether1:

Port Number	2
Role	root port
Edge Port	<input type="checkbox"/>
Edge Port Discovery	<input checked="" type="checkbox"/>
Point To Point Port	<input checked="" type="checkbox"/>
External FDB	<input type="checkbox"/>
Sending RSTP	<input checked="" type="checkbox"/>
Learning	<input checked="" type="checkbox"/>
Forwarding	<input checked="" type="checkbox"/>

Так как ether2 является альтернативным корнем, он не участвует в пересылании трафика в отличие от ether1 и ether3.

Смотрим пакеты в Wireshark: те, что относятся к протоколу STP, отправляются в сеть раз в 15 секунд (delay). В каждом таком пакете от портов можно увидеть, кто является корневым коммутатором в сети и роли, флаги самих же этих портов (learning/forwarding/...).

Пример стандартного STP-пакета приведён на скриншоте ниже.

```
8738 2286.589920 PcsCompu_37:30:03 Spanning-tree-(for... STP 53 RST. Root = 32768/0/08:00:27:37:30:03
8751 2288.600183 PcsCompu_37:30:03 Spanning-tree-(for... STP 53 RST. Root = 32768/0/08:00:27:37:30:03
8759 2290.611349 PcsCompu_37:30:03 Spanning-tree-(for... STP 53 RST. Root = 32768/0/08:00:27:37:30:03
8768 2292.620165 PcsCompu_37:30:03 Spanning-tree-(for... STP 53 RST. Root = 32768/0/08:00:27:37:30:03
8777 2294.631675 PcsCompu_37:30:03 Spanning-tree-(for... STP 53 RST. Root = 32768/0/08:00:27:37:30:03

Frame 8751: 53 bytes on wire (424 bits), 53 bytes captured (424 bits) on interface \Device\NPF_{28F6859A-4216-40A2-9A01-800000000000}
IEEE 802.3 Ethernet
Logical-Link Control
Spanning Tree Protocol
  Protocol Identifier: Spanning Tree Protocol (0x0000)
  Protocol Version Identifier: Rapid Spanning Tree (2)
  BPDU Type: Rapid/Multiple Spanning Tree (0x02)
  > BPDU flags: 0x3c, Forwarding, Learning, Port Role: Designated
  > Root Identifier: 32768 / 0 / 08:00:27:37:30:03
    Root Bridge Priority: 32768
    Root Bridge System ID Extension: 0
    Root Bridge System ID: PcsCompu_37:30:03 (08:00:27:37:30:03)
  Root Path Cost: 0
  > Bridge Identifier: 32768 / 0 / 08:00:27:37:30:03
    Port identifier: 0x8001
    Message Age: 0
    Max Age: 20
    Hello Time: 2
    Forward Delay: 15
    Version 1 Length: 0
```

Попробуем выключить-включить маршрутизатор mt-02: через STP в сеть отправляются пакеты Topology Change, т.е. информирование об изменениях в существующем дереве. Выбирается новый Root Bridge и Root Port, некоторые порты меняют свои роли. По включению роутера снова происходит Topology Change и всё возвращается в исходное состояние.

```
206 82.909394 PcsCompu_be:f2:d5 Spanning-tree-(for... STP 53 RST. TC + Root = 32768/0/08:00:27:37:30:03
207 82.909456 PcsCompu_37:30:03 Spanning-tree-(for... STP 53 RST. TC + Root = 32768/0/08:00:27:37:30:03
225 84.910043 PcsCompu_be:f2:d5 Spanning-tree-(for... STP 53 RST. TC + Root = 32768/0/08:00:27:37:30:03
226 84.910169 PcsCompu_37:30:03 Spanning-tree-(for... STP 53 RST. TC + Root = 32768/0/08:00:27:37:30:03
234 86.912516 PcsCompu_37:30:03 Spanning-tree-(for... STP 53 RST. Root = 32768/0/08:00:27:37:30:03 Co
238 88.914448 PcsCompu_37:30:03 Spanning-tree-(for... STP 53 RST. Root = 32768/0/08:00:27:37:30:03 Co
244 90.916955 PcsCompu_37:30:03 Spanning-tree-(for... STP 53 RST. Root = 32768/0/08:00:27:37:30:03 Co

Frame 226: 53 bytes on wire (424 bits), 53 bytes captured (424 bits) on interface \Device\NPF_{28F6859A-4216-40A2-9A01-800000000000}
IEEE 802.3 Ethernet
Logical-Link Control
Spanning Tree Protocol
  Protocol Identifier: Spanning Tree Protocol (0x0000)
  Protocol Version Identifier: Rapid Spanning Tree (2)
  BPDU Type: Rapid/Multiple Spanning Tree (0x02)
  > BPDU flags: 0x3d, Forwarding, Learning, Port Role: Designated, Topology Change
  > Root Identifier: 32768 / 0 / 08:00:27:37:30:03
    Root Bridge Priority: 32768
    Root Bridge System ID Extension: 0
    Root Bridge System ID: PcsCompu_37:30:03 (08:00:27:37:30:03)
  Root Path Cost: 0
  > Bridge Identifier: 32768 / 0 / 08:00:27:37:30:03
```

Все задания практической работы выполнены успешно.