

Моделирование и анализ локальной вычислительной сети

Если двум ПК необходимо взаимодействовать друг с другом, то для этого они должны использовать один и тот же набор правил. Эти правила реализуются программным обеспечением, находящимся в оперативной памяти обоих ПК или в сетевой плате ПК.

Данные правила определяют, как подключенные к сети ПК передают и принимают несущие информацию сигналы, и в каком порядке это происходит. ПО отвечает за оформление данных в виде пакетов надлежащего размера. Наряду с информацией пакеты содержат заголовки, указывающие размер пакета, его начало и место назначения.

Устройства на пути следования пакета анализируют его заголовок и определяют, куда следует направить пакет. При принятии решений о пересылке пакетов разные устройства используют различную степень детализации. Например, концентраторы не проверяют заголовки, а просто выполняют широковещательную рассылку всех полученных кадров. Между тем маршрутизаторы тщательно анализируют заголовок каждого пакета, точно определяя его место назначения.

При принятии решения о маршрутизации маршрутизаторы руководствуются своей таблицей маршрутизации. Что представляет собой эта таблица? В общем случае, это набор правил, каждое из которых содержит в себе некоторое условие, при выполнении которого пакет передается по маршруту, указанному в этом правиле. Условие, на основании выполнения которого принимается решение, это или конкретный адрес, либо адрес сети.

В терминологии сетей ТСП/IP **маской подсети** или **маской сети** называется битовая маска, определяющая, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Например, узел с IP-адресом 12.34.56.78 и маской подсети 255.255.0.0 находится в сети 12.34.0.0.

Другой вариант определения - это определение подсети IP-адресов. Например, с помощью маски подсети можно сказать, что один диапазон IP-адресов будет в одной подсети, а другой район соответственно в другой подсети.

Чтобы получить адрес сети, зная IP-адрес и маску подсети, необходимо применить к ним операцию поразрядной конъюнкции (логическое И). Например, в случае более сложной маски (битовые операции в IPv6 выглядят идентично):

IP-адрес: 11000000 10101000 00000001 00000010 (192.168.1.2)

Маска подсети: 11111111 11111111 11111111 00000000 (255.255.255.0)

Адрес сети: 11000000 10101000 00000001 00000000 (192.168.1.0)

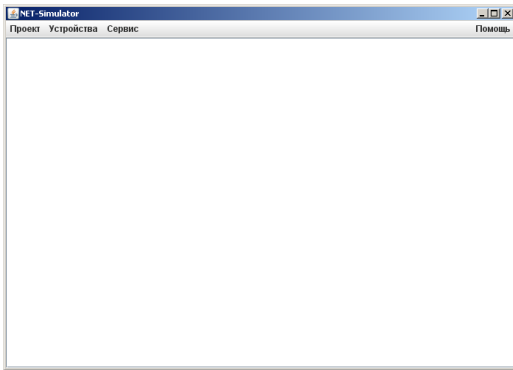
Разбиение одной большой сети на несколько маленьких подсетей позволяет упростить маршрутизацию. Например, пусть таблица маршрутизации некоторого маршрутизатора содержит следующую запись:

Сеть назначения	Маска	Адрес шлюза
192.168.1.0	255.255.255.0	192.168.15.1

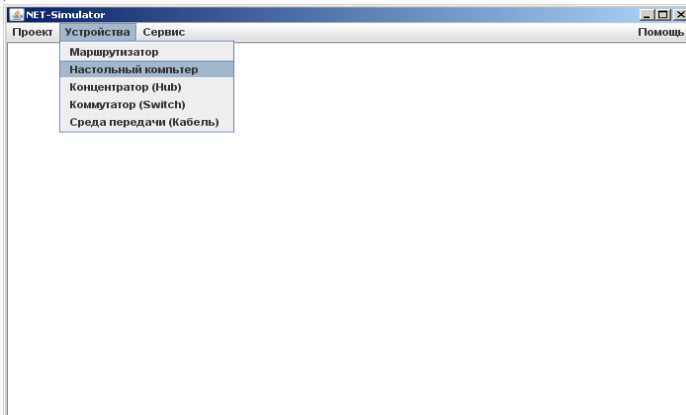
Пусть теперь маршрутизатор получает пакет данных с адресом назначения 192.168.1.2. Обработывая построчно таблицу маршрутизации, он обнаруживает, что при наложении маски 255.255.255.0 на адрес 192.168.1.2 получается адрес сети 192.168.1.0. В таблице маршрутизации этой сети соответствует шлюз 192.168.15.1, которому и отправляется пакет.

Попробуем построить несколько простых сетей с помощью программы Net-Simulator.

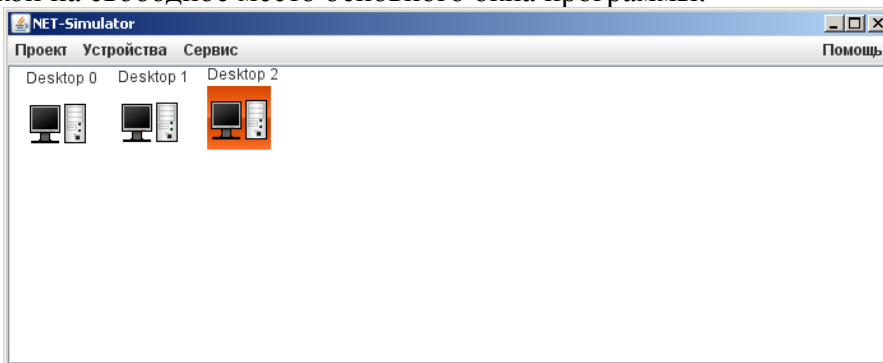
Основное окно программы выглядит таким образом:



Сперва попробуем построить простую сеть. Добавляем в наш проект несколько рабочих станций:



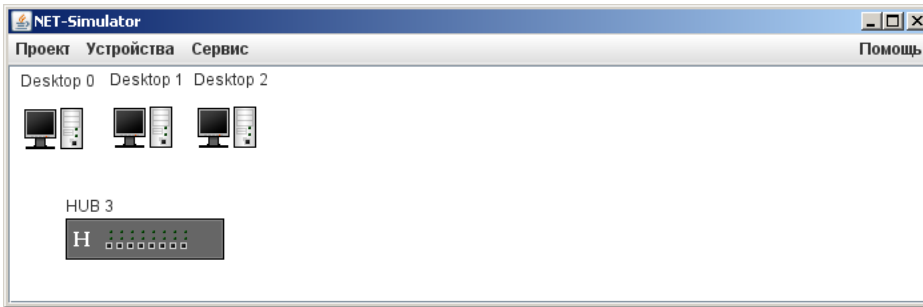
Устанавливаем в проект, например, три компьютера. Следует учесть, что новые устройства Net-Simulator добавляет в левый верхний угол окна, и их придется перетащить мышкой на свободное место основного окна программы.



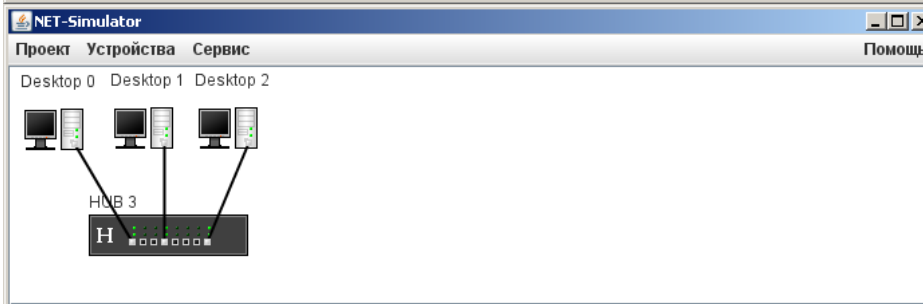
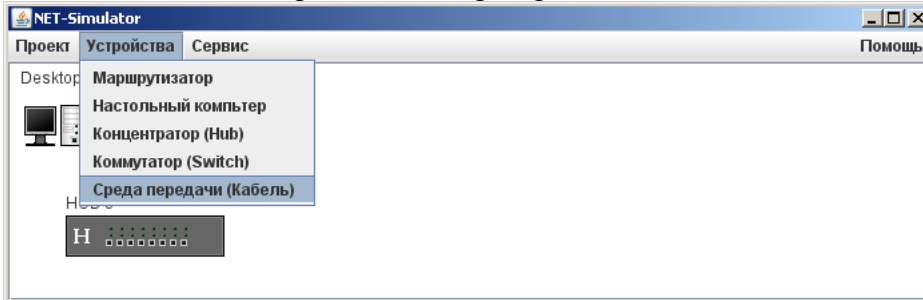
Добавляем активное сетевое оборудование, в нашем примере — коммутатор (hub)



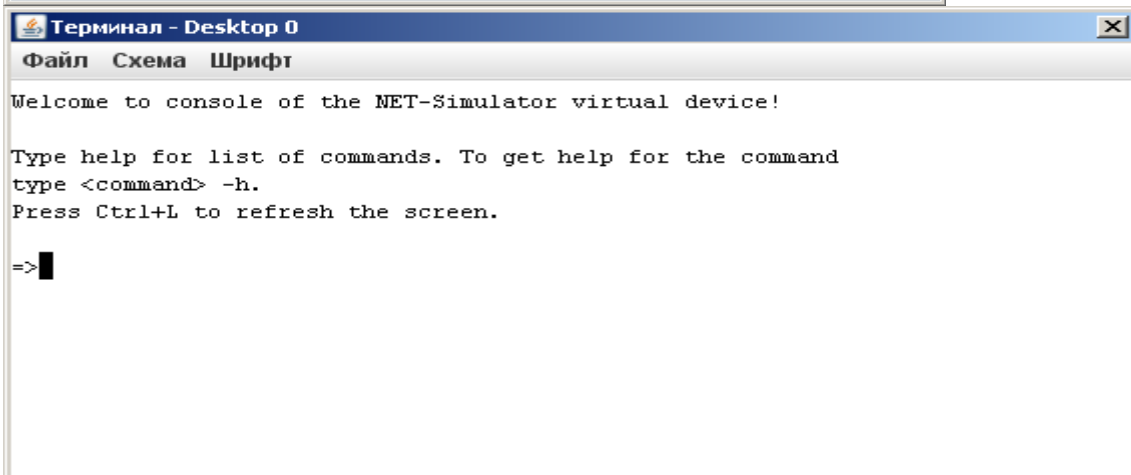
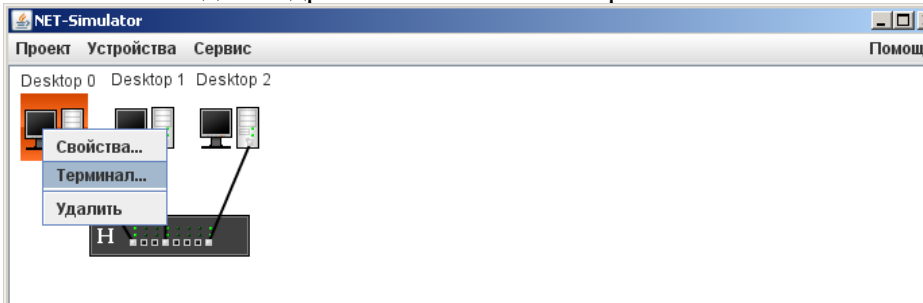
Перетаскиваем его на свободное место



Соединяем компьютеры и концентратор:



После этого задаем адреса нашим компьютерам:



Открывается окно эмуляции терминала нашего компьютера:

Выберем для нашего проекта диапазон IP 192.168.1.0 по маске 255.255.255.0

Компьютеру Desktop 0 присвоим адрес 192.168.1.1, Desktop 1 — 192.168.1.2 и Desktop2 — 192.168.1.3

```
Терминал - Desktop 1
Файл  Схема  Шрифт

Welcome to console of the NET-Simulator virtual device!

Type help for list of commands. To get help for the command
type <command> -h.
Press Ctrl+L to refresh the screen.

=>ifconfig eth0 192.168.1.2 -netmask 255.255.255.0 -up
=>
```

```
Терминал - Desktop 2
Файл  Схема  Шрифт

Welcome to console of the NET-Simulator virtual device!

Type help for list of commands. To get help for the command
type <command> -h.
Press Ctrl+L to refresh the screen.

=>ifconfig eth0 192.168.1.3 -netmask 255.255.255.0 -up
=>
```

После задания адресов мы можем проверить работоспособность построенной сети с помощью команды ping:

```
Терминал - Desktop 2
Файл  Схема  Шрифт

type <command> -h.
Press Ctrl+L to refresh the screen.

=>ifconfig eth0 192.168.1.3 -netmask 255.255.255.0 -up
=>ping 192.168.1.1
PING 192.168.1.1
64 bytes from 192.168.1.1: icmp_seq=0 ttl=63 time=203 ms
64 bytes from 192.168.1.1: icmp_seq=1 ttl=63 time=0 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=63 time=0 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=63 time=16 ms
=>
```

Таким образом, у нас получилась сеть, состоящая из трех рабочих станций, адреса которых лежат в одной подсети — 192.168.1.0 по маске 255.255.255.0, или, используя другой вид записи — 192.168.1.0/24

В данном примере мы не добавляли никаких маршрутов, потому что при присвоении адреса и маски компьютеру он автоматически добавляет в таблицу маршрутизации запись для подсети, в которой находится его адрес. Это можно увидеть с помощью команды route:

```
Терминал - Desktop 1
Файл Схема Шрифт

Welcome to console of the NET-Simulator virtual device!

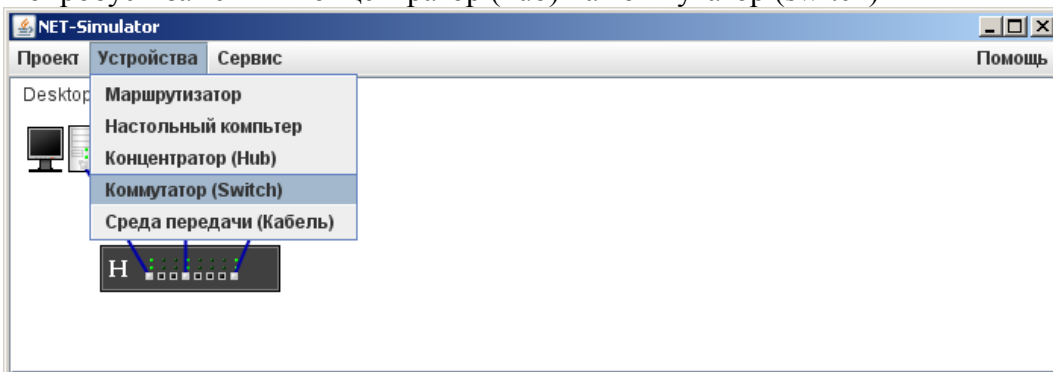
Type help for list of commands. To get help for the command
type <command> -h.
Press Ctrl+L to refresh the screen.

=>ifconfig eth0 192.168.1.2 -netmask 255.255.255.0 -up
=>route
IP routing table
Destination      Gateway          Netmask          Flags Metric Iface
192.168.1.0      *                255.255.255.0   U        1     eth0
=>
```

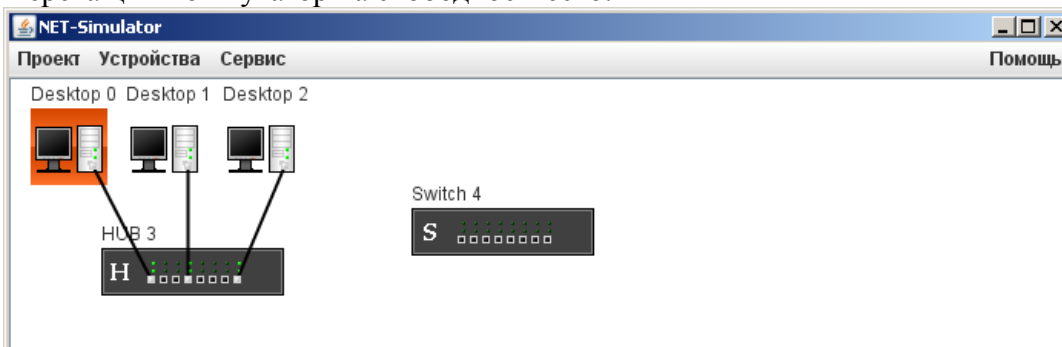
Звездочка в колонке Gateway обозначает, что для этого маршрута не используется шлюз — поскольку все пакеты, которые распространяются в пределах одной подсети передаются напрямую от отправителя к получателю.

В момент работы команды ping хорошо заметна отличительная особенность концентратора — полученный от компьютера пакет он передает на все свои порты.

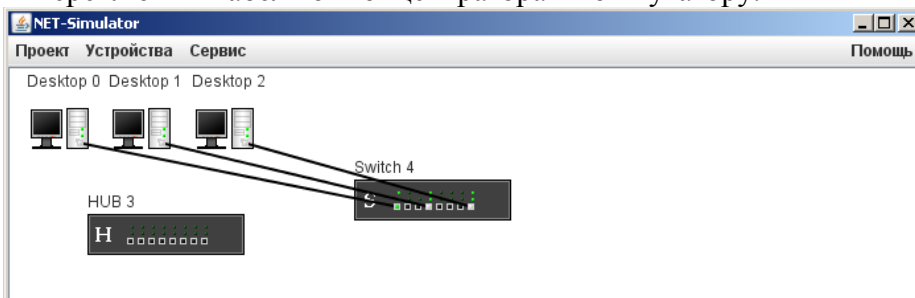
Попробуем заменить концентратор (hub) на коммутатор (switch)



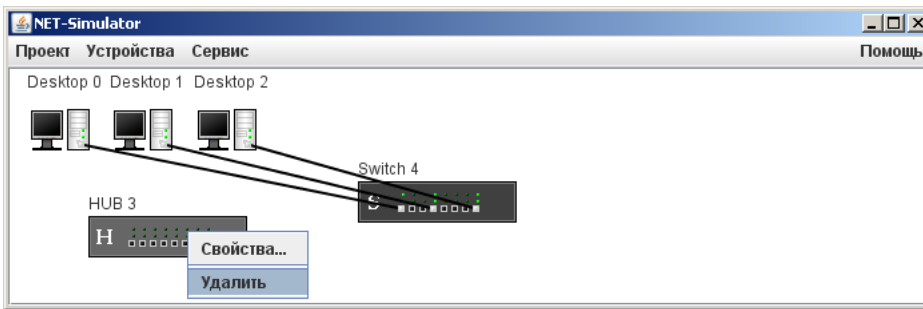
Перетащим коммутатор на свободное место:



И переключим кабели от концентратора к коммутатору:



Затем удалим концентратор:



Теперь, если мы проверим работоспособность сети командой ping, мы увидим, что в момент передачи пакета задействуются не все подключенные порты, как это было в случае с коммутатором, а только те, которые нужны для передачи пакета. Это видно по мигающим зеленым индикаторам на подключенных портах коммутатора. Таким образом, при использовании коммутаторов вместо концентраторов падает нагрузка на сеть, поскольку исключается передача пакетов, не предназначенных компьютеру. Это достигается за счет того, что коммутатор хранит в своей памяти таблицу физических адресов (MAC-address table) и при передаче пакета используя эту таблицу, передает пакет только в нужный порт. Таблицу адресов можно посмотреть в консоли концентратора, используя команду mactable:

```

Терминал - Switch 4
Файл  Схема  Шрифт

Welcome to console of the NET-Simulator virtual device!

Type help for list of commands. To get help for the command
type <command> -h.
Press Ctrl+L to refresh the screen.

=>mactable
MACAddress      port
0:0:0:0:19:0    7
0:0:0:0:13:0    0
=>

```

Эта таблица не полная, поскольку в нашем примере мы не задействовали один из компьютеров, и коммутатор еще не знает, какой у него физический адрес. Если мы откроем консоль Desktop 1 и выполним команду, например, ping 192.168.1.1, то в таблицу добавится и адрес Desktop 1:

```

Терминал - Desktop 1
Файл  Схема  Шрифт

Welcome to console of the NET-Simulator virtual device!

Type help for list of commands. To get help for the command
type <command> -h.
Press Ctrl+L to refresh the screen.

=>ifconfig eth0 192.168.1.2 -netmask 255.255.255.0 -up
=>route
IP routing table
Destination      Gateway          Netmask          Flags Metric Iface
192.168.1.0      *                255.255.255.0   U      1      eth0
=>ping 192.168.1.1
PING 192.168.1.1
64 bytes from 192.168.1.1: icmp_seq=0 ttl=63 time=203 ms
=>

```

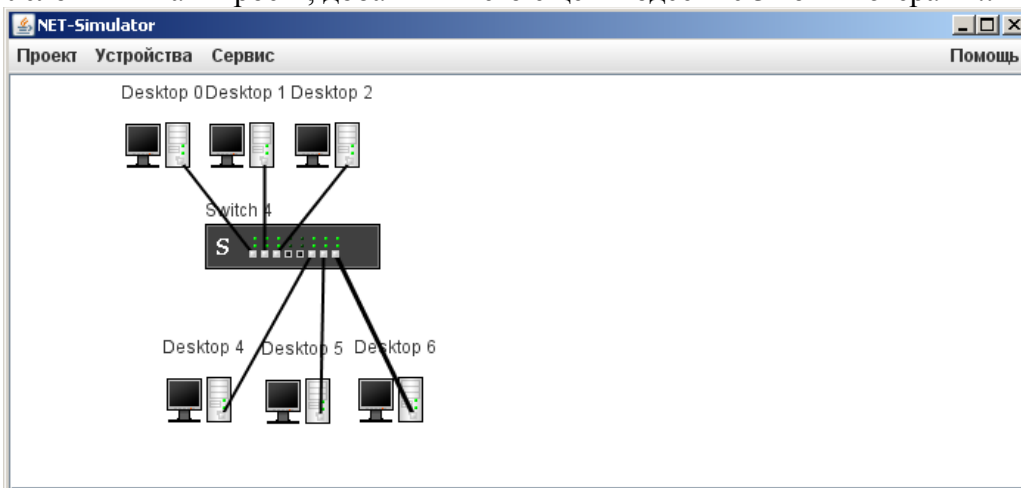
```
Терминал - Switch 4
Файл  Схема  Шрифт

Welcome to console of the NET-Simulator virtual device!

Type help for list of commands. To get help for the command
type <command> -h.
Press Ctrl+L to refresh the screen.

=>macaddress
MACAddress    port
0:0:0:0:19:0  7
0:0:0:0:13:0  0
=>macaddress
MACAddress    port
0:0:0:0:19:0  7
0:0:0:0:13:0  0
=>macaddress
MACAddress    port
0:0:0:0:19:0  7
0:0:0:0:13:0  0
0:0:0:0:D:0   3
=>
```

Усложним наш проект, добавив в него еще 1 подсеть с 3 компьютерами.:



Установим на них адреса из диапазона 172.16.1.0/24, то есть 172.16.1.0 по маске 255.255.255.0

```
Терминал - Desktop 4
Файл  Схема  Шрифт

Welcome to console of the NET-Simulator virtual device!

Type help for list of commands. To get help for the command
type <command> -h.
Press Ctrl+L to refresh the screen.

=>ifconfig eth0 172.16.1.1 -netmask 255.255.255.0 -up
=>
```

Компьютерам Desktop 5 и Desktop 6 назначим адреса 172.16.1.2 и 172.16.1.3 соответственно.

Проверим доступность компьютеров подсети 172.16.1.0/24:

```
Терминал - Desktop 4
Файл  Схема  Шрифт

Welcome to console of the NET-Simulator virtual device!

Type help for list of commands. To get help for the command
type <command> -h.
Press Ctrl+L to refresh the screen.

=>ifconfig eth0 172.16.1.1 -netmask 255.255.255.0 -up
=>ping 172.16.1.3
PING 172.16.1.3
64 bytes from 172.16.1.3: icmp_seq=0 ttl=63 time=203 ms
64 bytes from 172.16.1.3: icmp_seq=1 ttl=63 time=15 ms
64 bytes from 172.16.1.3: icmp_seq=2 ttl=63 time=0 ms
=>
```

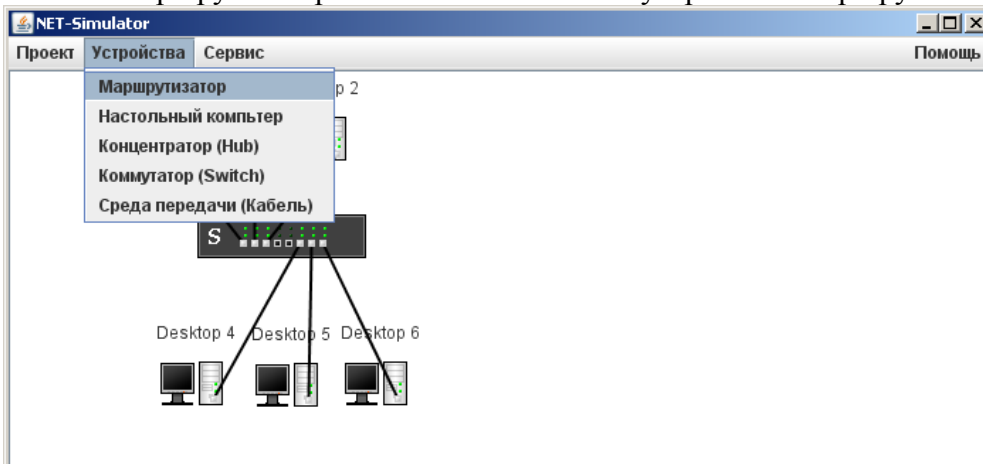
Проверим, доступны ли компьютеры второй подсети компьютерам первой:

```
Терминал - Desktop 0
Файл  Схема  Шрифт

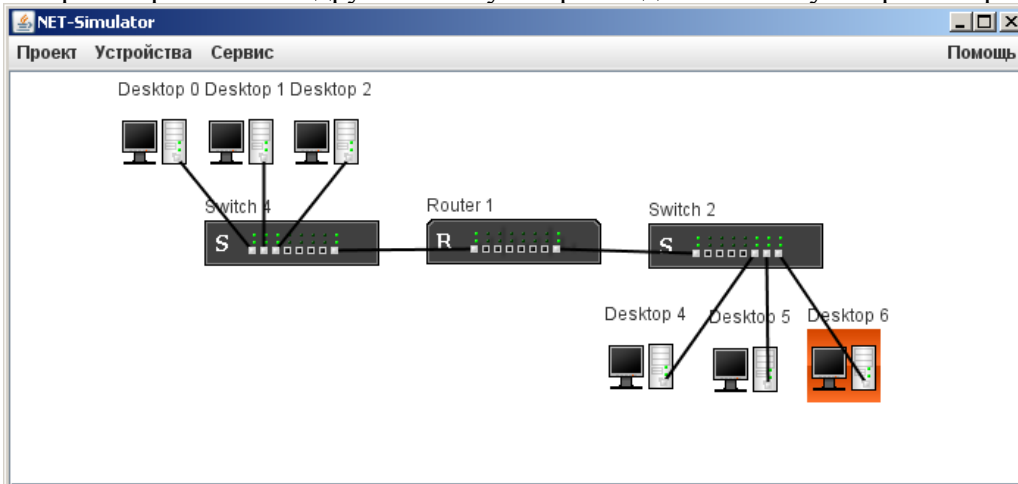
=>ifconfig eth0 192.168.1.1 -netmask 255.255.255.0 -up
=>route
IP routing table
Destination      Gateway          Netmask          Flags Metric Iface
=>route
IP routing table
Destination      Gateway          Netmask          Flags Metric Iface
192.168.1.0      *                255.255.255.0   U        1      eth0
=>ping 172.16.1.1
PING 172.16.1.1
icmp_seq=0 Destination Host Unreachable
icmp_seq=1 Destination Host Unreachable
icmp_seq=2 Destination Host Unreachable
=>
```

Таким образом мы видим, что хотя компьютеры первой и второй подсети подключены к одному и тому же устройству, между двумя подсетями пакеты не передаются. Это связано с тем, что в таблицах маршрутизации компьютеров нет информации о том, куда передавать пакеты, чтобы они попали в другую подсеть.

Для маршрутизации пакетов между подсетями используются аппаратные или программные маршрутизаторы. В Net-Simulator это устройство-маршрутизатор:



Добавим в наш проект маршрутизатор и еще один коммутатор. Переключим компьютеры второй сети на другой коммутатор. Соединим коммутаторы с маршрутизатором:



Установим на маршрутизаторе адреса: 192.168.1.254 для нулевого порта (eth0) и 172.16.1.254 для восьмого (eth7). Проверим на нем таблицу маршрутизации:

```

Терминал - Router 1
Файл Схема Шрифт
Welcome to console of the NET-Simulator virtual device!

Type help for list of commands. To get help for the command
type <command> -h.
Press Ctrl+L to refresh the screen.

=>ifconfig eth0 192.168.1.254 -netmask 255.255.255.0 -up
=>ifconfig eth7 172.16.1.254 -netmask 255.255.255.0 -up
=>route
IP routing table
Destination    Gateway      Netmask      Flags Metric Iface
172.16.1.0     *           255.255.255.0 U    1     eth7
192.168.1.0    *           255.255.255.0 U    1     eth0
=>

```

Теперь для каждого компьютера мы должны добавить в таблицу маршрутизации правило для пакетов, адрес назначения которых лежит в другой подсети. Например, для компьютера Desktop 0:

```

Терминал - Desktop 0
Файл Схема Шрифт
icmp_seq=0 Destination Host Unreachable
icmp_seq=1 Destination Host Unreachable
icmp_seq=2 Destination Host Unreachable
=>route add 172.16.1.0 -netmask 255.255.255.0 -gw 192.168.1.254
IP routing table
Destination    Gateway      Netmask      Flags Metric Iface
192.168.1.0    *           255.255.255.0 U    1     eth0
=>route -add 172.16.1.0 -netmask 255.255.255.0 -gw 192.168.1.254
=>route
IP routing table
Destination    Gateway      Netmask      Flags Metric Iface
172.16.1.0     192.168.1.254 255.255.255.0 UG    0     eth0
192.168.1.0    *           255.255.255.0 U    1     eth0
=>

```

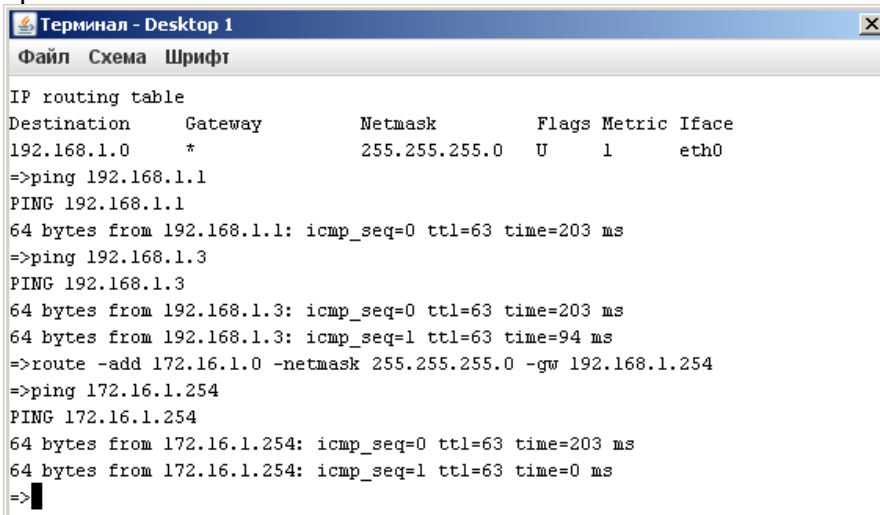
Итак, проверив таблицу маршрутизации мы видим в ней два правила:

- 1) Все пакеты, которые идут в адресное пространство подсети 172.16.1.0/24 отправлять по адресу 192.168.1.254

- 2) Все пакеты, которые идут в адресное пространство 192.168.1.0/24 отправлять напрямую, не используя шлюз, поскольку адрес Desktop 0 находится в этом же адресном пространстве.

Повторим команду `route -add 172.16.1.0 -netmask 255.255.255.0 -gw 192.168.1.254` для Desktop 1 и Desktop 2

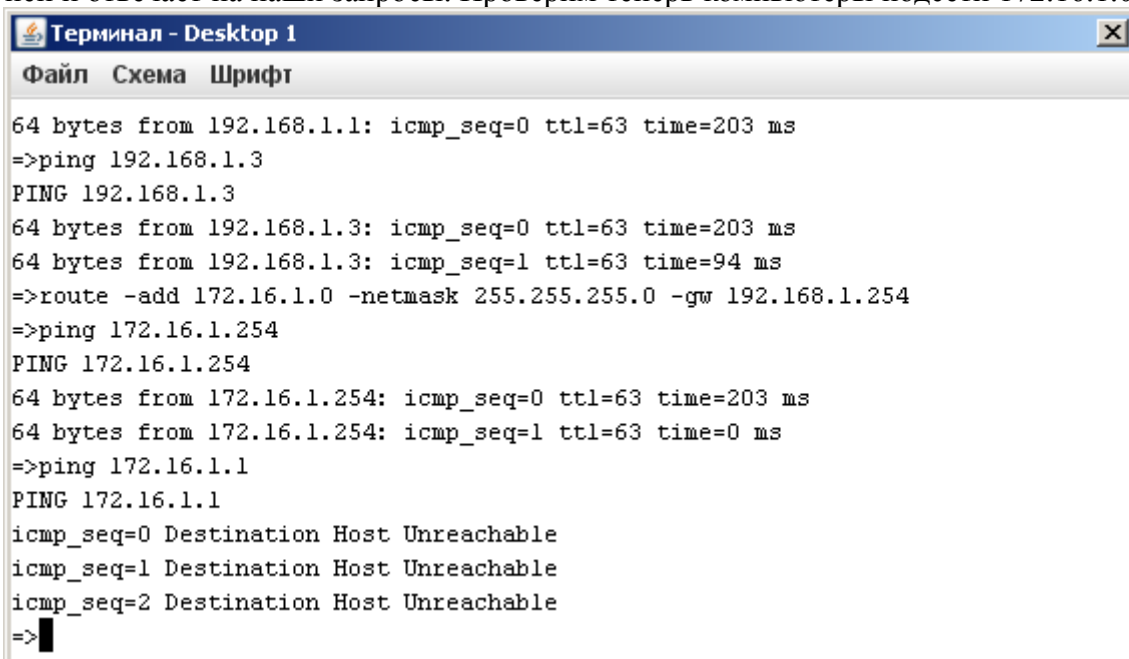
Проверим, доступны ли адреса подсети 172.16.1.0/24 от, например, компьютера Desktop1:



```
Терминал - Desktop 1
Файл  Схема  Шрифт

IP routing table
Destination      Gateway          Netmask          Flags Metric Iface
192.168.1.0      *                255.255.255.0   U        1     eth0
=>ping 192.168.1.1
PING 192.168.1.1
64 bytes from 192.168.1.1: icmp_seq=0 ttl=63 time=203 ms
=>ping 192.168.1.3
PING 192.168.1.3
64 bytes from 192.168.1.3: icmp_seq=0 ttl=63 time=203 ms
64 bytes from 192.168.1.3: icmp_seq=1 ttl=63 time=94 ms
=>route -add 172.16.1.0 -netmask 255.255.255.0 -gw 192.168.1.254
=>ping 172.16.1.254
PING 172.16.1.254
64 bytes from 172.16.1.254: icmp_seq=0 ttl=63 time=203 ms
64 bytes from 172.16.1.254: icmp_seq=1 ttl=63 time=0 ms
=>
```

Итак, мы видим, что порт маршрутизатора, который подключен в соседнюю сеть доступен и отвечает на наши запросы. Проверим теперь компьютеры подсети 172.16.1.0/24.



```
Терминал - Desktop 1
Файл  Схема  Шрифт

64 bytes from 192.168.1.1: icmp_seq=0 ttl=63 time=203 ms
=>ping 192.168.1.3
PING 192.168.1.3
64 bytes from 192.168.1.3: icmp_seq=0 ttl=63 time=203 ms
64 bytes from 192.168.1.3: icmp_seq=1 ttl=63 time=94 ms
=>route -add 172.16.1.0 -netmask 255.255.255.0 -gw 192.168.1.254
=>ping 172.16.1.254
PING 172.16.1.254
64 bytes from 172.16.1.254: icmp_seq=0 ttl=63 time=203 ms
64 bytes from 172.16.1.254: icmp_seq=1 ttl=63 time=0 ms
=>ping 172.16.1.1
PING 172.16.1.1
icmp_seq=0 Destination Host Unreachable
icmp_seq=1 Destination Host Unreachable
icmp_seq=2 Destination Host Unreachable
=>
```

В этом случае, если мы обратим внимание коммутаторы и маршрутизатор, мы видим, что сетевая активность есть. А не отвечает компьютер 172.16.1.1 только потому, что он не знает, куда отправлять ответ. А запрос к нему приходит, это заметно по активности портов.

Используем другой способ задания маршрутизации на компьютере Desktop 4. Как мы видим, обмен пакетами между компьютерами Desktop 4, Desktop 5 и Desktop 6 — это прямая передача данных в сети, а обмен с любым другим компьютером можно осуществить ТОЛЬКО через маршрутизатор Router 2, который подключен к сети Desktop 4 Desktop 5 Desktop 6 портом, который имеет адрес 172.16.1.254. Итак, зададим компьютерам в сети 172.16.1.0 ШЛЮЗ ПО УМОЛЧАНИЮ (шлюз, через который будут отправляться пакеты, для которых нет записей в таблице маршрутизации)

```
Терминал - Desktop 4
Файл  Схема  Шрифт
=>route -add 0.0.0.0 -netmask 0.0.0.0 -gw 172.16.1.254
=>
```

Таким образом, таблица маршрутизации для Desktop 4 выглядит следующим образом:

```
Терминал - Desktop 4
Файл  Схема  Шрифт
=>route
IP routing table
Destination      Gateway          Netmask          Flags Metric Iface
172.16.1.0       *                255.255.255.0   U      1      eth0
0.0.0.0          172.16.1.254    0.0.0.0          UG     0      eth0
=>
=>
```

То есть, если наша сеть имеет только один выход за пределы ее диапазона (172.16.1.0 по маске 255.255.255.0) то мы можем не задавать маршруты ко всем внешним подсетям, а задать только один маршрут по умолчанию.

Проверим доступность Desktop 4 со стороны Desktop 1:

```
Терминал - Desktop 1
Файл  Схема  Шрифт
=>
=>ping 172.16.1.1
PING 172.16.1.1
64 bytes from 172.16.1.1: icmp_seq=0 ttl=62 time=422 ms
64 bytes from 172.16.1.1: icmp_seq=1 ttl=62 time=0 ms
64 bytes from 172.16.1.1: icmp_seq=2 ttl=62 time=0 ms
64 bytes from 172.16.1.1: icmp_seq=3 ttl=62 time=15 ms
64 bytes from 172.16.1.1: icmp_seq=4 ttl=62 time=0 ms
=>
```

Вот теперь ответы приходят, потому что компьютер Desktop 4 теперь знает, куда отправлять ответ.

Зададим маршруты по умолчанию для Desktop 5 и Desktop 6:

```
Терминал - Desktop 5
Файл  Схема  Шрифт
Welcome to console of the NET-Simulator virtual device!

Type help for list of commands. To get help for the command
type <command> -h.
Press Ctrl+L to refresh the screen.

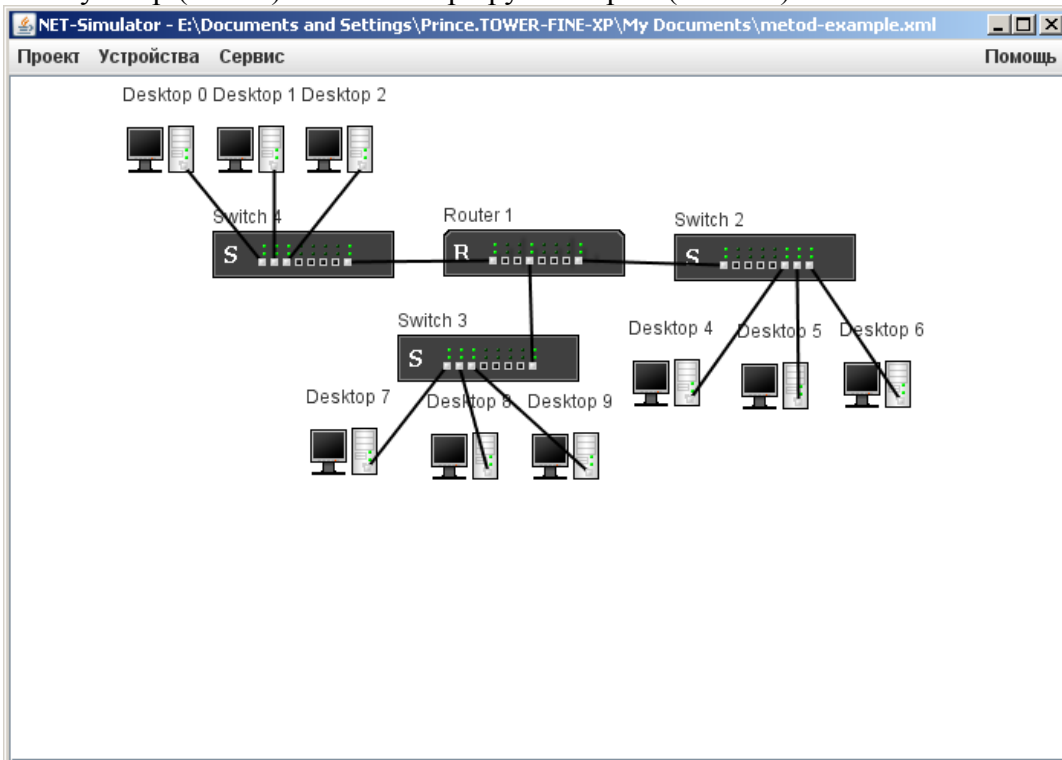
=>route -add 0.0.0.0 -netmask 0.0.0.0 -gw 172.16.1.254
=>route
IP routing table
Destination      Gateway          Netmask          Flags Metric Iface
172.16.1.0       *                255.255.255.0   U      1      eth0
0.0.0.0          172.16.1.254    0.0.0.0          UG     0      eth0
=>
```

```
Терминал - Desktop 6
Файл  Схема  Шрифт

type <command> -h.
Press Ctrl+L to refresh the screen.

=>route -add 0.0.0.0 -netmask 0.0.0.0 -gw 172.16.1.254
=>ping 192.168.1.1
PING 192.168.1.1
64 bytes from 192.168.1.1: icmp_seq=0 ttl=62 time=406 ms
64 bytes from 192.168.1.1: icmp_seq=1 ttl=62 time=16 ms
=>
```

Теперь создадим другую сеть, например, 10.10.10.0 по маске 255.255.255.0 соединим их через коммутатор (switch) с нашим маршрутизатором (Router1)



Укажем маршрутизатору IP адрес 10.10.10.254 и проверим таблицу маршрутизации:

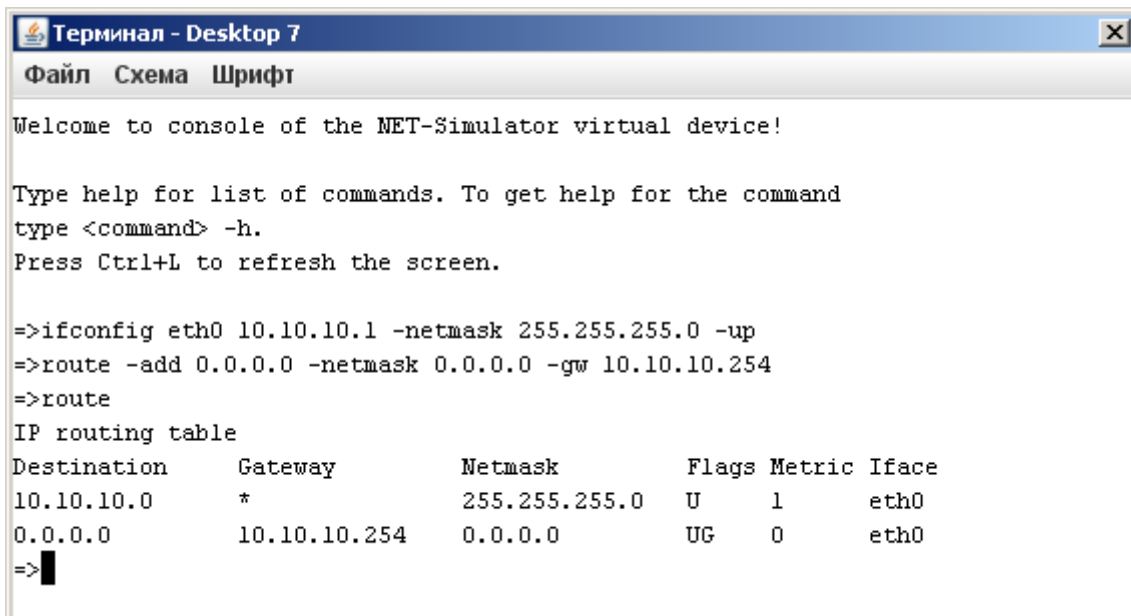
```
Терминал - Router 1
Файл  Схема  Шрифт

Welcome to console of the NET-Simulator virtual device!

Type help for list of commands. To get help for the command
type <command> -h.
Press Ctrl+L to refresh the screen.

=>ifconfig eth3 10.10.10.254 -netmask 255.255.255.0 -up
=>route
IP routing table
Destination      Gateway          Netmask          Flags Metric Iface
10.10.10.0       *                255.255.255.0   U      1      eth3
172.16.1.0       *                255.255.255.0   U      1      eth7
192.168.1.0      *                255.255.255.0   U      1      eth0
=>
```

Настроим компьютерам Desktop 7 Desktop 8 и Desktop 9 адреса 10.10.10.1 10.10.10.2 и 10.10.10.3 соответственно. Мы приводим пример для Desktop 7, а компьютеры Desktop 8 и Desktop 9 настраиваются совершенно аналогично. Кроме того, настроим этим компьютерам шлюз по умолчанию — 10.10.10.254:



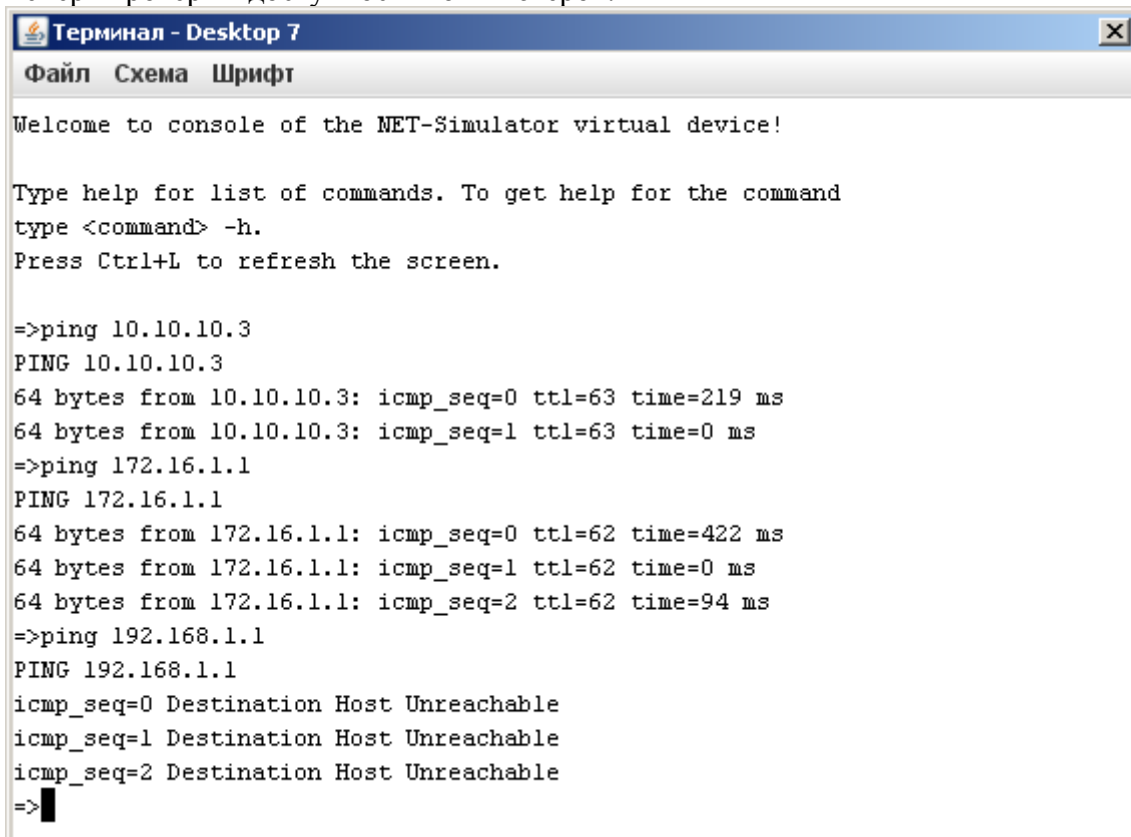
```
Терминал - Desktop 7
Файл Схема Шрифт

Welcome to console of the NET-Simulator virtual device!

Type help for list of commands. To get help for the command
type <command> -h.
Press Ctrl+L to refresh the screen.

=>ifconfig eth0 10.10.10.1 -netmask 255.255.255.0 -up
=>route -add 0.0.0.0 -netmask 0.0.0.0 -gw 10.10.10.254
=>route
IP routing table
Destination      Gateway          Netmask          Flags Metric Iface
10.10.10.0       *                255.255.255.0   U      1      eth0
0.0.0.0          10.10.10.254    0.0.0.0          UG     0      eth0
=>
```

Теперь проверим доступность компьютеров:



```
Терминал - Desktop 7
Файл Схема Шрифт

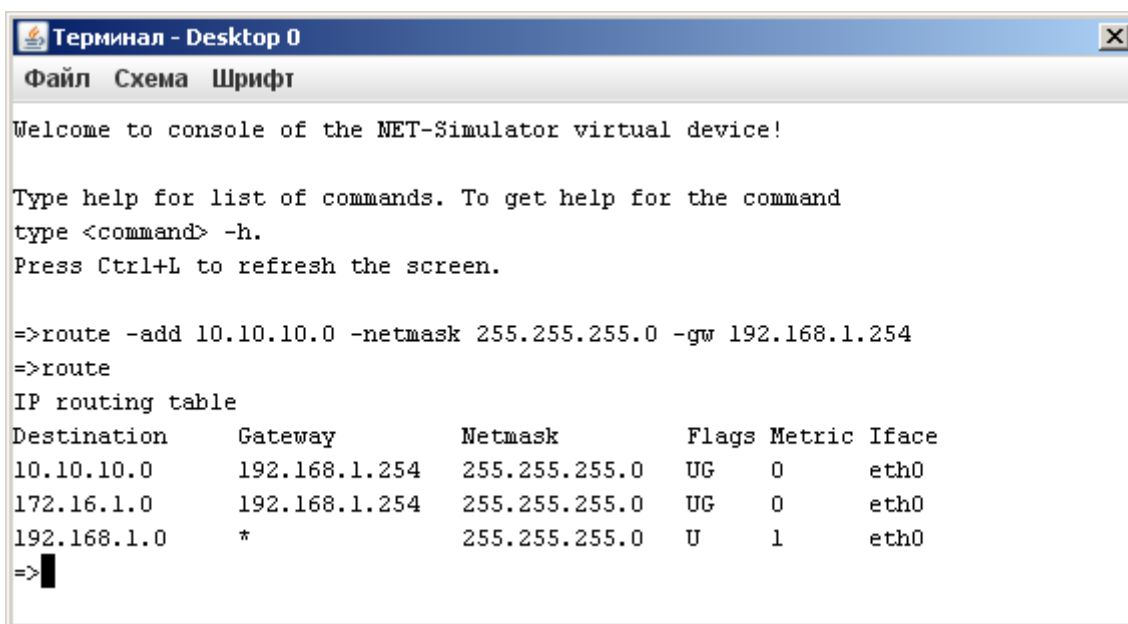
Welcome to console of the NET-Simulator virtual device!

Type help for list of commands. To get help for the command
type <command> -h.
Press Ctrl+L to refresh the screen.

=>ping 10.10.10.3
PING 10.10.10.3
64 bytes from 10.10.10.3: icmp_seq=0 ttl=63 time=219 ms
64 bytes from 10.10.10.3: icmp_seq=1 ttl=63 time=0 ms
=>ping 172.16.1.1
PING 172.16.1.1
64 bytes from 172.16.1.1: icmp_seq=0 ttl=62 time=422 ms
64 bytes from 172.16.1.1: icmp_seq=1 ttl=62 time=0 ms
64 bytes from 172.16.1.1: icmp_seq=2 ttl=62 time=94 ms
=>ping 192.168.1.1
PING 192.168.1.1
icmp_seq=0 Destination Host Unreachable
icmp_seq=1 Destination Host Unreachable
icmp_seq=2 Destination Host Unreachable
=>
```

Мы видим, что компьютеры сети 172.16.1.0 доступны, а компьютеры сети 192.168.1.0 — нет. Это опять таки произошло потому, что в сети 192.168.1.0 компьютеры не знают, куда отправлять пакеты, чтобы они попали в сеть 10.10.10.0 — у них нет такого маршрута и нет

шлюза по умолчанию. Зададим маршруты компьютерам Desktop 0 Desktop 1 и Desktop 2. Мы приводим пример для Desktop 0, для остальных двух задайте маршруты самостоятельно:



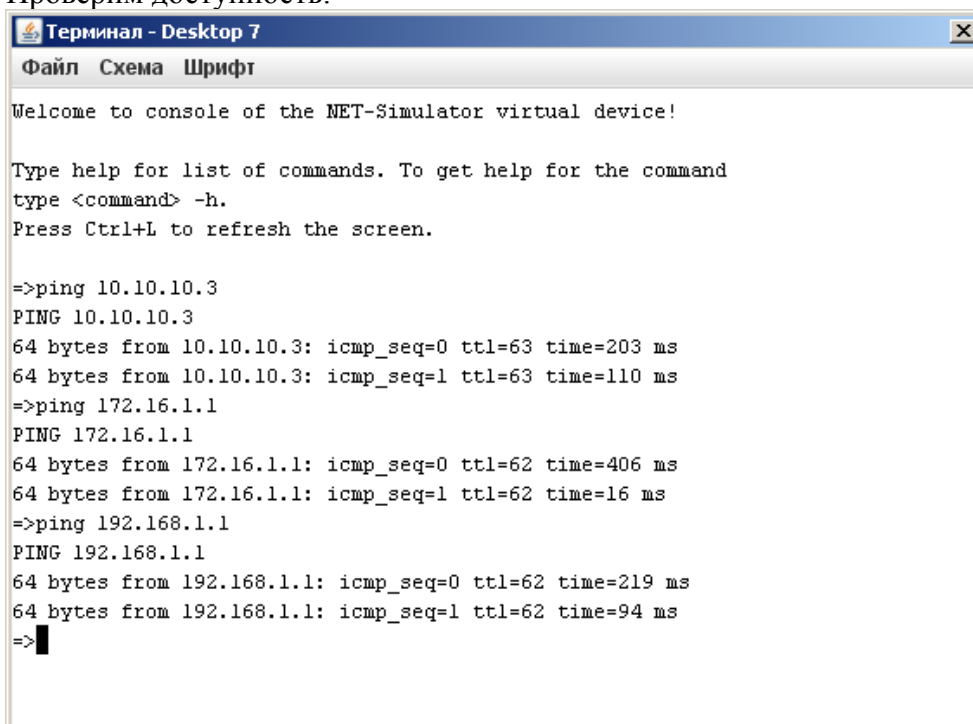
```
Терминал - Desktop 0
Файл  Схема  Шрифт

Welcome to console of the NET-Simulator virtual device!

Type help for list of commands. To get help for the command
type <command> -h.
Press Ctrl+L to refresh the screen.

=>route -add 10.10.10.0 -netmask 255.255.255.0 -gw 192.168.1.254
=>route
IP routing table
Destination      Gateway          Netmask          Flags Metric Iface
10.10.10.0       192.168.1.254   255.255.255.0   UG    0    eth0
172.16.1.0       192.168.1.254   255.255.255.0   UG    0    eth0
192.168.1.0      *                255.255.255.0   U    1    eth0
=>
```

Проверим доступность:



```
Терминал - Desktop 7
Файл  Схема  Шрифт

Welcome to console of the NET-Simulator virtual device!

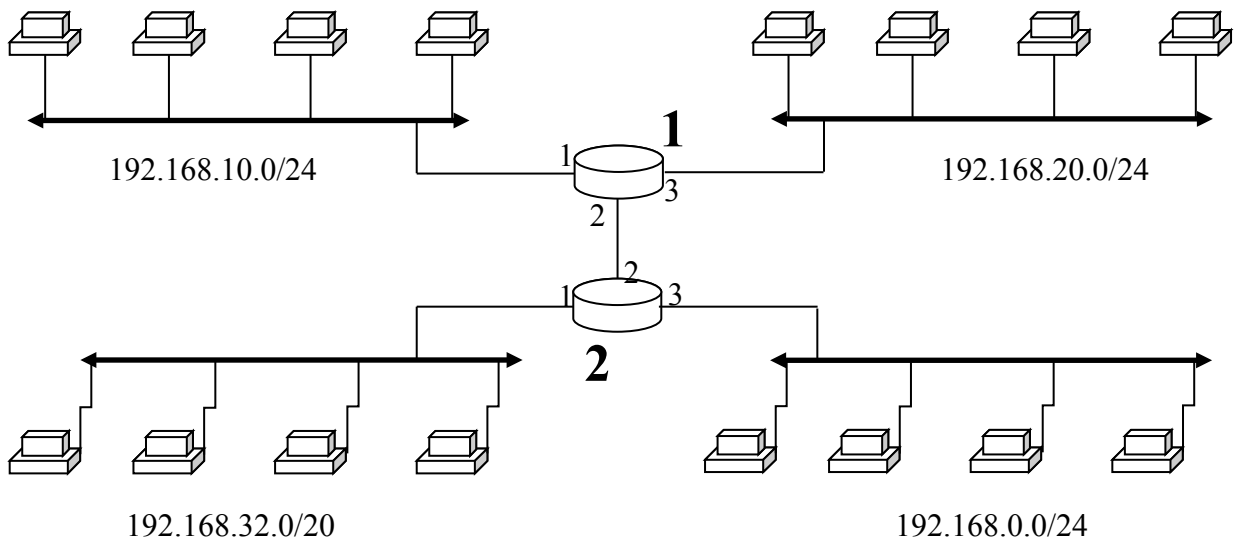
Type help for list of commands. To get help for the command
type <command> -h.
Press Ctrl+L to refresh the screen.

=>ping 10.10.10.3
PING 10.10.10.3
64 bytes from 10.10.10.3: icmp_seq=0 ttl=63 time=203 ms
64 bytes from 10.10.10.3: icmp_seq=1 ttl=63 time=110 ms
=>ping 172.16.1.1
PING 172.16.1.1
64 bytes from 172.16.1.1: icmp_seq=0 ttl=62 time=406 ms
64 bytes from 172.16.1.1: icmp_seq=1 ttl=62 time=16 ms
=>ping 192.168.1.1
PING 192.168.1.1
64 bytes from 192.168.1.1: icmp_seq=0 ttl=62 time=219 ms
64 bytes from 192.168.1.1: icmp_seq=1 ttl=62 time=94 ms
=>
```

Таким образом, становится ясным, что наиболее правильным построением маршрутизации является построение с использованием «шлюза по умолчанию», поскольку в этом случае при добавлении новых подсетей нам не нужно ничего настраивать на компьютерах, а достаточно один раз указать маршрут на маршрутизаторе, который является шлюзом для подсетей.

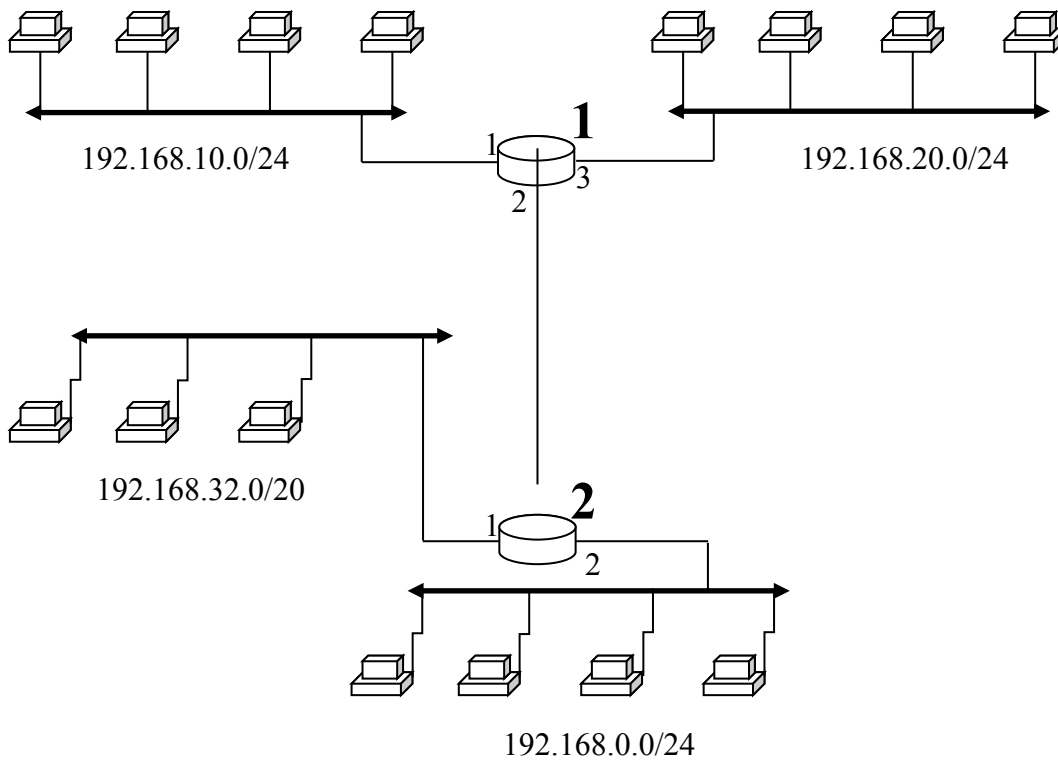
Задание
Построить часть своей модели ЛВС

Вариант 1

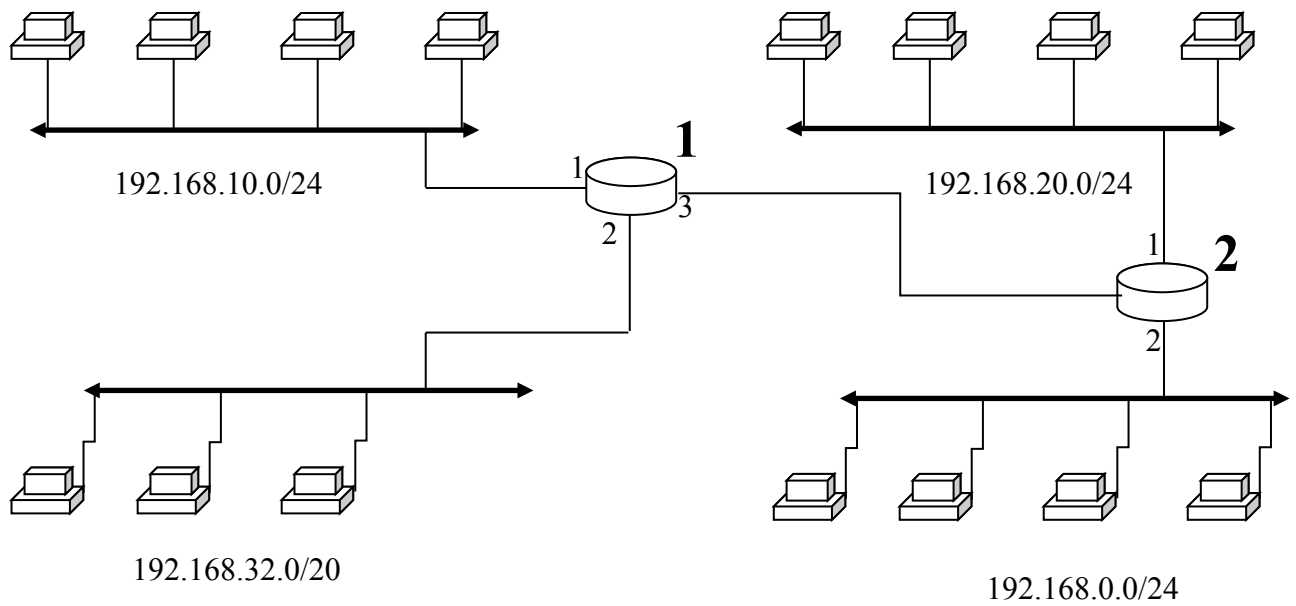


Вариант 2

Вар

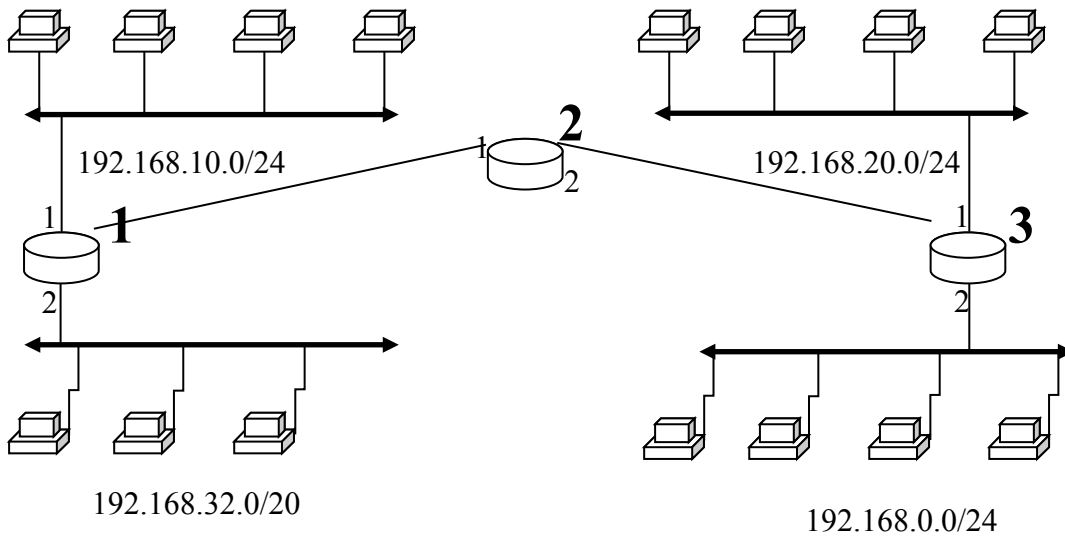


Вариант 3

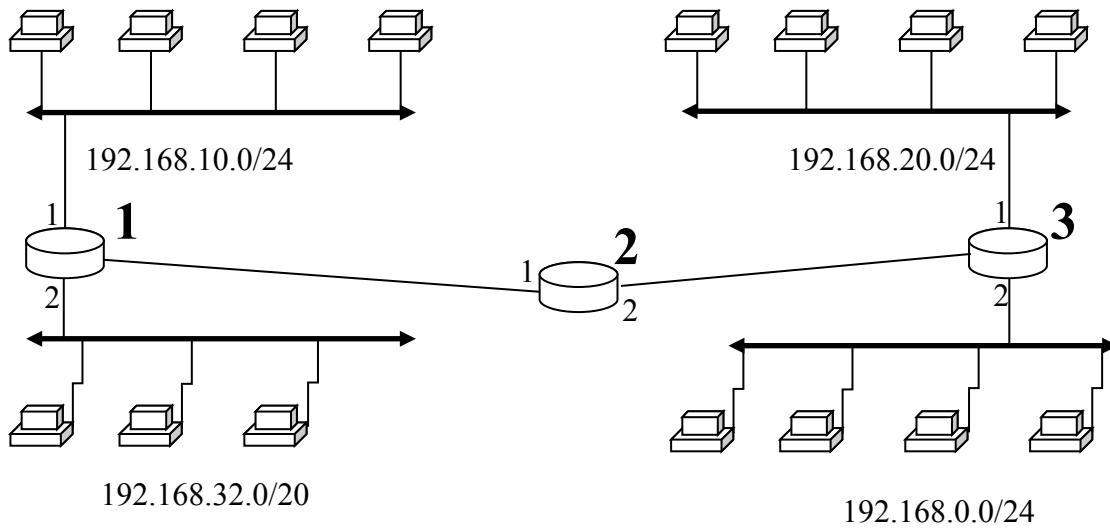


Вариант 4

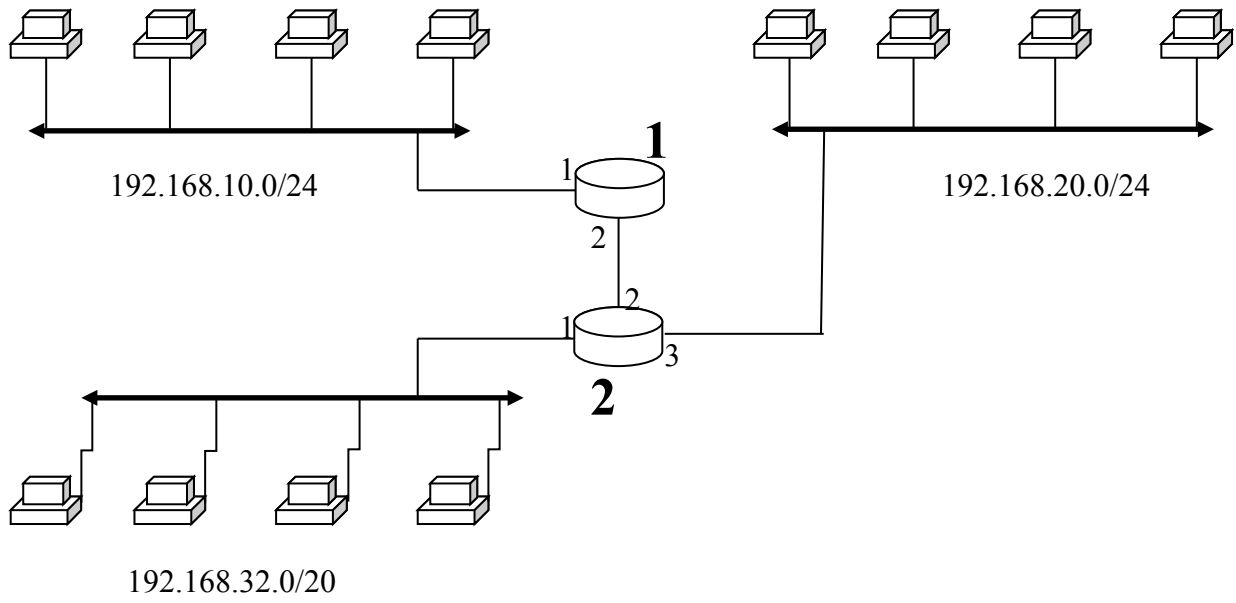
Вариант



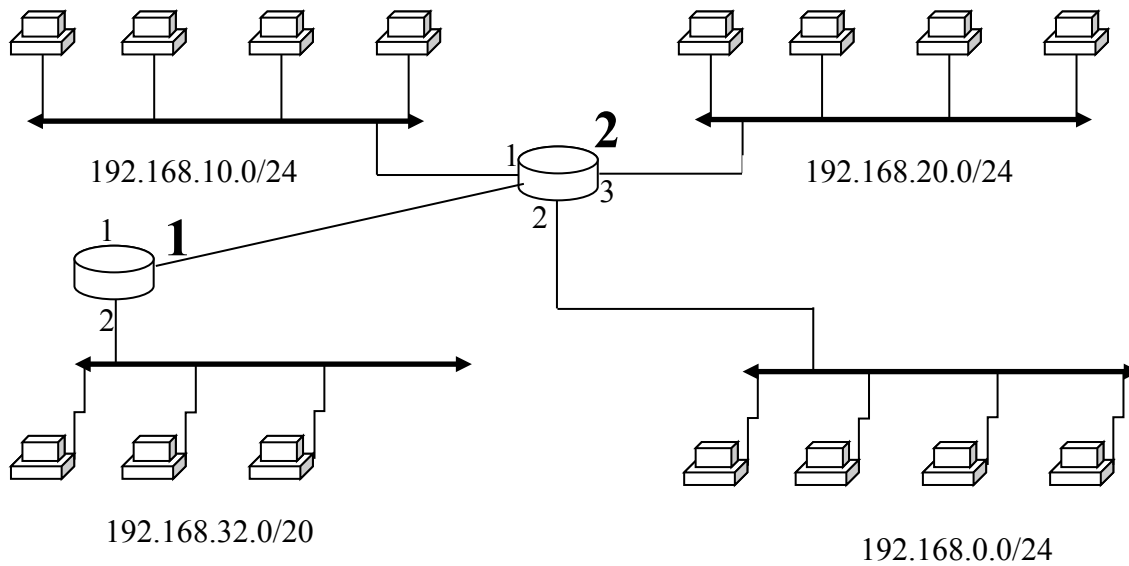
ВТ 5



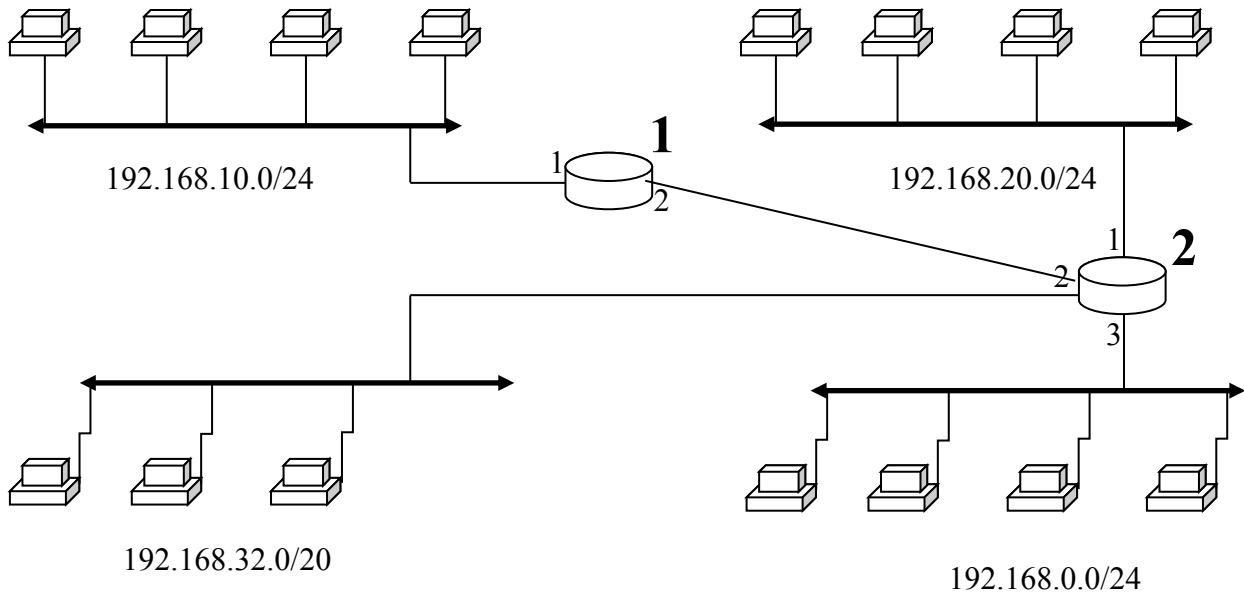
Вариант 6



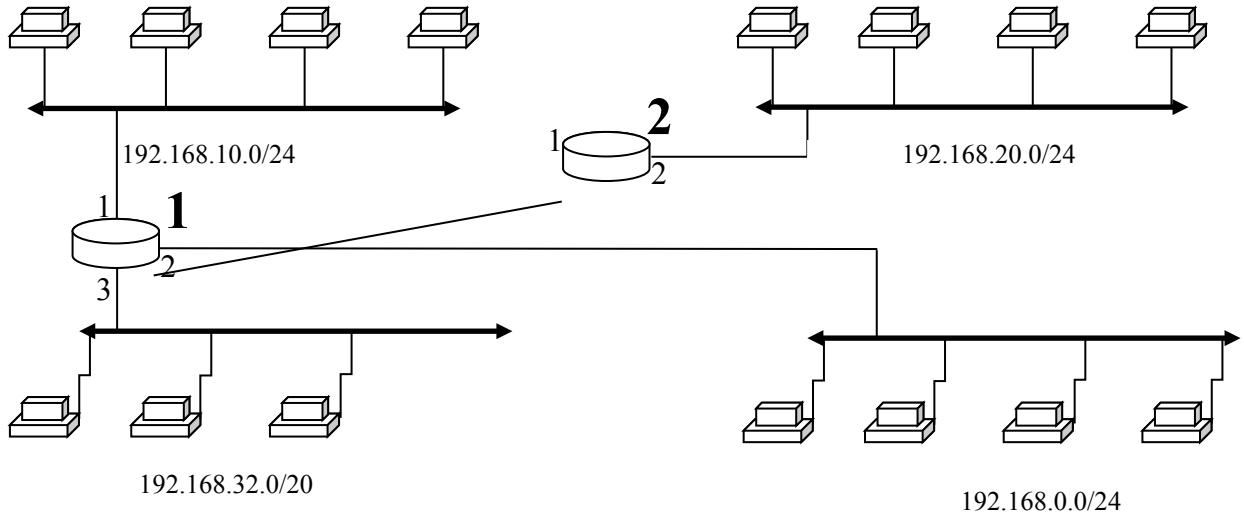
Вариант 7



Вариант 8

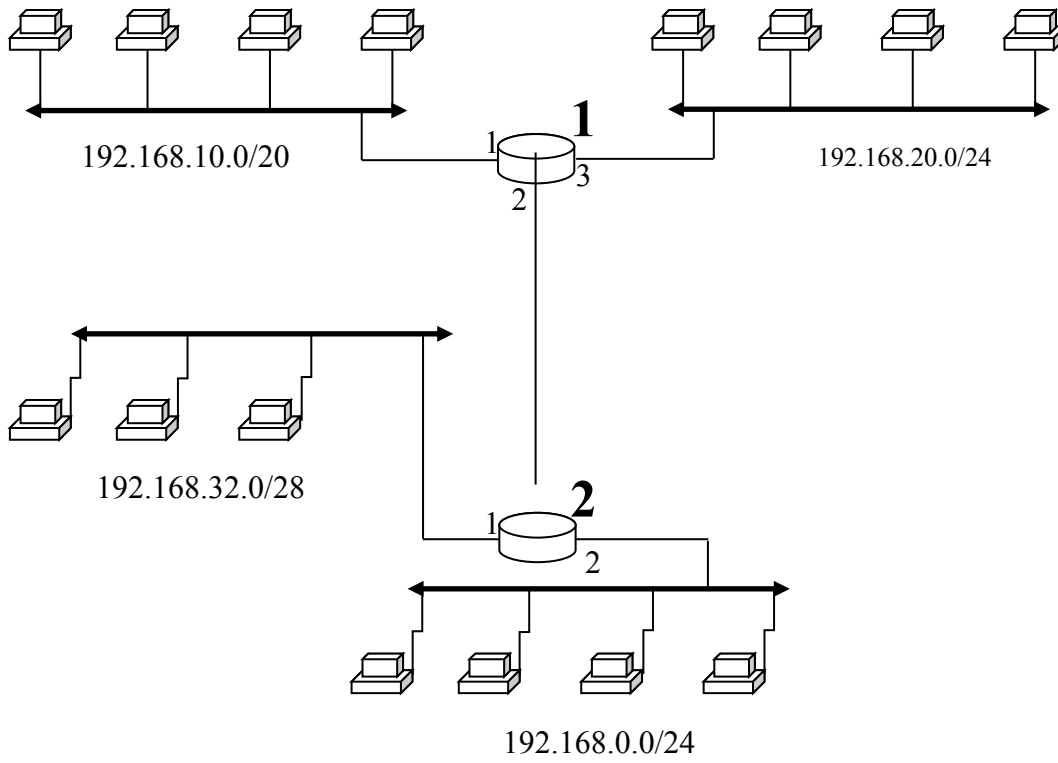


Вариант 9

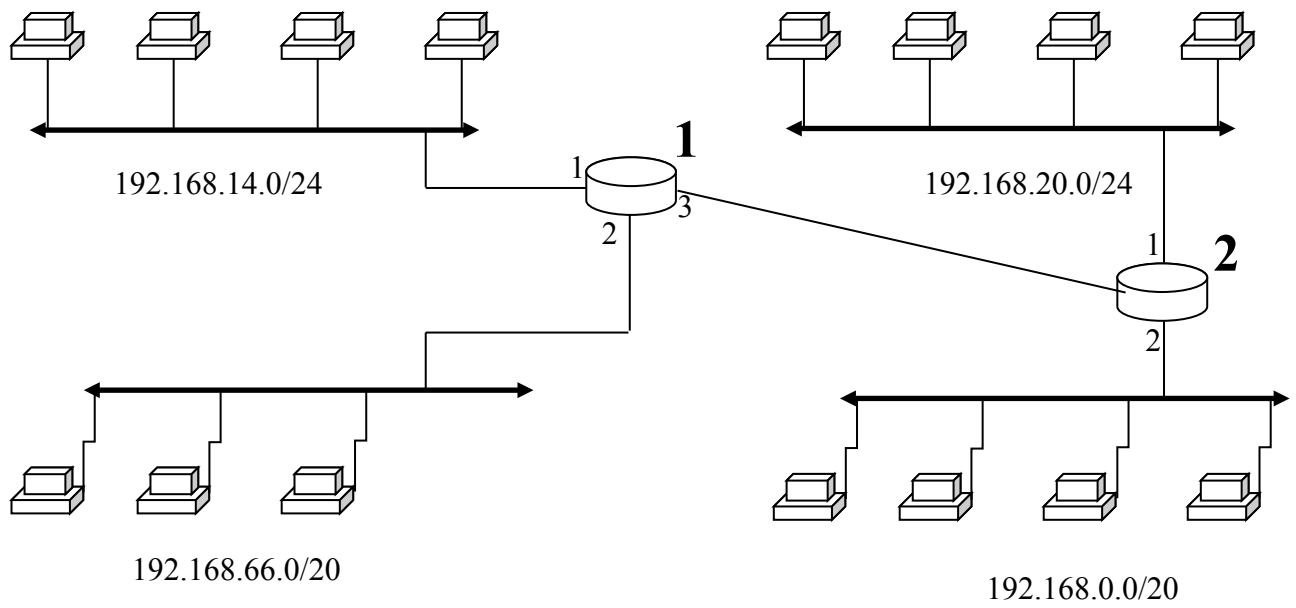


Вариант 10

Вар

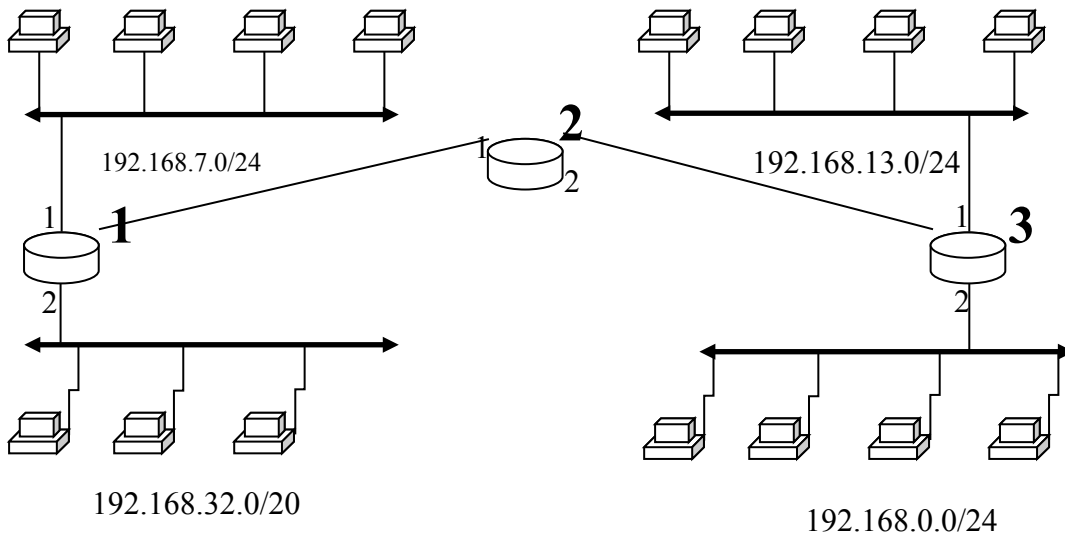


Вариант 11

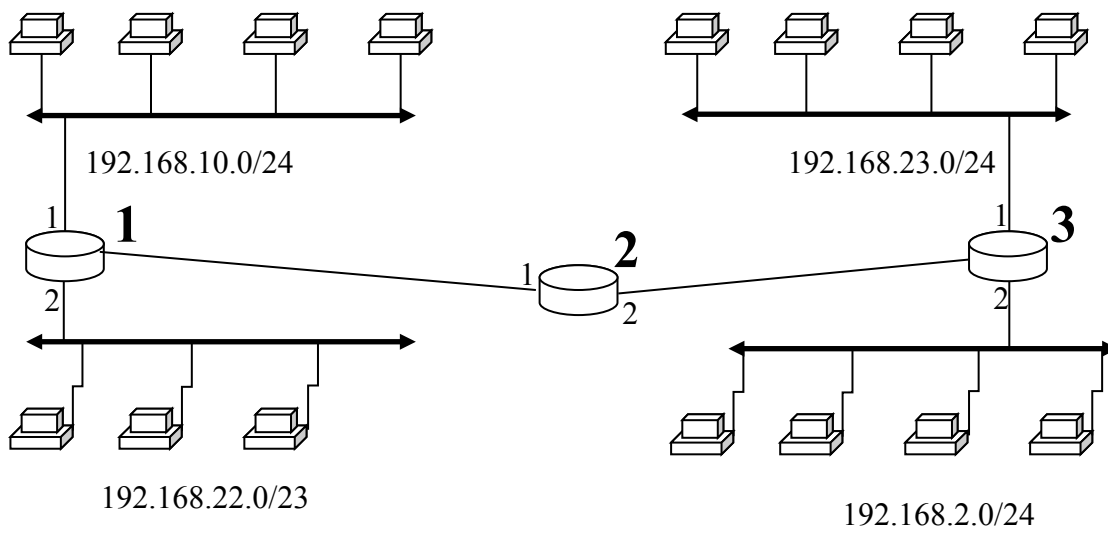


Вариант 12

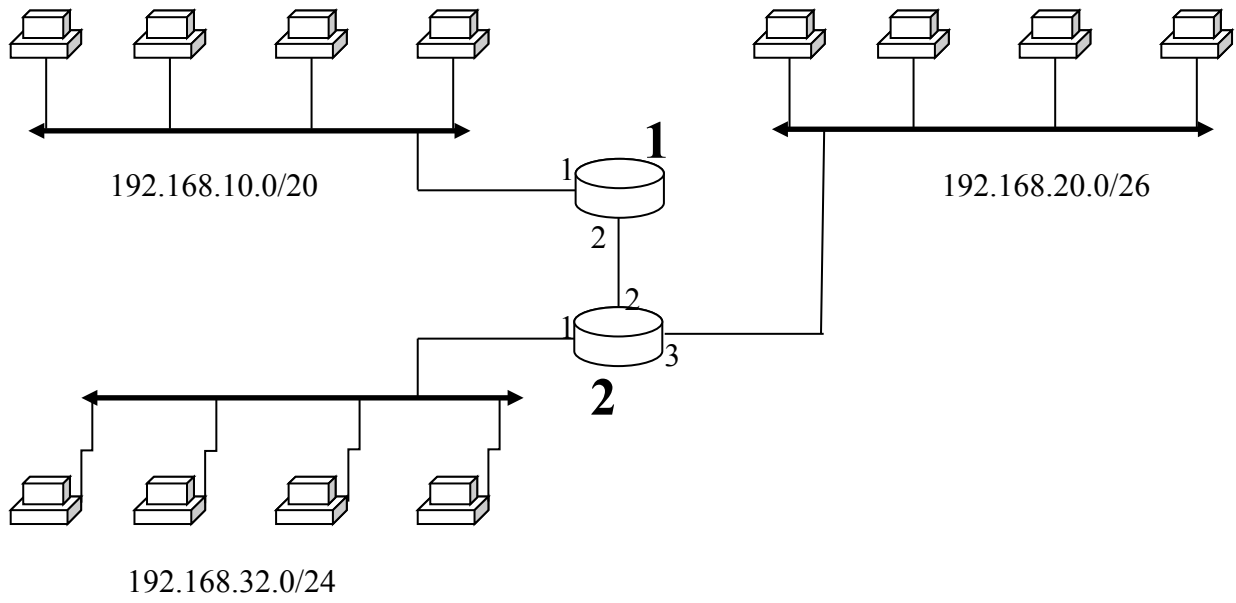
Вариант



Вариант 13



Вариант 14



Вариант 15

