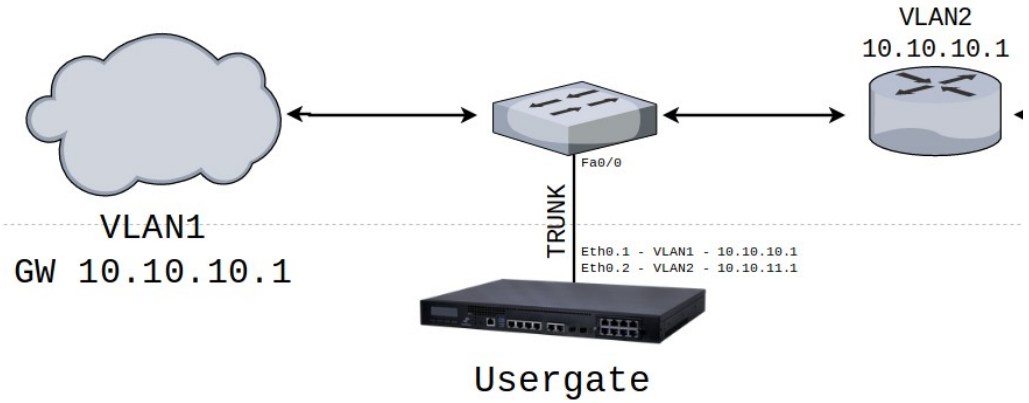
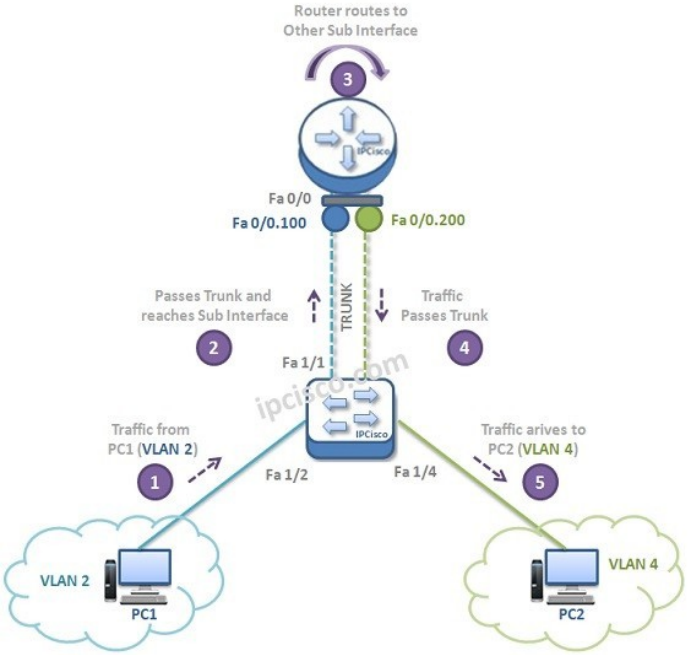
Router-on-stick
and
ACL





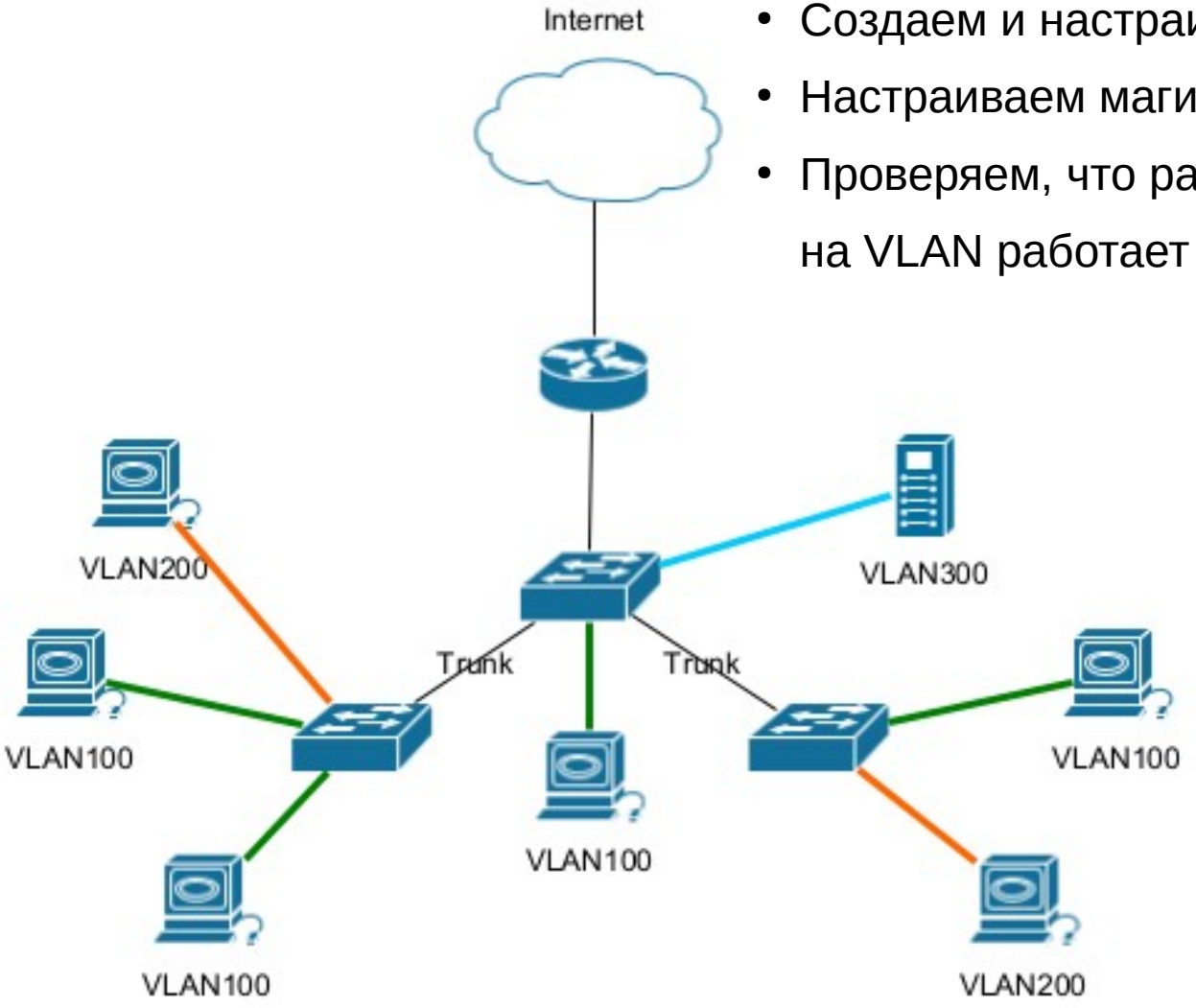
Задача:
Реализовать управление доступом между группами узлов сети.
Разделить широковещательный домен

Например:
ИТ специалистам — доступ в Интернет и ко всем узлам
«Гостям» - доступ только в Интернет

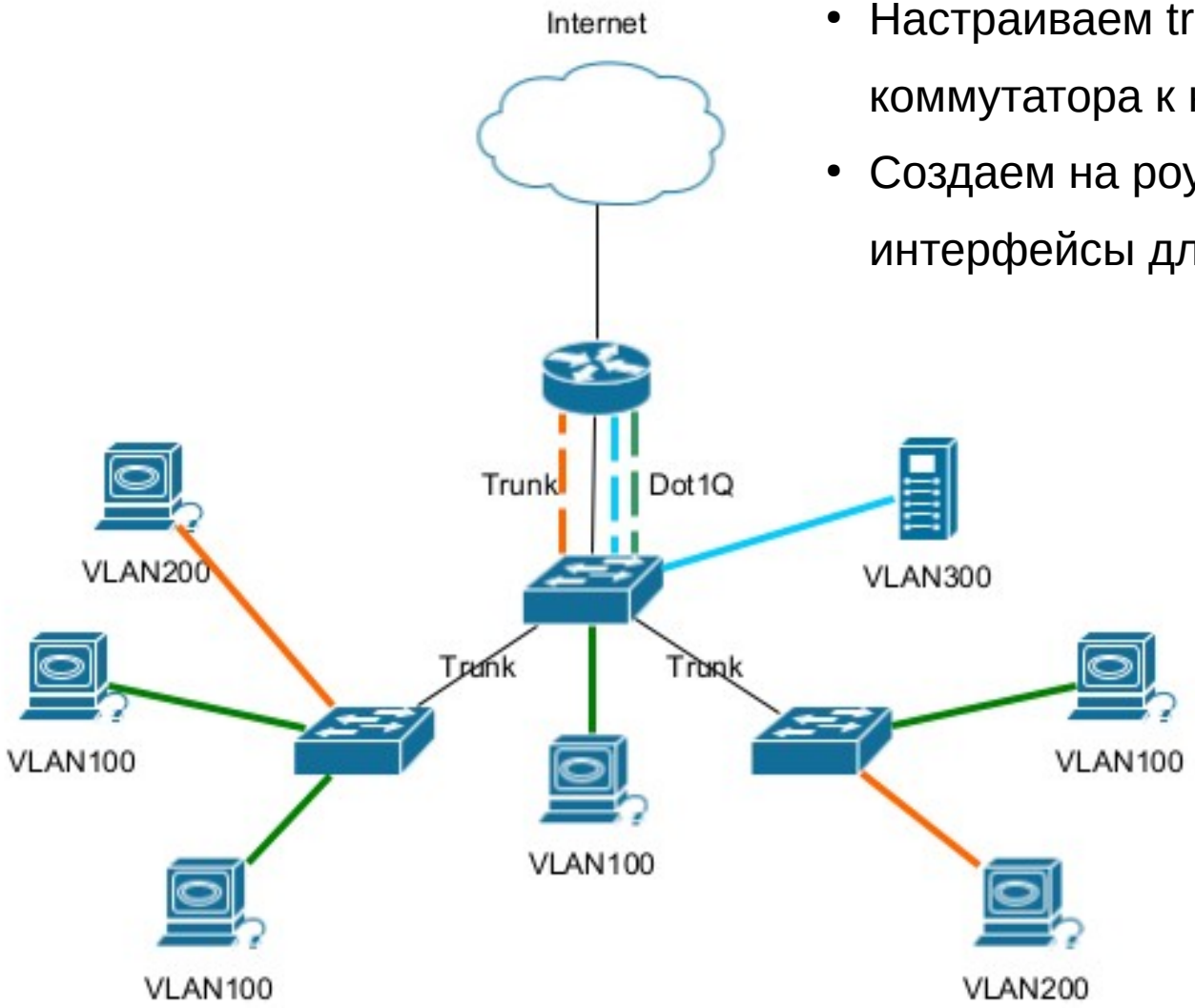
Решения:
1. Использовать маршрутизатор или файрволл
с соответствующим количеством маршрутизируемых портов.

2. Настроить правила брандмауэра на узлах

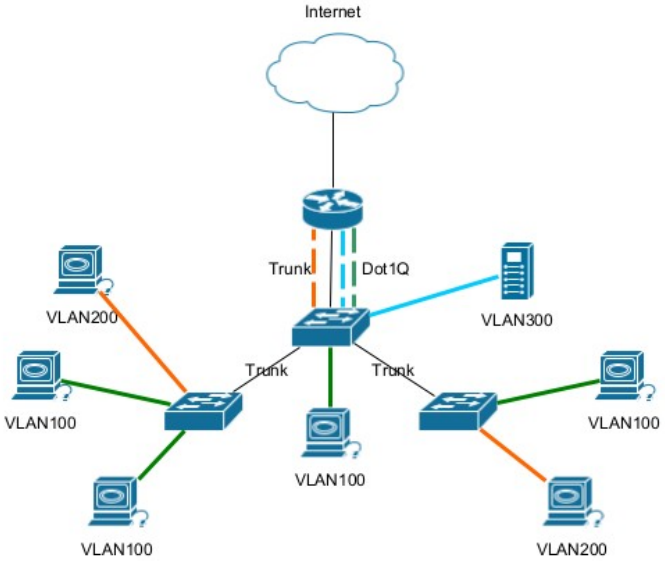
3. Разделить сеть на VLAN и использовать Router-on-stick

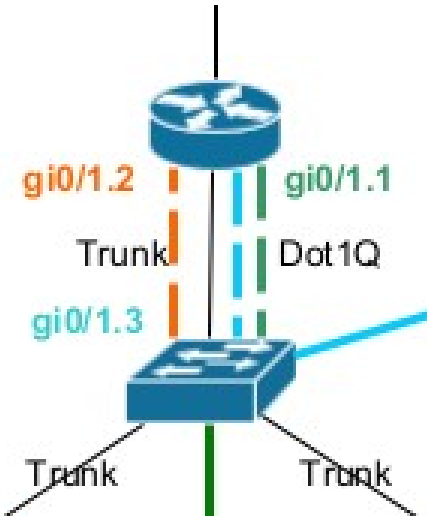


- Создаем и настраиваем VLAN
- Настраиваем магистрали (trunk)
- Проверяем, что разбиение на VLAN работает



- Настраиваем trunk на порту коммутатора к которому подключен роутер
- Создаем на роутере виртуальные интерфейсы для наших VLAN






```
Router(config)# interface gigabitethernet0/1
Router(config-if)# no ip address
```

Для VLAN 100

```
Router(config-if)# interface gigabitethernet0/1.1
Router(config-subif)# encapsulation dot1q 100
Router(config-if)# no shutdown
```

Для VLAN 200

```
Router(config-subif)# interface gigabitethernet0/1.2
Router(config-subif)# encapsulation dot1q 200
Router(config-if)# no shutdown
```

Для VLAN 300

```
Router(config-subif)# interface gigabitethernet0/1.3
Router(config-subif)# encapsulation dot1q 300
Router(config-if)# no shutdown
```

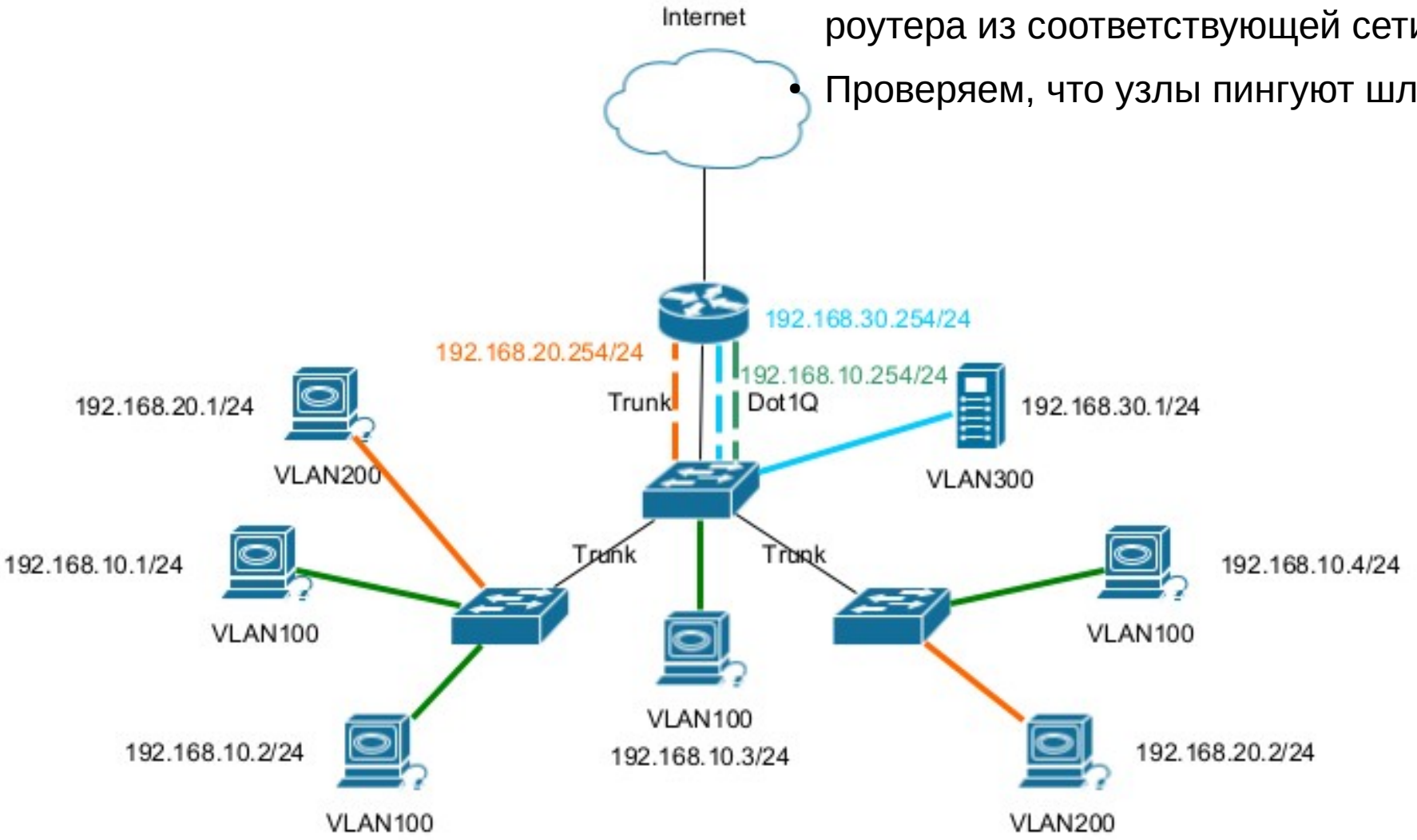
Для VLAN 1 - native

```
Router(config-subif)# interface gigabitethernet0/1.4
Router(config-subif)# encapsulation dot1q 1 native
Router(config-if)# no shutdown
```

Поднимаем интерфейс

```
Router(config)# interface gigabitethernet0/1
Router(config-if)# no shutdown
```

- Настраиваем IP адреса и шлюзы.
  Шлюз — адрес виртуального интерфейса роутера из соответствующей сети
- Проверяем, что узлы пингуют шлюзы



ACL

- Настраиваем ACL.

ip access-list <extended|standard> name
<permit|deny> <tcp|udp> source source-wildcard [operator [port]] destination destination-wildcard [operator [port]] [log]

Стандартные ACL

router(conf)# access-list <1-99> <permit | deny | remark> source [source-wildcard]

Применение ACL на интерфейсе:
router(conf-if)# ip access-group <1-99> <in | out>

Пример

R1(conf)# access-list 1 deny 10.0.3.2
R1(conf)# access-list 1 permit any

Применение ACL на интерфейсе
R1(conf)# interface fa0/1
R1(conf-if)# ip access-group 1 out

deny ip any any
в конце каждого ACL!

in — правило для входящих пакетов
out — правило для исходящих

ACL

Расширенные ACL

Расширенный ACL, при указании протоколов IP, ICMP и др., позволяет указывать IP-адреса отправителя и получателя:

router(conf)# access-list acl-number <permit|deny> <ip|icmp|ospf|eigrp...> source source-wildcard destination destination-wildcard

Расширенный ACL, при указании протоколов TCP или UDP, позволяет указывать и порты отправителя и/или получателя:

router(conf)# access-list acl-number <deny|permit> <tcp|udp> source source-wildcard [operator [port]] destination destination-wildcard [operator [port]] [log]

Пример

vlan = 10.0.1.0/24 и vlan2 = 10.0.2.0/24, Возникла необходимость запретить трафику ходить из vlan2 в vlan, но из vlan во vlan 2 доступ должен остаться прежним

```
ip access-list extended deny_vlan2_to_vlanX
 deny ip 10.0.2.0 0.0.0.255 10.0.1.0 0.0.0.255
 permit ip any any
```

назначить ACL:

```
interface vlanX
ip access-group deny_vlan2_to_vlanX out
```

ACL

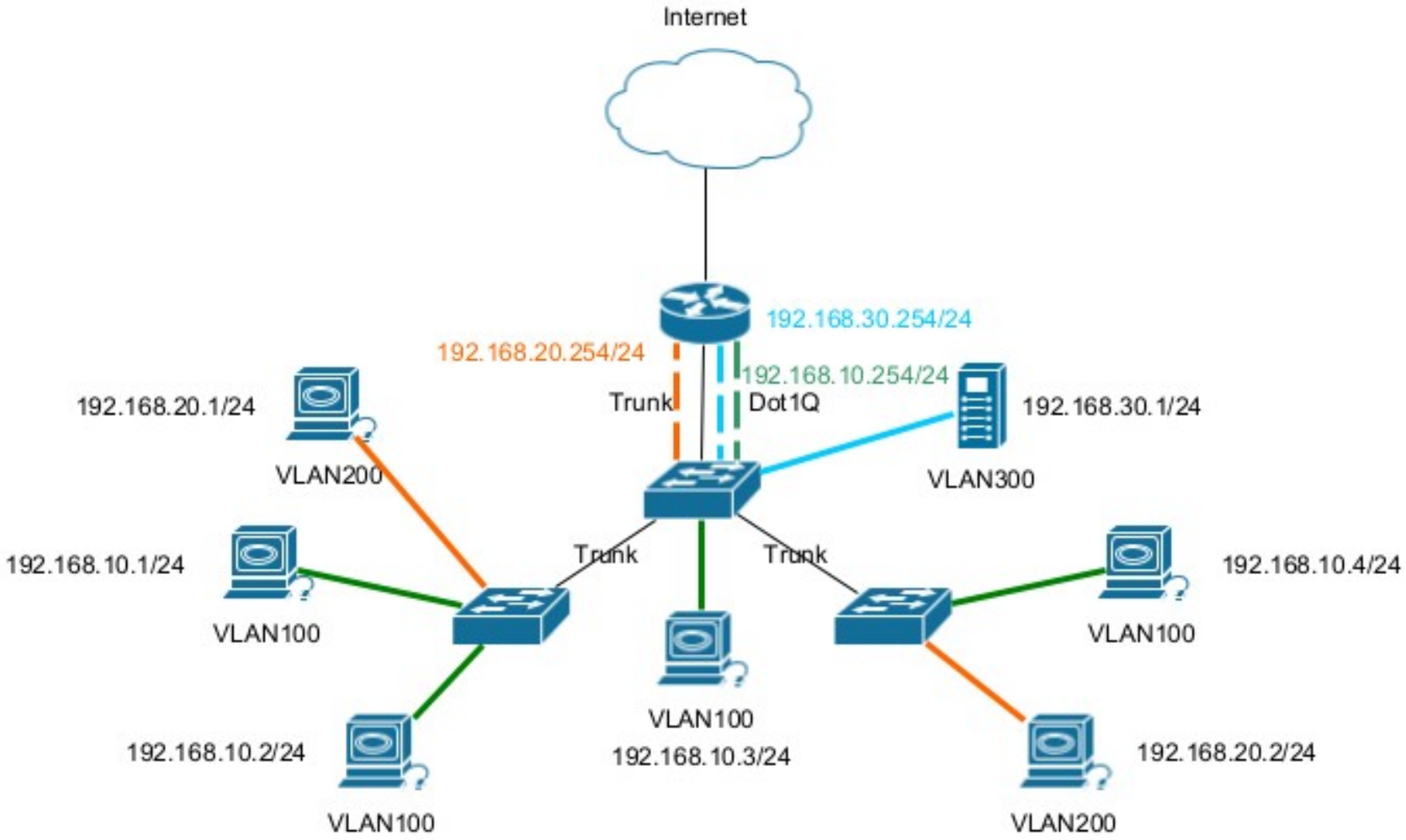
Wildcard — обратная маска

В обратной маске
там где находится адрес хоста идут 1
там где находится адрес сети идут 0
0 и 1 могут быть перемешаны!

Например
Для сети
172.16.2.4/30

Маска 11111111. 11111111. 11111111. 11111**00** → 255.255.255.252
Обратная маска/Wildcard 00000000.00000000.00000000.00000**11** → 0.0.0.3

Подробнее о том, зачем это придумано можно прочитать тут:
https://habr.com/ru/articles/131712/


Internet
192.168.30.254/24
192.168.20.254/24
192.168.10.254/24
Trunk
Dot1Q
192.168.20.1/24
VLAN200
192.168.30.1/24
VLAN300
192.168.10.1/24
VLAN100
Trunk
Trunk
192.168.10.4/24
VLAN100
VLAN100
192.168.10.3/24
192.168.10.2/24
VLAN100
192.168.20.2/24
VLAN200