Министерство науки и высшего образования Российской Федерации

Томский государственный университет
систем управления и радиоэлектроники

Е.Ю. Агеев

А.В. Бусыгина

ТЕОРИЯ ПОСТРОЕНИЯ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И СЕТЕЙ

Учебно-методическое пособие
по практическим занятиям и самостоятельной работе

Томск 2024

УДК 004.7
БКК 32.971.35
А23

Рецензент:
Рогожников Е.В., заведующий кафедрой ТОР ТУСУР, к.т.н., доцент

А23 Теория построения инфокоммуникационных систем и сетей: учеб.-метод. пособие по практическим занятиям и самостоятельной работе / Е.Ю. Агеев, А.В. Бусыгина – Томск : Томск. гос. ун-т систем упр. и радиоэлектроники, 2024. – 84 с.

Учебно-методическое пособие содержит указания для выполнения практических и самостоятельных работ. Цель данного практикума заключается в приобретении студентами навыков работы с сетевыми технологиями, построения компьютерных сетей различного уровня: локальных и глобальных компьютерных сетей. Практикум направлен на развитие практических навыков студентов в применении полученных теоретических знаний для решения задач, связанных с исследованиями современных инфокоммуникационных систем и сетей, а также расчета и проектирования компьютерных сетей. Предназначено для студентов технических специальностей, также пособие может быть полезно студентам смежных специальностей и ИТ.

Одобрено на заседании ПИШ, протокол № 2 от 21.10.2023

УДК 004.7
БКК 32.971.35

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Цель данного практикума заключается в приобретении студентами навыков работы с сетевыми технологиями, построения компьютерных сетей различного уровня: локальных и глобальных компьютерных сетей. Практикум направлен на развитие практических навыков студентов в применении полученных теоретических знаний для решения задач, связанных с исследованиями современных инфокоммуникационных систем и сетей, а также расчета и проектирования компьютерных сетей.

Практикум посвящен изучению следующих тем:

- Анализ широковещательного и группового трафика в Wireshark.
- Настройка скорости, дуплекса и других параметров сетевых интерфейсов Ethernet.
- Настройка VLAN, VTP и маршрутизации между VLAN.
- Сегментация сети с помощью подсетей, настройка маршрутизации.
- Анализ механизмов управления потоком и перегрузкой в протоколе TCP.
- Настройка аутентификации в беспроводной сети под управлением контроллера.
- Протокол расширенной аутентификации, стандарт IEEE 802.1X.

ПРАКТИКА №1
АНАЛИЗ ШИРОКОВЕЩАТЕЛЬНОГО И ГРУППОВОГО ТРАФИКА В WIRESHARK

Цель работы: научиться использовать сетевой анализатор Wireshark для анализа широковещательного и группового трафика.

ОБЩАЯ ИНФОРМАЦИЯ

Анализатор сетевого трафика Wireshark

Программа Wireshark популярный и распространенный анализатор сетевого трафика. Ее использование сетевыми инженерами стало де-факто стандартом. Это бесплатный и кроссплатформенный программный продукт. На рис. 1 изображена схематично структура сетевой подсистемы ОС. Вся базовая инфраструктура реализована в виде драйверов и работает в режиме ядра. Пользовательские процессы и реализации прикладных протоколов, в частности интерфейс Wireshark работают в пользовательском режиме.

На рисунке отображены два пользовательских процесса («сетевой процесс 1» и «сетевой процесс 2»).

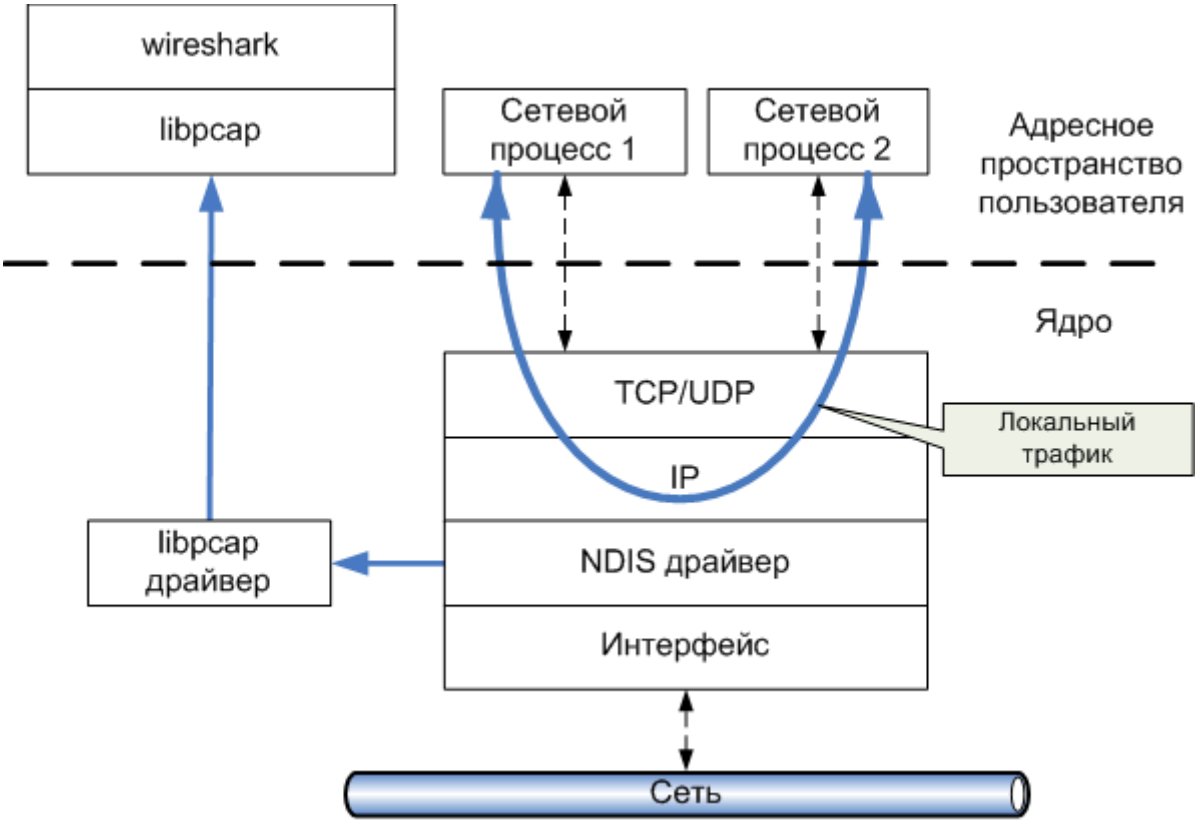


Рис. 1 Иллюстрация принципа захвата сетевого трафика Wireshark

Wireshark не позволяет производить анализ локального трафика, т.к. этот трафик не проходит через драйвер сетевого устройства. Т.е., если вы захотите проанализировать трафик между 2-мя сетевыми процессами на локальной машине (например, ftp-сервер и ftp-клиент), то это сделать не получится. Но при использовании виртуальных машин, трафик на виртуальных интерфейсах можно захватить, т.к. виртуальные машины эмулируют реальную среду и сетевые адаптеры.

На рис. 2. изображено основное окно программы Wireshark. Оно делится на 3 части (панели): список захваченных пакетов, детализация структуры протоколов в выделенном пакете и двоичные данные для выделенного пакета.

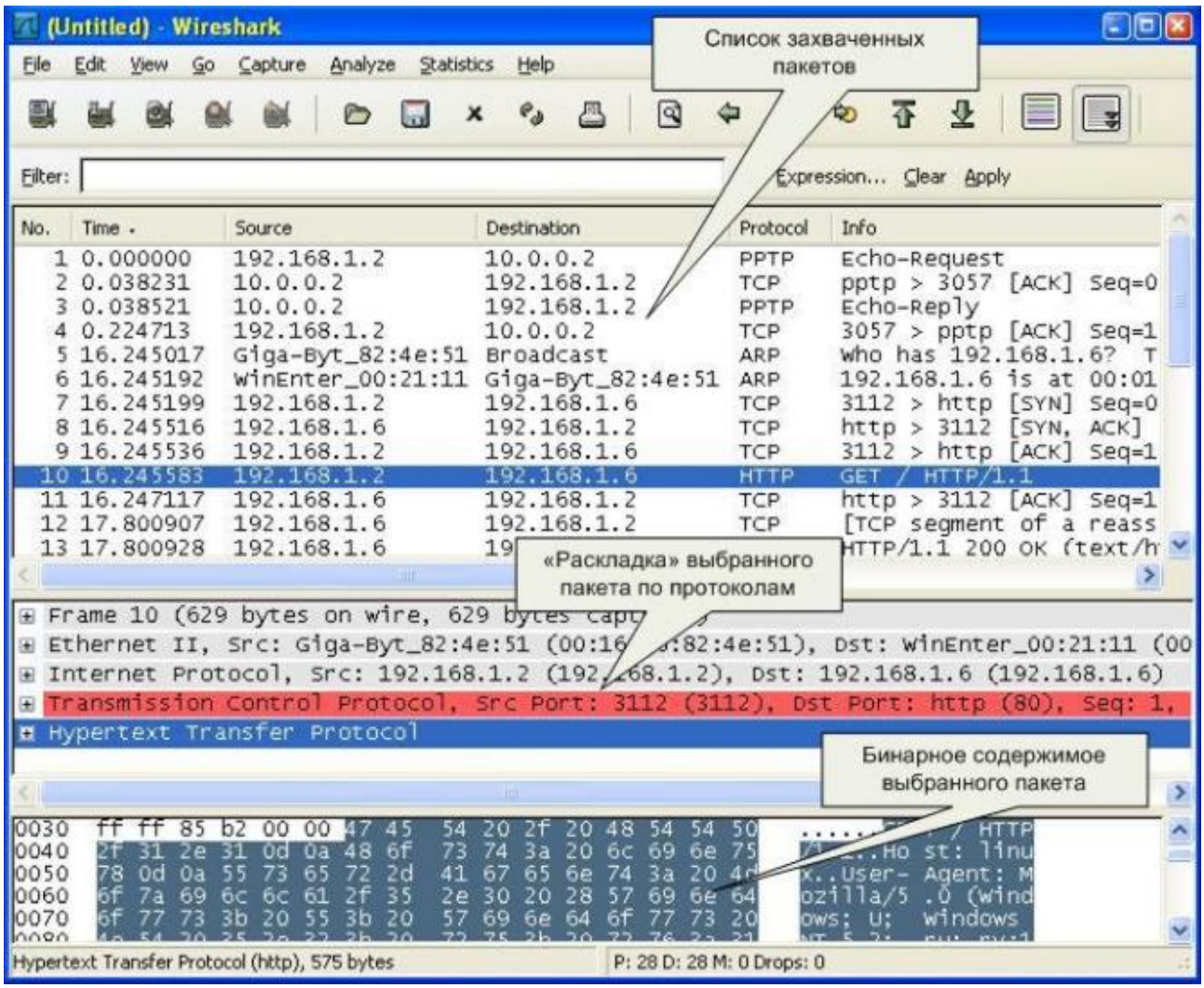


Рис. 2 Графический интерфейс Wireshark

Захват трафика (live capture)

Перед захватом трафика необходимо выбрать сетевые интерфейсы, пакеты с которых будут захвачены. Для этого можно нажать пиктограмму плавника акулы и выбрать из выпадающего списка требуемый интерфейс, рис. 3:

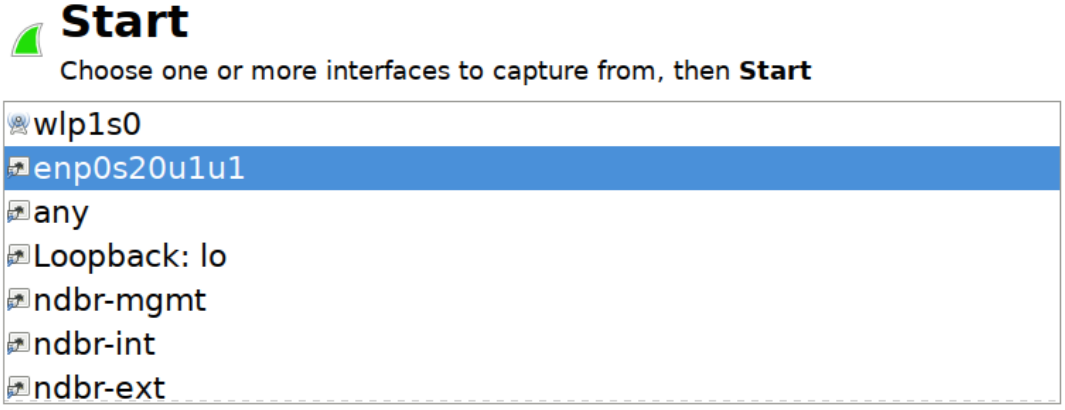


Рис. 3 Выбор сетевого интерфейса для захвата трафика

ПОДГОТОВЛЕННАЯ ВИРТУАЛЬНАЯ МАШИНА MININET

Работа выполняется в программе Mininet. Mininet — это эмулятор сети, который создает сеть из виртуальных компьютеров, коммутаторов, управляющих сетью контроллеров, при этом все они имеют реальные сетевые интерфейсы на основе стандартного сетевого программного обеспечения Linux.

Эмулятор Mininet широко используется в научных исследованиях, разработке, а также при обучении сетевым технологиям, создании прототипов, тестировании, отладке и других задачах, при решении которых необходимо наличие полной экспериментальной сети на ноутбуке или другом ПК. Мининет имеет интерфейс командной строки с поддержкой команд создания заданных топологий и протокола OpenFlow для отладки или запуска общесетевых тестов. Поддерживает произвольные пользовательские топологии и включает в себя базовый набор параметризованных топологий. Программа Mininet работает на платформе Linux. Для запуска ее на Windows мы будем использовать подготовленные образы виртуальной машины в формате OVF (Open Virtualization Format) — открытый стандарт для хранения и распространения виртуальных машин. Стандарт описывает открытый, переносимый, расширяемый формат для распространения образов виртуальных машин. Стандарт OVF не привязан к какой-либо реализации гипервизора или аппаратной архитектуре. Все доступные релизы доступны в репозитории https://github.com/mininet/mininet/releases/.

Скачайте последний актуальный релиз на основе Ubuntu 20.04 https://github.com/mininet/mininet/releases/download/2.3.0/mininet-2.3.0-210211-ubuntu-20.04.1-legacy-server-amd64-ovf.zip [1].

Логин и пароль для входа **mininet/mininet**

Программа Wireshark уже установлена в виртуальной машине, но для уменьшения размера образа не установлено графическое окружение рабочего стола, которое в Linux работает как отдельный сервис.

Для запуска Wireshark мы будем использовать внешний сервер графического режима. Такой сервер есть в составе популярного SSH-клиента **MobaXterm**

Загрузите его по ссылке:
https://download.mobatek.net/2212022060563542/MobaXterm_Portable_v22.1.zip [2].

Если мы хотим запускать программы на внешнем сервере графического окружения, то мы должны подключиться по протоколу SSH к виртуальной машине Mininet с указанием ключа перенаправления запросов выполнения графических программ на внешний сервер графического режима: -X или -Y, таким образом:

ssh -Y mininet@IPaddress_mininetVM

Здесь IP-адрес виртуальной машины это адрес сетевого адаптера, включенного в режиме «Сетевой мост».
(Если вы работаете с машиной в режиме NAT, по умолчанию, тогда можно настроить проброс портов в расширенных настройках сети и указывать адрес интерфейса сетевого адаптера VirtualBox HostOnly, обычно это 192.168.56.1).

Эмулятор запускается командой **mn** от имени администратора:

sudo mn

При подключении с пробросом X Window сессии через SSH соединения по сети на машине Mininet открывается сокет на порту 6000. Чтобы защитить это соединение в профиле

пользователя, который подключился по SSH (в нашем случае это пользователь mininet) в специальном файле сохраняются MAGIC-COOKIE — уникальный цифровой отпечаток, разрешающий выполнение графических программ этому пользователю. Другим пользователям, у которых не будет этого цифрового кода, не будет и разрешения.

Просмотреть сохраненный код можно командой:

xauth list $DISPLAY

или просто **xauth list** так как других переменных не будет.

Так как подключение по SSH выполняется пользователем mininet, а запуск программы- эмулятора от имени администратора, то возникнет проблема с запуском графических программ, таких как Wireshark в окне Mininet. Чтобы администратор мог запустить графические программы нужно скопировать в его профиль MAGIC-COOKIE из профиля пользователя mininet. Сделать это можно командой:

xauth add <вставить MAGIC-COOKIE>, скопированные из профиля пользователя mininet.

Таким образом, чтобы графические программы из оболочки эмулятора, который требует запуска от имени администратора, могли нормально запускаться через SSH соединение с пробросом X-сессии, открытое пользователем mininet, нужно:

1. После создания такого соединения посмотреть MAGIC-COOKIE из профиля пользователя mininet командой **xauth list**
2. Переключиться в профиль администратора командой: **sudo -i** или **sudo su -**
3. Скопировать MAGIC-COOKIE в профиль администратора командой: **xauth add** <вставить MAGIC-COOKIE>

При возникновении ошибки отсутствия файла для хранения MAGIC-COOKIE в профиле root-пользователя, создать этот файл с именем, которое выводится в сообщении ошибки, командой **touch**.

4. После этого можно запустить эмулятор sudo mn и выполнять в нем вызов графических программ, например, xterm.

ВЫПОЛНЕНИЕ РАБОТЫ

1. Запустите виртуальную машину Mininet. Проверьте IP-адрес на сетевом интерфейсе виртуальной машины и его доступность с основной машины. Для выведения информации об IP-адресах интерфейсов используйте команду:

ip address или

ifconfig

2. Запустите эмулятор Mininet командой

sudo mn

Эмулятор по умолчанию создает простую топологию из двух узлов с именами h1 и h2, подключенных к коммутатору. Командой net можно посмотреть структуру созданной сети, рис. 4.

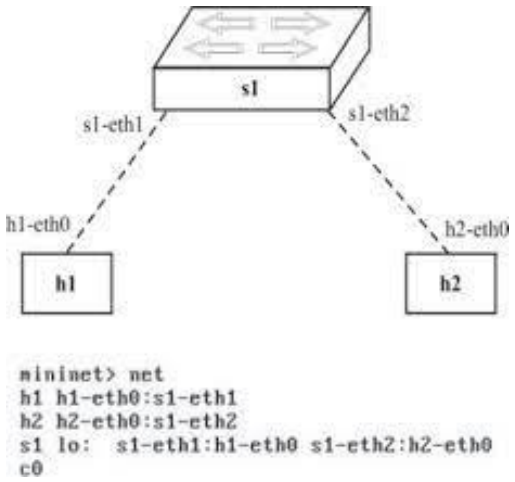


Рис. 4 Топология сети, создаваемой Mininet по умолчанию

3. Откройте окно командной строки на хостах эмулируемой сети командой:
xterm h1 h2

4. Запустите программу Wireshark для захвата трафика между h1 и h2 из командной строки mininet командой:
sh wireshark

5. Сгенерируйте широковещательный трафик в эмулируемой сети. Это можно сделать отправив эхо-запрос командой ping с узла h1 по IP-адресу узла h2. В первый момент узел h1 не знает MAC-адреса узла h2, который необходим ему для инкапсуляции эхо-запроса в Ethernet-кадр. Поэтому h1 автоматически запустит протокол ARP, выполняющий широковещательную рассылку с попыткой найти соответствующий MAC-адрес для IP-адреса h2.

6. Остановите захват пакетов в Wireshark, нажав на красную квадратную пиктограмму остановки. Сохраните результат в файл **broadcast.pcapng**. Не закрывайте программу!

7. Повторно запустите захват трафика между h1 и h2, опять нажав на пиктограмму плавника акулы.

8. Для генерации группового трафика на узле h1 запустите сервер iperf командой:
iperf -s -u -B 239.0.0.1

На узле h2 добавьте в таблицу маршрутизации запись для отправки группового трафика:
ip route add 224.0.0.0/4 dev h2-eth0

На узле h2 запустите клиент iperf, создающий трафик и отправляющий его в группу 239.0.0.1 командой:

iperf -u -c 239.0.0.1 -b 50m -t 10

9. Остановите захват пакетов в Wireshark. Сохраните результат в файл multicast.pcapng.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какой адрес имеет широковещательный пакет ARP на канальном уровне?
2. Какой адрес имеет пакет ARP на сетевом уровне?
3. Какой адрес имеет пакет группового вещания на канальном уровне?
4. Какой адрес имеет пакет группового вещания на сетевом уровне?

ПРАКТИКА №2
НАСТРОЙКА СКОРОСТИ, ДУПЛЕКСА И ДРУГИХ ПАРАМЕТРОВ СЕТЕВЫХ ИНТЕРФЕЙСОВ ETHERNET

Цель работы: изучить режимы работы сетевых интерфейсов Ethernet и характеристики, регистрируемые драйвером сетевого интерфейса, в частности режимы выбора скорости и дуплекса.

ОБЩАЯ ИНФОРМАЦИЯ

Операционная система Linux. Работа с ifconfig

В операционной системе Linux есть известные утилиты для работы с сетевыми интерфейсам. Одна из них — утилита ifconfig.

Вывод команды ifconfig имеет общий формат следующего вида:

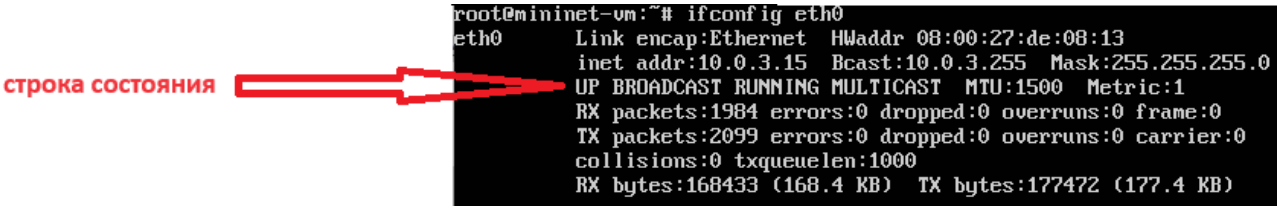


Рис.1 Вывод команды ifconfig на Linux-дистрибутивах семейства Debian, Ubuntu.

1. Строка MAC-адреса

Показывает MAC-адрес адаптера и тип инкапсуляции. Тип инкапсуляции относится к методу упаковки данных на канальном уровне. Сегодня тип инкапсуляции не имеет большого значения, поскольку почти все интерфейсы это Ethernet-интерфейсы. Ethernet выиграл битву за господство, как технология канального уровня. Раньше вы могли встретить такие инкапсуляции, как ax25 (любительское радио X.25) или протокол PPP, «точка-точка» на последовательных каналах, или Token Ring на интерфейсе Token Ring.

2. Информационная строка IP-адреса

Вторая строка вывода ifconfig включает адрес IPv4 или IPv6, настроенный для интерфейса. Для адреса IPv4 также отображаются настроенная сетевая маска и широковещательный адрес.

3. Строка состояния

Отображает флаги состояния, связанные с интерфейсом. Флаг UP показывает, что интерфейс включен. Флаг BROADCAST — то что он поддерживает передачу широковещательных сообщений. Флаг RUNNING демонстрирует что через интерфейс могут передаваться данные, т. е. протокол канального уровня работоспособен. Флаг MULTICAST — индикатор того что групповые сообщения тоже поддерживаются. Кроме того, строка состояния включает максимальную единицу передачи (MTU), это размер поля данных в кадре Ethernet. Значение по умолчанию 1500 байт и метрику для интерфейса. Метрика используется протоколом информации о маршрутизации (RIP) для построения таблиц маршрутизации в сети. Изменить метрику командой ifconfig нельзя, для этого нужно

использовать команды изменения таблицы маршрутизации.

Далее несколько строк показывают статистику по интерфейсу.

RX packets — общее число полученных пакетов.

TX packets — общее число отправленных пакетов.

Строка «Ошибки» для каждого направления, приема и передачи. **Errors:** общее число ошибок. **Dropped** — число отброшенных пакетов. Пакеты отбрасываются, если они «неправильные», например, если получен пакет IPv6 на интерфейсе, не поддерживающем IPv6.

Overruns — число случаев переполнения буфера приема пакетов на интерфейсе.

В потоке принятых пакетов «ошибка кадра», **Frames:** число ошибочных кадров. Такими кадрами могут быть кадры с неверной контрольной суммой, кадры, размер которых не кратен 8 битам (любой кадр переносит целое число байт). Такого рода проблемы чаще всего связаны с плохим кабелем или разъемом на стороне устройства или коммутатора.

В потоке переданных пакетов «ошибка потери несущей частоты», **Carrier**. Может быть вызвана сильными внешними помехами (удар молнии), неправильной настройкой режима дуплекса на одной из взаимодействующих сторон, проблемами с физическим соединением в кабеле.

Collisions: число зарегистрированных коллизий, возникших при передаче данных.

Txqueuelen: размер буфера на передаче в байтах.

В последней строке показано общее число полученных и переданных байт.

В операционных системах семейства RedHat Linux и классических UNIX, немного другой вид вывода команды ifconfig. Первой строкой отображаются флаги, Рис. 2.

```
[phoenixmap@localhost ~]$ ifconfig -a
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::dcea:a867:893a:9ca0  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:45:31:34  txqueuelen 1000  (Ethernet)
        RX packets 10515  bytes 14502986 (13.8 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2644  bytes 163130 (159.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Рис. 2 Вывод команды ifconfig в RedHat, CentOS Linux и UNIX

С помощью утилиты ifconfig можно включить или выключить интерфейс, рис. 3:

```
# ifconfig eth0 up
OR
# ifup eth0
```

```
# ifconfig eth0 down
OR
# ifdown eth0
```

Рис. 3 Включение и выключение интерфейса командой ifconfig

Можно настроить IP-адрес на интерфейсе, рис. 4:

```
ifconfig eth0 172.16.25.125
```

Рис. 4 Настройка адреса на интерфейсе командой ifconfig

Если указать только адрес, то маска будет назначена по классовой схеме, т. е. будет определен класс сети для назначенного адреса и применена стандартная маска для этого класса.

Отдельно изменить маску можно командой рис. 5:

```
ifconfig eth0 netmask 255.255.255.224
```

Рис. 5 Изменение маски для назначенного на интерфейсе адреса

Можно назначить широковещательный адрес, рис. 6, или все параметры настройки IP сразу, рис. 7.

```
ifconfig eth0 broadcast 172.16.25.63
```

Рис. 6 Назначение широковещательного адреса на интерфейсе

```
ifconfig eth0 172.16.25.125 netmask 255.255.255.224 broadcast 172.16.25.63
```

Рис. 7 Полная настройка IP на интерфейсе

Можно изменить значение MTU, рис. 8:

```
ifconfig eth0 mtu 1000
```

Рис. 8 Изменение MTU с помощью ifconfig

Однако, изменить скорость работы или режим дуплекса с помощью команды ifconfig нельзя. Текущие значения этих параметров также не выводятся.

Операционная система Linux. Работа с утилитой ip

Вторая утилита для работы с сетевыми интерфейсами это ip из пакета iproute2. Данная утилита более современная. Именно она предустановлена по умолчанию на актуальных версиях дистрибутивов Linux. Утилита имеет широкий спектр возможностей и более универсальна, по сравнению с ifconfig. Параметры физического и канального уровней отображаются и настраиваются с помощью подкоманды ip link. Так команда

ip link show

выведет параметры, практически аналогичные тем, что отображаются командой ifconfig для интерфейса, рис. 9:

```
mininet@mininet-vm:~$ ip link show
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 fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 08:00:27:a7:17:94 brd ff:ff:ff:ff:ff:ff
```

Рис. 9 Вывод команды ip link show для сетевых интерфейсов

За исключением того, что подкоманда ip link не выводит параметры настроек сетевого уровня, т. е. IP-адреса. Здесь также отображаются флаги. Флаги BROADCAST и MULTICAST по значению совпадают со значением в команде ifconfig. Флаг UP после MULTICAST показывает, что на интерфейсе настроен IP-адрес, флаг LOWER_UP демонстрирует, что ниже лежащий уровень, т. е. канальный, тоже в готовности: кабель подключен и есть несущая.

Для вывода параметров интерфейсов можно использовать сокращенную запись, указав только первую букву подкоманды и не записывая совсем show: **ip l**

Команда имеет специальный ключ -c включающий режим цветовой подсветки вывода, более наглядный для человеческого глаза, рис. 10:

```
mininet@mininet-vm:~$ ip -c route
default via 10.0.2.2 dev eth0 proto dhcp src 10.0.2.15 metric 100
10.0.2.0/24 dev eth0 proto kernel scope link src 10.0.2.15
10.0.2.2 dev eth0 proto dhcp scope link src 10.0.2.15 metric 100
```

Рис. 10 Включение цветовой подсветки вывода в команде ip

Многие подкоманды команды ip поддерживают ключ —brief, дающий сокращенный вывод наиболее важной информации, рис. 11:

```
live@live:~$ ip --brief -c l
lo               UNKNOWN        00:00:00:00:00:00 <LOOPBACK,UP,LOWER_UP>
enp2s0           DOWN           68:f7:28:41:39:6e <NO-CARRIER,BROADCAST,MULTICAST,UP>
wlp3s0           UP             ac:b5:7d:f0:d1:f9 <BROADCAST,MULTICAST,UP,LOWER_UP>
```

Рис. 11 Сокращенный вывод для ip link show, в котором отображается только имя интерфейса, флаги и адрес канального уровня

В целом можно сказать, что команда ip имеет более широкие возможности, по сравнению с ifconfig. Используя эту команду и ее подкоманды, мы можем выполнять все те же настройки, которые делались в ifconfig: включение и выключение интерфейса, настройка IP-адреса, настройка MAC-адреса, изменение значения MTU и т. д., рис. 12.

Old command (Deprecated)	New command
`ifconfig enp6s0 down`	`ip link set enp6s0 down`
`ifconfig enp6s0 up`	`ip link set enp6s0 up`
`ifconfig enp6s0 192.168.2.24`	`ip addr add 192.168.2.24/24 dev enp6s0`
`ifconfig enp6s0 netmask 255.255.255.0`	`ip addr add 192.168.1.1/24 dev enp6s0`
`ifconfig enp6s0 mtu 9000`	`ip link set enp6s0 mtu 9000`
`ifconfig enp6s0:0 192.168.2.25`	`ip addr add 192.168.2.25/24 dev enp6s0`

Рис. 12 Сравнение команд ifconfig и аналогичных команд ip

Кроме того, можно выполнять некоторые настройки, которые командой ifconfig сделать было невозможно, например, добавлять или удалять маршруты из таблицы маршрутизации, изменять размер буфера в очереди передачи и др. Однако, что касается скорости работы интерфейса и режима дуплекса, командой ip также нельзя ни просмотреть, ни изменить этих параметров.

Операционная система Linux. Утилита dmesg

В UNIX-подобных операционных системах для вывода буфера сообщений ядра в стандартный поток вывода (stdout) (по умолчанию на экран) используется утилита (и соответствующая команда) dmesg (сокращение от англ. diagnostic message). Эта команда регистрирует множество событий, происходящих при загрузке и работе операционной системы. Маршрутизация выполняется ядром операционной системы. При включении компьютера, сетевой адаптер настраивается на определенный режим работы и сообщения об этом регистрируются ядром операционной системы. Для фильтрации вывода dmesg можно использовать простой конвейер, формируемый символом вертикальной черты. Такой

конвейер соединяет вывод одной команды с вводом другой. В качестве фильтра можно использовать утилиту grep, выполняющую поиск строк, содержащих заданный шаблон, рис. 13:

```
mininet@mininet-vm:~$ sudo dmesg |grep eth
[    1.239913] e1000 0000:00:03.0 eth0: (PCI:33MHz:32-bit) 08:00:27:a7:17:94
[    1.240182] e1000 0000:00:03.0 eth0: Intel(R) PRO/1000 Network Connection
[   11.178557] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
```

Рис. 13 Вывод сообщений ядра операционной системы, относящихся к настройке сетевого интерфейса с заданным именем, сформированных при загрузке ОС.

Разумеется, таким способом можно посмотреть настройки скорости и дуплекса, но не изменить их. Кроме того, некоторые сетевые адаптеры не будут выдавать сообщений при настройке. Это зависит от управляющего их работой драйвера.

Операционная система Linux. Сервис управления сетью, NetworkManager

Инициализация сетевых интерфейсов при загрузке операционной системы, выполняется специальным сервисом управления сетью. Для этого сервиса есть командная строка в разных видах и возможностях. Например, **networkctl**, с помощью которой мы можем увидеть текущие настройки, рис. 14:

```
mininet@mininet-vm:~$ networkctl status eth0
* 2: eth0
             Link File: /usr/lib/systemd/network/99-default.link
          Network File: /run/systemd/network/10-netplan-eth0.network
                  Type: ether
                 State: routable (configured)
                  Path: pci-0000:00:03.0
                Driver: e1000
                Vendor: Intel Corporation
                 Model: 82540EM Gigabit Ethernet Controller (PRO/1000 MT Desktop Adapter)
            HW Address: 08:00:27:a7:17:94 (PCS Systemtechnik GmbH)
                   MTU: 1500 (min: 46, max: 16110)
  Queue Length (Tx/Rx): 1/1
      Auto negotiation: yes
                 Speed: 1Gbps
                Duplex: full
                  Port: tp
               Address: 10.0.2.15 (DHCP4)
               Gateway: 10.0.2.2
                   DNS: 192.168.0.1

Nov 05 20:24:32 mininet-vm systemd-networkd[288]: eth0: Link UP
Nov 05 20:24:32 mininet-vm systemd-networkd[288]: eth0: Gained carrier
Nov 05 20:24:32 mininet-vm systemd-networkd[288]: eth0: DHCPv4 address 10.0.2.15/24 via 10.0.2.2
```

Рис. 14 Вывод параметров интерфейса через networkctl

Изменения настроек, в том числе для скорости и дуплекса, делаются путем редактирования конфигурационного файла. Есть другая утилита командной строки **nmcli**, обращающаяся к API сервиса и позволяющая читать и редактировать любые параметры интерфейса, рис. 15:

```
mininet@mininet-vm:~$ nmcli -f CAPABILITIES.SPEED dev show eth0
CAPABILITIES.SPEED:                     1000 Mb/s
```

Рис. 15 Вывод скорости интерфейса с помощью nmcli

Эта утилита обладает очень большими возможностями и, в результате, оказывается довольно сложна в практическом применении. На рис. 16 показан вывод сводной

информации по интерфейсу в nmcli, в котором, правда, отсутствуют параметры скорости и дуплекса.

```
mininet@mininet-vm:~$ nmcli dev show eth0
GENERAL.DEVICE:                         eth0
GENERAL.TYPE:                           ethernet
GENERAL.HWADDR:                         08:00:27:A7:17:94
GENERAL.MTU:                            1500
GENERAL.STATE:                          10 (unmanaged)
GENERAL.CONNECTION:                     --
GENERAL.CON-PATH:                       --
WIRED-PROPERTIES.CARRIER:               on
IP4.ADDRESS[1]:                         10.0.2.15/24
IP4.GATEWAY:                            10.0.2.2
IP4.ROUTE[1]:                           dst = 0.0.0.0/0, nh = 10.0.2.2, mt = 100
IP4.ROUTE[2]:                           dst = 10.0.2.0/24, nh = 0.0.0.0, mt = 0
IP4.ROUTE[3]:                           dst = 10.0.2.2/32, nh = 0.0.0.0, mt = 100
IP6.GATEWAY:                            --
```

Рис. 16 Вывод общей информации по интерфейсу с помощью nmcli

Для редактирования параметров в утилите nmcli есть встроенный редактор, в котором можно изменить любое значение. Вызвать этот редактор для конкретного интерфейса можно командой: **nmcli connection edit ethernet-enp0s8**

Для настройки именно значений скорости, дуплекса и еще некоторых характеристик интерфейса есть специализированные программы: **mii-tool** и **ethtool**. Утилита mii-tool уже устарела, т. к. может работать только с 10Мбит/с и 100Мбит/с интерфейсами. Тем не менее, она присутствует во многих дистрибутивах Linux по умолчанию. Запуск без параметров утилиты **mii-tool** покажет текущее состояние всех сетевых линков и их режим работы. На рис. 17 показано, как вывести информацию по конкретному интерфейсу.

```
mininet@mininet-vm:~$ sudo mii-tool eth0
eth0: no autonegotiation, 1000baseT-FD flow-control, link ok
```

Рис. 17 Использование mii-tool для вывода параметров скорости и дуплекса для одного интерфейса

Ключ **-v** даст более подробный вывод. Чтобы установить новые параметры, используется ключ **-F**:

mii-tool -F 100baseTx-FD eth0

Здесь FD — это Full Duplex. С 2003г на смену mii-tool пришла более современная ethtool.
На рис. 18 показан вывод команды ethtool для одного интерфейса.

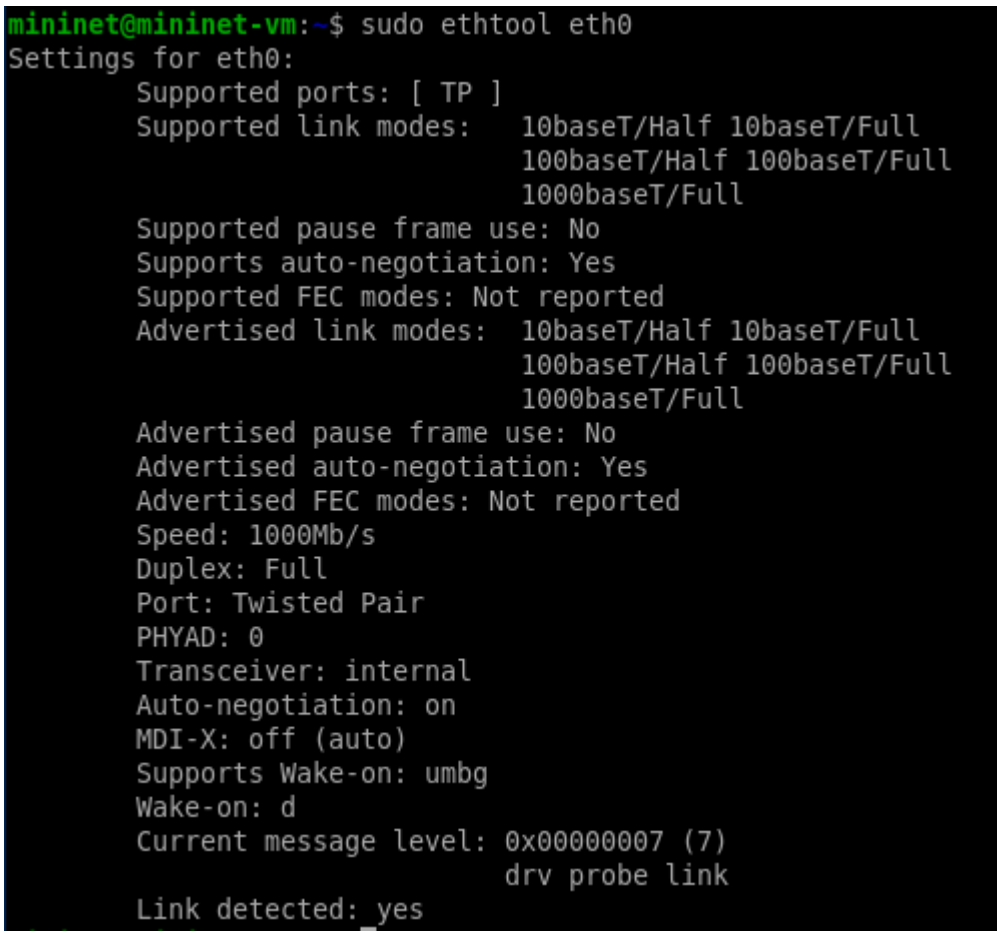


Рис. 18 Отображение настроек скорости и дуплекса интерфейса в команде ethtool

Для изменения значения используется ключ **-s**:

sudo ethtool -s [device_name] autoneg [on/off] speed [10/100/1000] duplex [half/full]

Имейте в виду, что изменения настройки будут работать только на настоящих, железных сетевых интерфейсах. Виртуальные интерфейсы, эмулируемые программно, фактически не имеют таких параметров, как скорость и дуплекс, хотя в выводе команды они могут отображаться, но изменяться при изменении настроек никак не будут.

Операционная система Windows

В ОС Windows информация о скорости и режиме дуплекса на сетевом интерфейсе легко доступна в окне параметров интерфейса, рис. 19

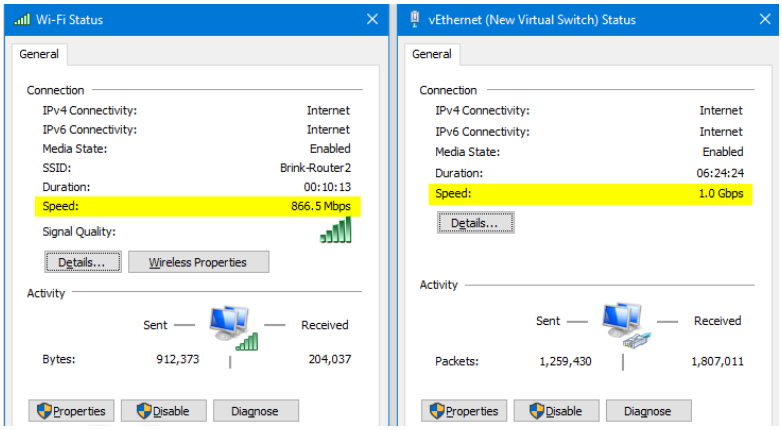


Рис. 19 Отображение скорости работы интерфейса в Windows

И может быть изменена в свойствах через вкладку «Расширенные настройки» (Advanced), рис. 20 или в командной строке Power Shell, рис. 21.

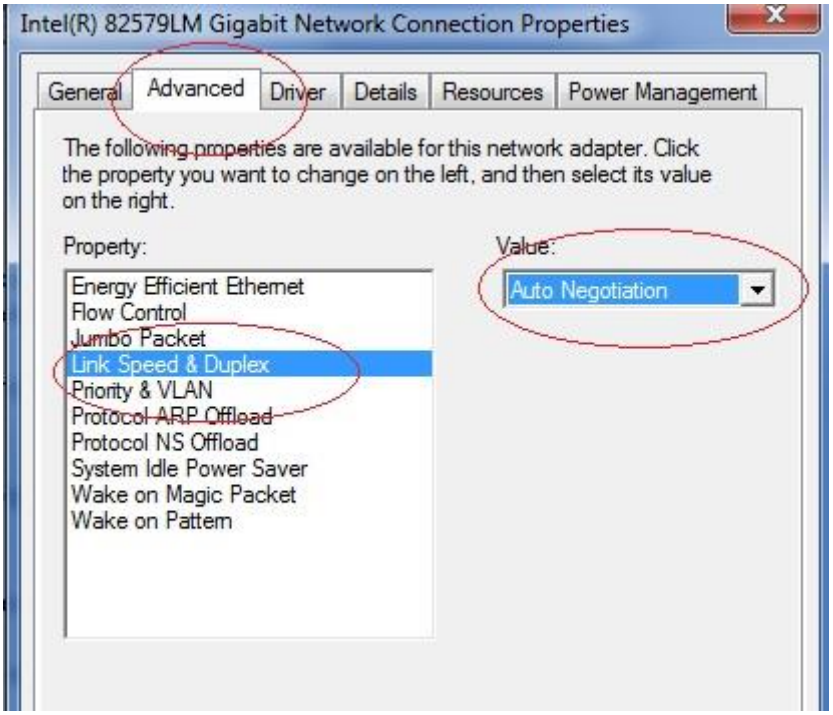


Рис. 20 Изменение мшкепараметров скорости/дуплекса в Windows

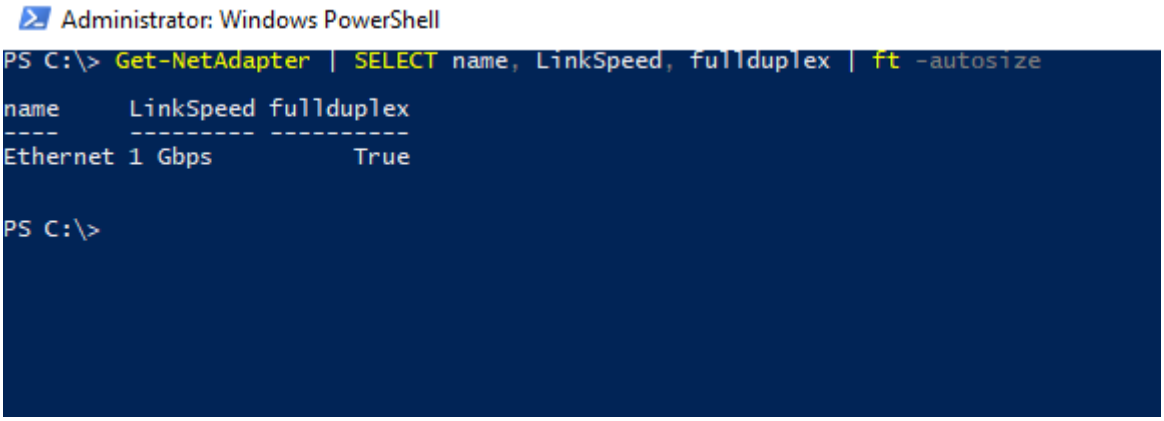


Рис. 21 Просмотр параметров скорости и дуплекса в Power Shell

Операционная система RouterOS

В операционной системе RouterOS, созданной на основе Linux и работающей на маршрутизаторах и многофункциональных устройствах Mikrotik команда:

interface ethernet monitor <номер интерфейса>

выводит параметры скорости и дуплекса и ряд дополнительных параметров, рис. 22. Если мы хотим увидеть параметры всех интерфейсов, то вместо номера нужно ввести ключевое слово [**find**] в квадратных скобках, рис. 23.

```
[admin@MikroTik] > interface ethernet monitor 0
                     name: ether1
                   status: link-ok
          auto-negotiation: done
                     rate: 1Gbps
               full-duplex: yes
            tx-flow-control: no
            rx-flow-control: no
                advertising: 10M-half,10M-full,100M-half,100M-full,1000M-full
  link-partner-advertising:
     default-cable-setting: standard
```

Рис. 22 Вывод настроек скорости и дуплекса на интерфейсе Mikrotik

```
[admin@MikroTik] > interface ethernet/monitor [find]
                         name: ether1     ether2
                       status: link-ok    link-ok
            auto-negotiation: done       done
                         rate: 1Gbps      1Gbps
                 full-duplex: yes         yes
              tx-flow-control: no          no
              rx-flow-control: no          no
                  advertising: 10M-half    10M-half
                               10M-full    10M-full
                               100M-half   100M-half
                               100M-full   100M-full
                               1000M-full  1000M-full
    link-partner-advertising:
       default-cable-setting: standard    standard
```

Рис. 23 Вывод параметров всех интерфейсов на устройстве Mikrotik

Подробную информацию по параметрам канального уровня можно получить командой:

interface ethernet print detail

Как показано на рис. 24. В отличие от Linux, информация о скорости и дуплексном режиме интерфейса выводится этой же командой. Каких-то дополнительных команд для этого вводить не требуется.

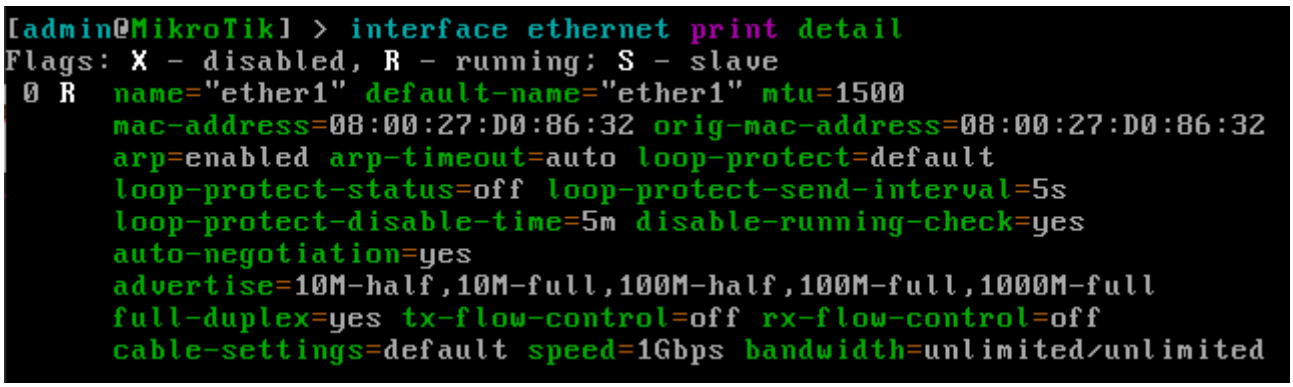


Рис. 24 Детальная информация о настройках интерфейса Mikrotik, включая скорость и дуплекс.

Изменить параметры скорости и дуплекса можно командой:

```
/interface ethernet set [find default-name=ether1] auto-negotiation=no speed=10M full-duplex=no
```

Здесь параметры меняются на интерфейсе ether1, отключается режим полного дуплекса и задается скорость 10 Мбит/с. Параметр auto-negotiation это протокол автоматического согласования настроек со второй стороной на другом конце линка (это может быть коммутатор или другое устройство), если протокол автоматического согласования не отключить, то изменить настройки не удастся. По этому протоколу оба устройства выбирают режим, который они оба поддерживают с наилучшими характеристиками для передачи данных, т. е. максимально возможная скорость и режим полного дуплекса, если он поддерживается.

Операционная система Cisco IOS

На коммутаторах и маршрутизаторах Cisco Systems есть команда проверки состояния интерфейса:

show interfaces

Если ввести ее без параметров, то мы получим вывод информации по всем существующим интерфейсам. С указанием имени конкретного интерфейса, вывод будет только для него, рис. 25.

```
GigabitEthernet0/0 is up, line protocol is up (connected)
  Hardware is CN Gigabit Ethernet, address is 0001.634a.3201 (bia 0001.634a.3201)
  Internet address is 2.2.2.2/8
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mb/s, media type is RJ45
  output flow-control is unsupported, input flow-control is unsupported
  ARP type: ARPA, ARP Timeout 04:00:00,
  Last input 00:00:08, output 00:00:05, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0 (size/max/drops); Total output drops: 0
  Queueing strategy: fifo
  Output queue :0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 watchdog, 1017 multicast, 0 pause input
     0 input packets with dribble condition detected
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

Рис. 25 Вывод информации об интерфейсе на устройстве Cisco

Здесь отдельной строкой показан режим дуплекса, в котором находится интерфейс и скорость на которой он работает. Но это так для современных устройств, можно сказать, что это результат постепенной эволюции. На более старых маршрутизаторах результат выполнения той же команды не покажет строку скорости и дуплекса, рис. 26.

```
Router#sh int fa0/0
FastEthernet0/0 is up, line protocol is up (connected)
  Hardware is Lance, address is 0090.2bd2.9101 (bia 0090.2bd2.9101)
  Internet address is 1.1.1.2/8
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  ARP type: ARPA, ARP Timeout 04:00:00,
  Last input 00:00:08, output 00:00:05, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0 (size/max/drops); Total output drops: 0
  Queueing strategy: fifo
  Output queue :0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 input packets with dribble condition detected
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

Рис. 26 Вывод команды свойств интерфейса на маршрутизаторе предыдущего поколения

Увидеть режим дуплекса и скорости на интерфейсе в таком случае можно, просмотрев конфигурацию маршрутизатора командой **show running-config**, рис 27.

```
interface FastEthernet0/0
 ip address 1.1.1.2 255.0.0.0
 duplex auto
 speed auto
```

Рис. 27 Настройки дуплекса и скорости интерфейса в конфигурации устройства

Правда, как в данном случае, мы видим, что интерфейс работает в режиме автоматического согласования обоих параметров, но не сможем понять, какой именно режим дуплекса и скорости согласован с партнером. Для выяснения этой информации мы можем использовать протокол CDP, Cisco Discovery Protocol, который позволяет Cisco устройствам обнаруживать друг друга в сети и обмениваться информацией. Администратор, используя этот протокол, может построить «карту» сети, если у него ранее не было соответствующей документации. Команда:

show cdp neighbours detail

покажет развернутую информацию о подключенном на другой стороне линии устройстве, в том числе о режиме дуплекса, рис.28:

```
Router#show cdp neighbors detail

Device ID: Switch
Entry address(es):
Platform: cisco 2950, Capabilities: Switch
Interface: FastEthernet0/0, Port ID (outgoing port): FastEthernet0/1
Holdtime: 135

Version :
Cisco Internetwork Operating System Software
IOS (tm) C2950 Software (C2950-I6Q4L2-M), Version 12.1(22)EA4, RELEASE
SOFTWARE(fc1)
Copyright (c) 1986-2005 by cisco Systems, Inc.
Compiled Wed 18-May-05 22:31 by jharirba

advertisement version: 2
Duplex: full
```

Рис. 28 Настройки дуплекса в выводе команды соседних устройств по протоколу CDP

ВЫПОЛНЕНИЕ РАБОТЫ

В практической работе вы изменяете настройки скорости работы интерфейса и режима дуплекса в трех разных сценариях:

1. Для устройств Cisco.
2. Для устройств Mikrotik.
3. Для Linux.

Первый сценарий

Первый сценарий можно выполнить в программе Cisco Packet Tracer. Рекомендуется использовать новую версию: 8.2, 8.2.1. Установочный файл для Windows можно скачать здесь. Перед установкой отключите Интернет на компьютере. При запуске программа выходит в Интернет и открывает окно авторизации. Нужно войти с аккаунтом студента сетевой академии Cisco или портала навыков для всех. Если отключить Интернет на компьютере на момент запуска программы или создать в межсетевом экране правило, запрещающее выход в

Интернет для этой программы, то программа нормально запускается без авторизации и работает без ограничений. После запуска программы Интернет можно подключить, повторные запросы авторизации не создаются.

В программе Cisco Packet Tracer создайте топологию, как показано на рисунке 29:

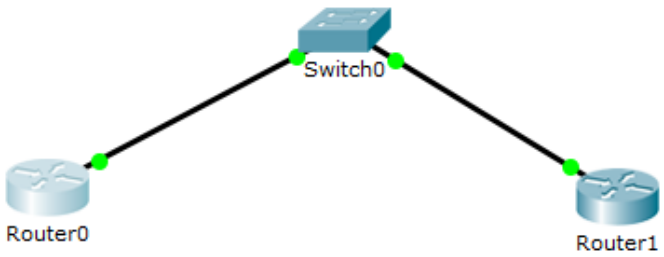


Рис. 29 Топология сети для устройств Cisco

Используйте маршрутизатор 4331 и коммутатор 2960.

Щелкните мышкой на пиктограмму маршрутизатора и перейдите на вкладку настройки Config, рис. 30. Выберите первый сетевой интерфейс в списке и настройте на нем IP-адрес и маску подсети, например, как показано на рис. 31.

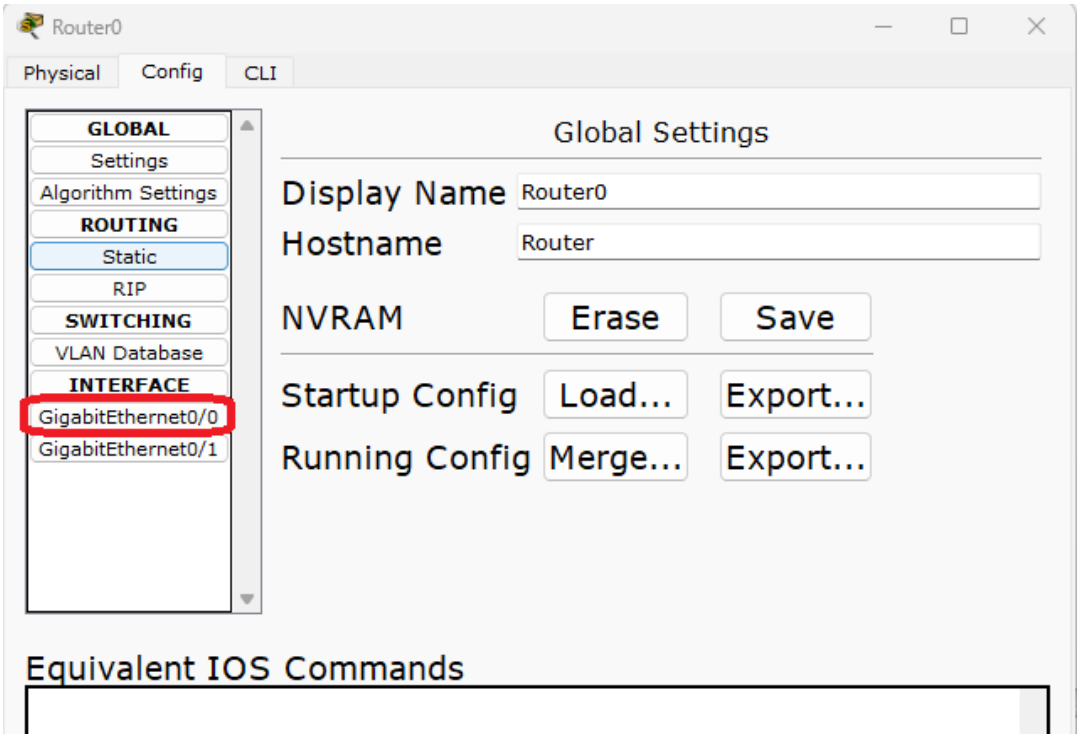


Рис. 30 Выбор интерфейса на кладке настройки

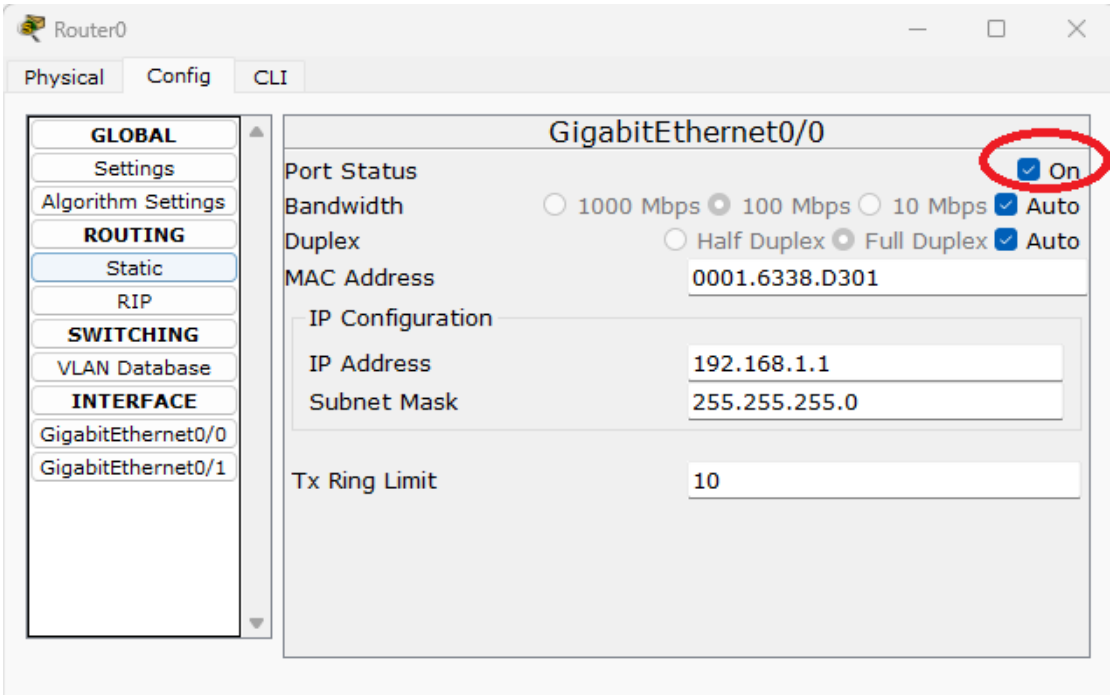


Рис. 31 Настройка IP-адреса и маски подсети интерфейса маршрутизатора

Поставьте галочку включения интерфейса, отмеченную на рис. 31. Аналогичным образом настройте интерфейс на втором маршрутизаторе, задав ему другой адрес в той же сети, например, 192.168.1.2.

Перейдите на вкладку командной строки: CLI — Command Line Interface, рис. 32:

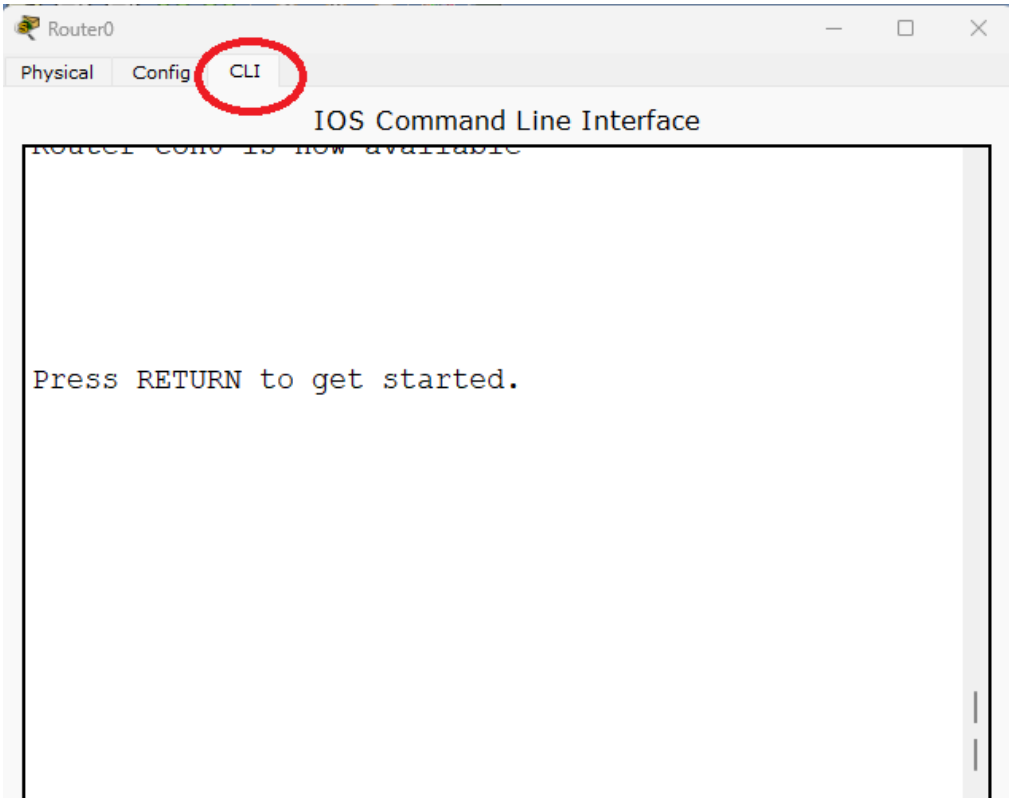


Рис. 30 Вкладка командной строки устройства в Cisco Packet Tracer

Нажмите <Enter>, откроется приглашение командной строки в пользовательском режиме, индикация этого режима — закрывающая треугольная скобка. Перейдите в режим администратора командой **enable**

Режим администратора отображается символом решетки # в приглашении командной строки.

Выполните эхо-запрос по адресу второго маршрутизатора командой:

ping 192.168.1.2

Обратите внимание на время задержки при получении ответных пакетов.

Выполните команду **show interfaces** и проанализируйте режим, в котором находится подключенный к коммутатору интерфейс. Выполните такие же действия на интерфейсе коммутатора, к которому подключен маршрутизатор, и сравните результаты.

Вернитесь на маршрутизатор. Перейдите в режим глобальной настройки командой:

configure terminal

Режим глобальной настройки отображается словом (config) в скобках в приглашении командной строки. Из режима глобальной настройки перейдите в режим настройки интерфейса командой:

interface GigabitEthernet0/0

Режим настройки интерфейса отображается словом (config-if) в приглашении командной строки. В этом режиме измените настройки дуплекса и скорости работы интерфейса, последовательно вводя команды:

duplex half

speed 10

Для возврата в режим администратора введите команду **end**

Повторите команду просмотра состояния интерфейса, изменилось ли значение скорости и дуплекса согласно вашим настройкам? Перейдите на коммутатор и проверьте состояние интерфейса на коммутаторе. Какие изменения вы наблюдаете.

Выполните те же настройки на втором маршрутизаторе.

После того, как световые индикаторы на интерфейсах приобретут зеленый цвет, выполните отправку эхо-запросов командой ping с одного маршрутизатор на другой еще раз. Изменилось ли время отклика?

Второй сценарий

Скачайте с официального сайта Mikrotik образ маршрутизатора для работы в облаке в формате для экспорта в виртуальную машину
https://download.mikrotik.com/routeros/6.49.10/chr-6.49.10.ova [3].

Выполните экспорт, создав виртуальную машину с маршрутизатором Микротик. Выберите тип сетевого соединения: «Внутренняя сеть». Для создания второго маршрутизатора выполните операцию клонирования, рис. 31

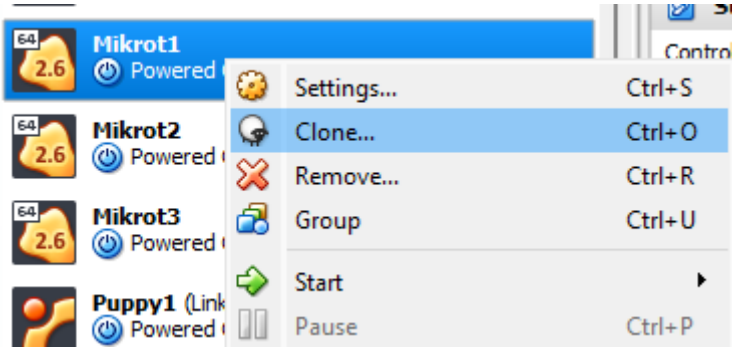


Рис. 31 Клонирование виртуальной машины в Virtualbox

Связанное клонирование — рекомендуемый режим. При этом виртуальные образы занимают меньше места на хост-машине. Запустите обе виртуальные машины с маршрутизаторами Mikrotik. Благодаря выбранному типу сетевого соединения они оба оказываются как бы физически соединены. Логин по умолчанию для этого маршрутизатора **admin** пароль пустой.

Проверьте текущие настройки скорости работы интерфейсов и режим дуплекса на этих маршрутизаторах, как описано в разделе общей информации. Задайте IP-адрес на маршрутизаторе командой:

ip address add address=192.168.1.1/24 interface=ether1

На втором маршрутизаторе измените хостовую часть адреса. Выполните отправку эхо-запросов с одного маршрутизатора на другой командой ping, зафиксируйте время получения отклика.

Измените значение скорости работы и дуплекса на одном маршрутизаторе командой:

/interface ethernet set [find default-name=ether1] auto-negotiation=no speed=10M full-duplex=no

Проверьте, получилось ли изменить режим работы, например, командой

interface ethernet monitor 0

Повторно выполните отправку эхо-запросов в адрес второго маршрутизатора. Изменилось ли время отклика?

Третий сценарий

В программе VirtualBox создайте клон машины Mininet и соедините его с исходной машиной Mininet также соединением «внутренняя сеть», как в предыдущем случае.

Проанализируйте состояние сетевых интерфейсов с помощью стандартных утилит **ifconfig** и **ip**.

Настройте IP-адреса на интерфейсах и проверьте связь между машинами.

В виртуальной машине Mininet утилита ethtool уже установлена. Посмотрите, какие значения используются для скорости интерфейса и режима дуплекса, какие возможны варианты. Введите команду:

sudo ethtool eth0

На одной машине измените скорость работы интерфейса и режим дуплекса утилитой ethtool, как это описано в разделе общей информации. Задайте скорость 10 Мбит/с и

отключите полный дуплекс. Еще раз выполните **sudo ethtool eth0**. Изменилось ли значение скорости и дуплекса в соответствии с заданными значениями?

ПРАКТИКА №3
НАСТРОЙКА VLAN, VTP И МАРШРУТИЗАЦИИ МЕЖДУ VLAN

Общие сведения

Для проверки настроек VLAN на коммутаторе Cisco используйте команду

S1# show vlan brief

Или

S1# show vlan

Порты коммутатора Cisco могут быть в четырех состояниях: Dynamic Auto, Dynamic Desirable, Trunk и Access. В таблице рис. 1 на перекрестии столбцов показано итоговое состояние линка при соответствующих режимах на его концах.

	Dynamic Auto	Dynamic Desirable	Trunk	Access
Dynamic Auto	Access	Trunk	Trunk	Access
Dynamic Desirable	Trunk	Trunk	Trunk	Access
Trunk	Trunk	Trunk	Trunk	Limited Connectivity
Access	Access	Access	Limited Connectivity	Access

Рис. 1 Состояния линка при соответствующих режимах на его концах.

Например, если на двух соединенных коммутаторах оба конца соединения будут в состоянии Dynamic Auto, линк будет в режиме Access. По умолчанию интерфейсы на Cisco коммутаторе в состоянии Dynamic Auto.

Для перевода интерфейса в режим Access нужно выполнить команду:

switchport mode access

Для перевода интерфейса в режим Trunk нужно выполнить команду:

switchport mode trunk

в контексте настройки интерфейса или группы интерфейсов. Для настройки группы интерфейсов используйте команду:

interface range f0/1 -10

Для создания VLAN используйте команды

S1(config)# vlan 10

S2(config-vlan)# name ВашеИмяVlan

Для включения интерфейса в нужную VLAN используйте команду:

S1(config-if)# switchport access vlan 10

С указанием номера соответствующей VLAN.

Для проверки настроек транковых интерфейсов на коммутаторе Cisco используйте команду

S2# show interfaces trunk

Протокол VTP - VLAN Trunking Protocol — проприетарный протокол компании Cisco Systems, предназначенный для создания, удаления и переименования VLAN на сетевых устройствах. Передавать информацию о том, какой порт находится в каком VLANe, он не может.

Коммутаторы по отношению к протоколу VTP могут быть в трех состояниях:

VTP Server

VTP Client
VTP Transparent

Настройка единой конфигурации VLAN выполняется для общего домена, который задается в конфигурации VTP. Для снижения ошибок при создании конфигурации VLAN вы можете настроить один из коммутаторов в домене в режим VTP Server и на нем создавать все необходимые VLAN. Остальные коммутаторы в том же домене перевести в режим VTP Client. Все созданные на сервере VTP VLAN будут создаваться на клиентских коммутаторах автоматически. Остается только включить в них необходимые интерфейсы. Конфигурация VLAN хранится в отдельном файле базы данных VLAN, а не в конфигурационном файле основной конфигурации. Для защиты доступа к файлу базы данных VLAN в настройках VTP может быть задан пароль. Есть три версии протокола. Наиболее совершенной и защищенной является последняя версия, VTPv3.

Подобный по характеристикам открытый протокол, разработанный IEEE – GVRP (GARP VLAN Registration Protocol) для регистрации транковых VLAN между коммутаторами. Сейчас этот протокол уже считается устаревшим, хотя его еще можно встретить на многих устройствах. Он заменен обновленным протоколом MVRP - Multiple VLAN Registration Protocol. Все эти изменения вошли в обновлённый стандарт 802.1Q.

По умолчанию все Cisco коммутаторы работают, как VTP Server при этом не имеют в конфигурации VTP имени домена (значение Null). Если имя домена одинаковое, то два сервера VTP тоже будут обмениваться информацией. Чья информация важнее определяется специальным параметром - номером ревизии. Этот параметр увеличивается с каждым созданием VLAN на устройстве. Тот VTP Server, у которого больше номер ревизии считается, что имеет более свежую информацию, и она будет копироваться с него на другие коммутаторы в том же домене.

Настройка режима VTP выполняется из глобальной конфигурации:

S1(config)# vtp mode server

Задание имени домена

S1(config)# vtp domain gr720

Задание пароля на базу VLAN

S1(config)# vtp password tusur

Для маршрутизации через один физический интерфейс, этот интерфейс между маршрутизатором и коммутатором должен быть переведен в транковый режим. На нем создаются подынтерфейсы. Например, командой:

interface FastEthernet0/0.10

Указание дополнительного числа, отделенного точкой после номера интерфейса означает создание логического подынтерфейса. Сам номер может быть любым, но часто при создании подынтерфейсов используют на нем тот же номер, что и номер VLAN, для которой создается интерфейс. Прежде чем настраивать адрес на интерфейсе, нужно указать тип инкапсуляции, т.к. существует несколько разных типов.

R1(config-subif)# encapsulation dot1Q 10

Здесь dot1Q это стандарт IEEE802.1Q, а номер – это номер VLAN, для которой создается интерфейс. Если в имени подынтерфейса номер не влияет на работу и нужен для информации самому администратору, то в настройке инкапсуляции именно этот номер и привязывает к подынтерфейсу соответствующую VLAN.

После этого можно настроить адрес на подынтерфейсе:

R1(config-subif)# ip address 192.168.1.1 255.255.255.0

Когда закончена настройка всех подынтерфейсов, нужно включить интерфейс:

R1(config-subif)# exit
R1(config)# interface g0/1
R1(config-if)# no shutdown

Подынтерфейсы включать не нужно, они включаются автоматически. На интерфейсе никаких адресов настраивать не нужно. Маршрутизация между VLAN будет выполняться автоматически, т.к. они сопряжены с подключенными к маршрутизатору подынтерфейсами.

ВЫПОЛНЕНИЕ РАБОТЫ

Работа выполняется в программе Cisco Packet Tracer версии 6.2 или выше.

Создайте топологию, как показано на рис.2 Используйте коммутаторы Cisco 2960, тип маршрутизатора может быть любым.

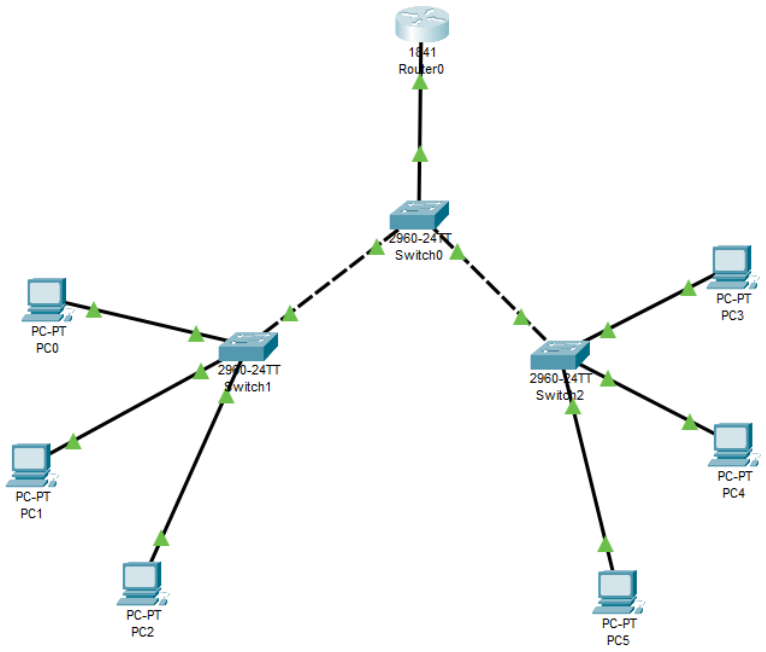


Рис. 2 Топология создаваемой сети

На коммутаторе Switch0 создайте VLAN для управления номер 99, а также две VLAN с номерами 10 и 20 и соответствующими именами 1) ваше имя и 2) ваша фамилия

На коммутаторе Switch0 настройте домен VTP с названием вашей группы, задайте пароль на файл базы данных VLAN

Коммутаторы Switch1 и Switch2 переведите в режим клиента VTP и включите в домен, созданный на шаге 3.

Интерфейсы между коммутаторами и между коммутатором и маршрутизатором переведите в режим транка.

Интерфейсы, к которым подключены компьютеры, переведите в режим доступа (access).

На коммутаторе Switch1 включите интерфейсы, к которым подключены компьютеры, в VLAN10 ваше имя

На коммутаторе Switch2 включите интерфейсы, к которым подключены компьютеры, в VLAN20 ваша фамилия

На маршрутизаторе создайте подынтерфейсы, соответствующие созданным VLAN, назначьте на подынтерфейсы первые адреса из двух подсетей 192.168.x.0/24 и 192.168.x+1.0/24. Где x ваш номер в списке группы.

На компьютеры, подключенные к Switch1, назначьте произвольные адреса из первой подсети, на компьютеры, подключенные к Switch2, произвольные адреса из второй подсети. В качестве адреса шлюза укажите соответствующий адрес подынтерфейса маршрутизатора.

Проверьте связность сети. Отправьте эхо-запрос от любого компьютера левой подсети к любому компьютеру в правой подсети.

ПРАКТИКА №4
СЕГМЕНТАЦИЯ СЕТИ С ПОМОЩЬЮ ПОДСЕТЕЙ, НАСТРОЙКА МАРШРУТИЗАЦИИ

Цель работы – Освоить принципы, инструменты и методики разделения выделенного предприятию адресного блока на подсети. Получить практические навыки разделения сетей на нужное количество подсетей нужного размера. Настроить статическую маршрутизацию для рассчитанных подсетей в виртуальной сети из 3 виртуальных машин.

Общая информация

IP-адрес в 4 версии протокола IP имеет размер 32 бита. При этом в адресе содержится две части: адрес сети и адрес устройства, часто говорят «хоста», рис. 1

Рис. 1 Сетевая и хостовая части IP-адреса

Для создания иерархии больших и малых компьютерных сетей сначала использовали классовую схему, рис. 2

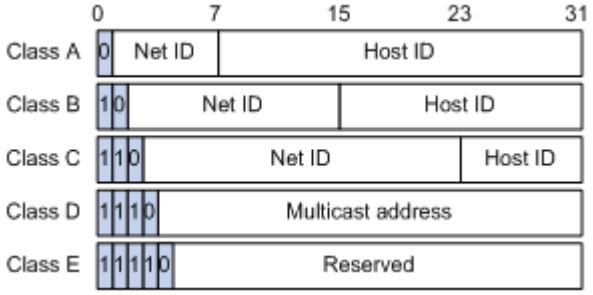


Рис. 2 Классы сетей IPv4

Было определено пять классов A, B, C, D и E. Класс E экспериментальный, поэтому практически в сетях адреса этого класса не применяются. Класс D — групповое вещание, специальный случай. В адресах группового вещания нет сетевой и хостовой части, 28 младших разрядов этого класса это номер группы, а старшие 4 разряда имеют фиксированное значение. На практике мы чаще всего встречаемся с адресами сетей классов A, B и C.

Для сетей класса A сетевая часть адреса это старшие 8 разрядов или старший байт, оставшиеся 24 разряда это хостовая часть. Это очень большая сеть, больше 16 млн. адресов. Для сетей класса B сетевая часть 16 разрядов, такого же размера хостовая часть, это сети среднего размера. Малые сети класса C имеют в сетевой части 24 разряда и только 8 в хостовой, что дает 256 адресов для хостов.

Адреса IPv4 записываются в десятично-точечной нотации, когда байты адреса отделяются друг от друга точками, а двоичные значения байт переводятся в десятичную форму. К какому классу относится адрес определяется по значению старшего байта. Как показано на рис. 2. Для сетей класса A самый старший бит старшего байта всегда равен 0.

Это значит что диапазон значений старшего байта IP адреса для сетей класса A лежит от 0 до 127. Для сетей класса B зафиксированы значения двух старших бит старшего байта, они равны 10, поэтому диапазон значений старшего байта для сетей класса B от 128 до 191. Для сетей класса C зафиксированы три старших бита, поэтому диапазон значений старшего байта от 192 до 223. Таким образом, просто посмотрев на IP адрес, мы можем сразу сказать к какому классу сетей он относится.

Классовая схема давала очень грубое деление на сети разного размера, поэтому вскоре она была усовершенствована. Добавлено дополнительное число — маска подсети для выделения сетевой части адреса. Маска содержит подряд идущие единицы, определяющие длину сетевой части, которые сменяются нолями в хостовой части адреса. Для выделения из адреса сетевой части, т. е. адреса сети, выполняется операция побитового логического умножения адреса и маски. В результате этой операции обнуляется хостовая часть адреса и мы получаем адрес сети. С появлением маски подсети, деление IP адреса на сетевую и хостовую части можно выполнять с точностью до одного двоичного разряда, что дает достаточную гибкость при выборе размера сети.

Примеры IP адресов:
17.16.2.15 = **00011101**.00010000.00000010.00001111
178.68.128.168 = **10110010.01000100**.10000000.10101000
217.20.147.94 = **11011001.00010100.10010011**.01011110

Здесь часть выделенная красным относится к адресу сети, в которой находится этот хост, на основании классовой схемы. Совместимость с классовой схемой деления сетей по размеру была сохранена, если используемая маска не меняет это значение.

В маске единицы и нули не могут перемешиваться, сначала идут 1, потом 0. Для записи сетевой маски также используется десятично-точечная нотация.
Примеры масок сети:

255.255.255.0 = 11111111.11111111.11111111.00000000 — стандартная маска для класса C
255.0.0.0 = 11111111.00000000.00000000.00000000 — стандартная маска для класса A
255.255.240.0 = 11111111.11111111.11110000.00000000 — нестандартная маска

- сетевая маска - это не IP адрес - она используется для того, чтобы выделить сетевую часть IP адреса;

Сетевой префикс. Префиксная нотация записи маски была предложена в стандарте бесклассовой маршрутизации CIDR. Число в префиксе обозначает количество бит, установленных в 1 и определяющих сетевую часть адреса.

Примеры префиксной записи маски:
/8 = 11111111.00000000.00000000.00000000 соответствует маске 255.0.0.0 в десятично-точечной нотации
/16 = 11111111.11111111.00000000.00000000 соответствует маске 255.255.0.0 в десятично-точечной нотации
/20 = 11111111.11111111.11110000.0000000 соответствует маске 255.255.240.0 в десятично-точечной нотации.

Технология применения масок разной длины, англ. VLSM - Variable Length Subnet Masks описана в RFC 950. Она появилась не сразу, сначала маршрутизаторы умели работать только с классовыми сетями, а если они разбивались на подсети, то все подсети должны были быть одинаковыми. Применение масок переменной длины позволило рациональнее использовать адресное пространство. При делении сети на подсети в структуре адреса появляется дополнительная область, адресующая подсеть, рис. 3

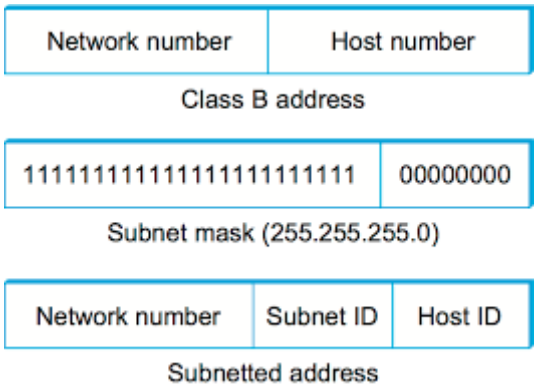


Рис. 3 Деление сети на подсети

Необходимое число разрядов для создания подсетей мы занимаем из хостовой части. Так чтобы организовать две подсети достаточно одного разряда, вместо маски /24 используем /25:

/25 = 11111111.11111111.11111111.10000000 = 255.255.255.128

Исходя из количества бит, отводимых под адрес хоста, можно определить максимальное количество компьютеров в подсети, т. е. ее размер. Так, с помощью 6 бит можно получить 64 адреса (два в шестой степени). Однако первое и последнее число не могут использоваться в качестве адресов хостов, поскольку им назначена особая роль – адрес сети и адрес широковещания. Если в поле для адресации хоста все биты равны нулю, то такой адрес будет адресом сети (подсети) и его нельзя назначить устройству. Так же, если в поле для адресации хоста все биты равны единице, то такой адрес является специальным, широковещательным адресом для данной сети (подсети).

Методика разбиения сети на подсети

На рис. 4 показан пример разбиения сети класса B 172.16.10.0 у которой уже была нестандартная маска 255.255.255.0 на подсети равного размера. При делении на подсети мы можем исходить из необходимого числа подсетей, например для 16 подсетей нам потребуется занять в хостовой части 4 разряда. Но чаще выбор определяется числом пользователей в каждой созданной подсети, ведь сети создаются для пользователей и это число нам известно. В примере на рис. 4 маску изменили на 255.255.255.240, т. е. продлили на 4 разряда. В хостовой части осталось тоже 4 разряда. Этот остаток определяет размер подсети. В 4 разрядах 16 адресов, поэтому первая подсеть 172.16.10.0 с маской 255.255.255.240 закончится адресом 172.16.10.15. Счет идет с 0, поэтому адрес 15 последний, 172.16.10.16 это следующая подсеть с той же маской. Адреса подсетей формируются в четырех старших битах последнего октета, которые были взяты для организации подсетей. Для первой подсети здесь будет значение 0000, для второй 0001, для третьей 0010 и так далее. В десятично-

точечной нотации адреса подсетей будут идти с шагом +16, т. к. таков размер оставшегося хостового пространства.

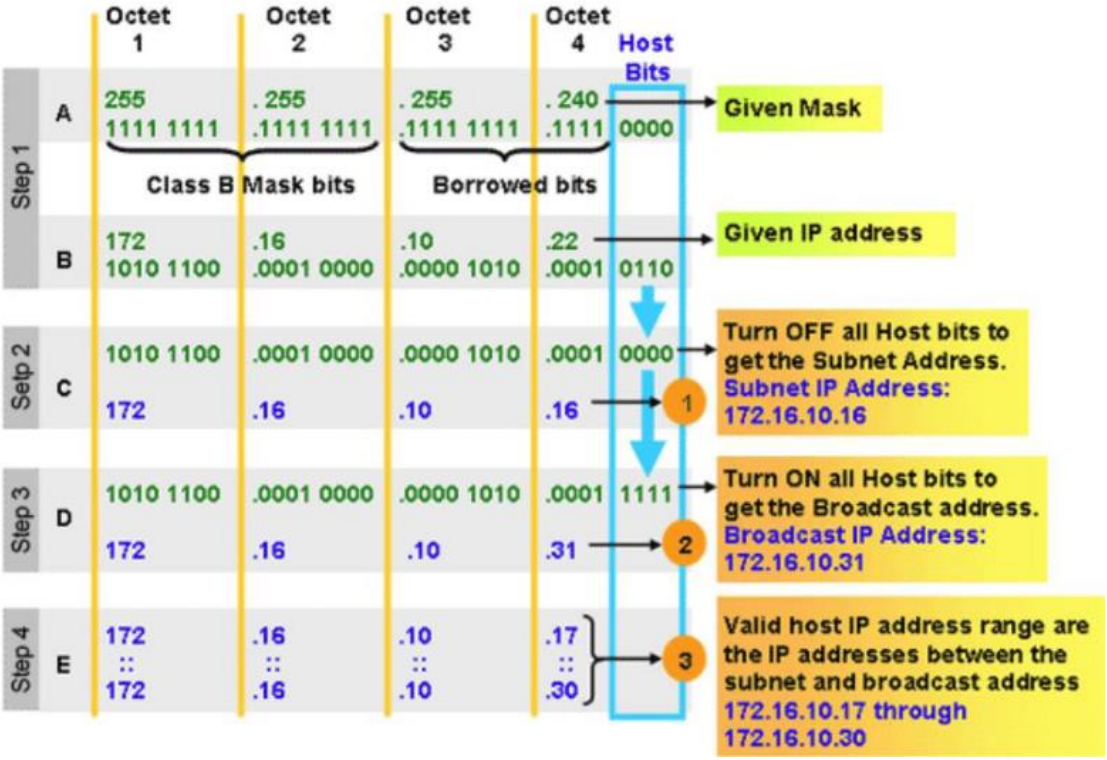


Рис. 4 Разбиение сети 172.16.10.0 на подсети

Пример разбиения сети на несколько разных подсетей.

Пусть на предприятии есть три отдела с известным количеством компьютеров в отделах, трафик между отделами требуется логически и физически разделить.
При делении сети на подсети узлы, находящиеся в различных подсетях, не могут взаимодействовать на сетевом уровне напрямую. Для взаимодействия таких узлов предприятию необходим маршрутизатор, пересылающие пакеты между подсетями.

Схема сети предприятия с тремя маршрутизаторами, тремя коммутаторами и с указанием количества узлов в сегментах показана на рисунке 5.

Пусть на всю сеть компании провайдер выделил один блок IP-адресов класса C 192.168.1.0/24, который требуется разделить на 6 разных по размеру подсетей.

Требуемое количество адресов для пользователей в каждой подсети:
Подсеть A – 100 пользователей
Подсеть B – 50 пользователей
Подсеть C – 20 пользователей

Нужно учесть, что каждая пара маршрутизаторов тоже соединяется между собой отдельной подсетью, поэтому потребуется ещё 3 небольших подсети. Поскольку такие подсети содержат только по два хоста, для них достаточно использовать префикс подсети /30:
Подсеть D - 2 узла
Подсеть E - 2 узла
Подсеть F - 2 узла

Здесь термин «узел» и «хост» можно рассматривать как синонимы, оба они обозначают устройство в сети.

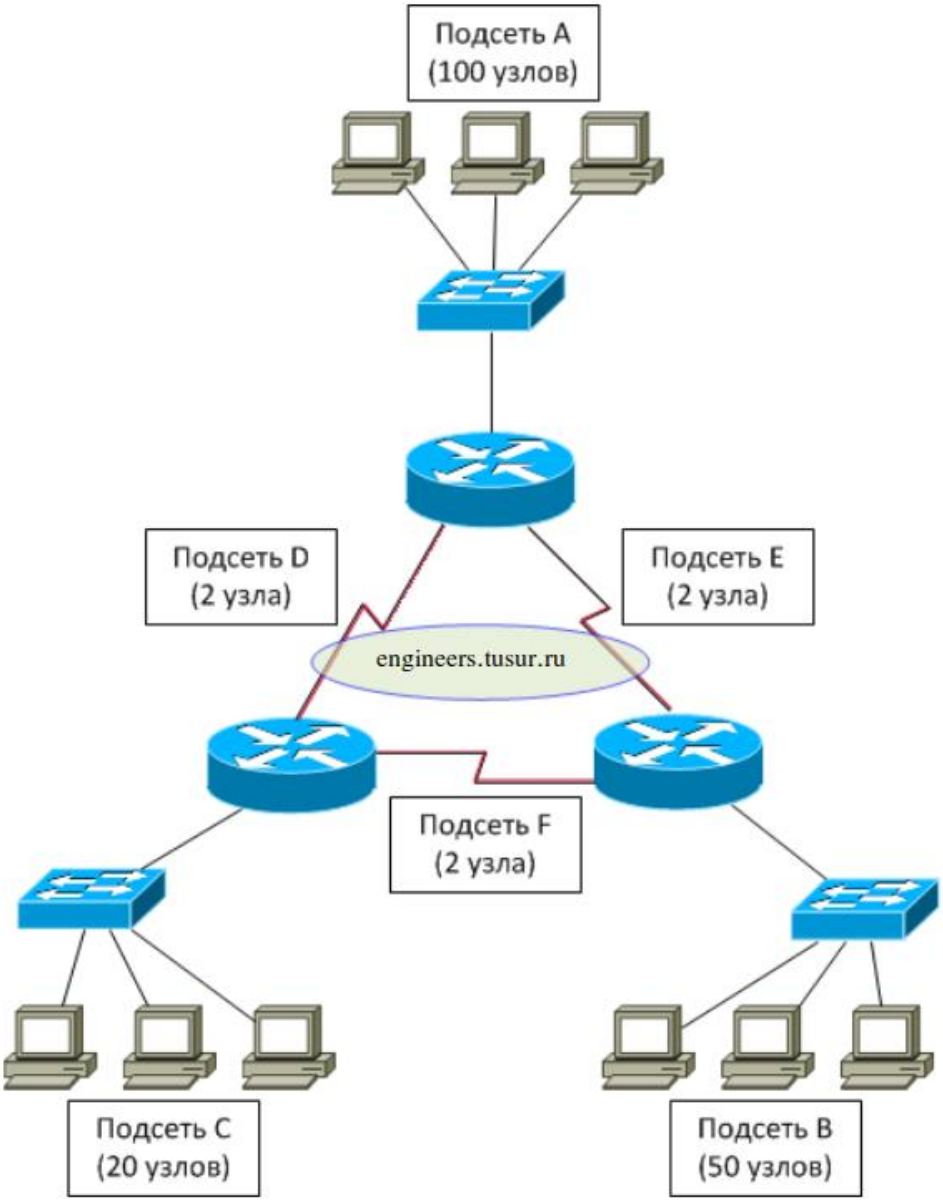


Рис. 5 Топология для расчета подсетей по индивидуальному заданию

Выделяем наибольшую по размеру подсеть A.

Начнём с выделения подсети с максимальным количеством узлов.

В двоичном виде IP-адрес 192.168.1.0/24 выглядит так (сетевая часть адреса красным цветом):

11000000.10101000.00000001.00000000 .

"/24" - это префиксная, краткая форма записи маски сети. Полная запись маски сети 255.255.255.0. В двоичном отображении маска сети выглядит так (хостовая часть выделена красным):

11111111.11111111.11111111.**00000000**

Нам доступно 8 бит для деления сети на подсети. При подборе требуемого количества бит в маске для заданного количества узлов в подсети мы возводим 2 в степень, равную числу разрядов. Для первой подсети A нам требуется выделить IP-адреса для 100 компьютеров. Если мы возьмем один разряд из восьми для организации подсетей, то сможем создать две подсети размером 2^7 = 128 — 2(служебные адреса) = 126 адресов.

Меняем маску с "/24" на "/25", в двоичном формате это будет:

11111111.11111111.11111111.10000000).

Применим новую маску к нашей сети (побитовая операция AND) и получим 2 подсети (порция сети выделена красным цветом):

Подсеть A - **11000000.10101000.00000001.0**0000000 (сеть 192.168.0.0/25)
Подсеть 2 - **11000000.10101000.00000001.1**0000000 (сеть 192.168.0.128/25)

Адресное пространство в подсети A занято с адреса 192.168.0.0 до адреса 192.168.0.127. При этом сам адрес 192.168.0.0 это адрес подсети, а адрес 192.168.0.127 это широковещательный адрес. Для назначения компьютерам нам доступны адреса, начиная с 192.168.0.1 до 192.168.0.126. Хотя нам требуется 100 адресов, мы заняли 126. Несмотря на то, что не нужные нам 26 адресов не будут заняты, использовать их в других подсетях уже не получится, адресное пространство в 126 адресов полностью занято подсетью A. Это можно считать запасом адресов на будущее в этой подсети.

Берем следующую по размеру подсеть B.

Для этой подсети нам нужно 50 адресов, у нас осталась одна подсеть 2: 192.168.1.128/25, в которой доступны 126 адресов. Если мы продлим маску еще на один разряд, мы сможем разбить полученную подсеть еще на две части и в хостовой части для каждой из них останется 6 бит. Это дает размер сети 2^6 = (64 — 2) = 62 адреса, что больше чем нам требуется. Занимаем еще один бит у порции адреса, маска становится /26, применяем её к нашей подсети 2 и получаем две новых подподсети:

Подсеть B - **11000000.10101001.00000000.10**000000 (сеть 192.168.0.128/26)
Подсеть 3 - **11000000.10101001.00000000.11**000000 (сеть 192.168.0.192/26)

На самом деле так не говорят, подсеть от подсети это тоже подсеть, а не подподсеть.

Приступим к следующей по размеру подсети C.

В ней нам требуется 20 адресов. Для этого достаточно 5 разрядов в хостовой части, поэтому продлим маску еще на один бит /27. Мы опять получим две подсети:
Подсеть C - **11000000.10101001.00000001.110**00000 (сеть 192.168.1.192/27)
Подсеть 4 - **11000000.10101001.00000001.111**00000 (сеть 192.168.1.224/27)

Остались наименьшие по размерам подсети D, E и F.

Для них требуется всего по два адреса. С учетом еще двух служебных адресов, которые есть в каждой сети, для таких сетей требуется два разряда в хостовой части, т. е. мы можем использовать маску /30. Если мы продлим маску, полученную в подсети C на три разряда до /30, то мы сможем организовать восемь таких малых подсетей, но нам требуется только три. Возьмем три смежные подсети, оставшиеся подсети 5 и 6 оставим как резерв:

Разделим Подсеть 4 на более мелкие подсети:
Подсеть 4 - **11000000.10101000.00000001.111**00000 (сеть 192.168.1.224/27)
Подсеть D - **11000000.10101000.00000001.111000**00 (сеть 192.168.1.224/30)
Подсеть E - **11000000.10101000.00000001.111001**00 (сеть 192.168.1.228/30)
Подсеть F - **11000000.10101000.00000001.111010**00 (сеть 192.168.1.232/30)
Подсеть 5 - 11000000.10101000.00000001.11101100 (сеть 192.168.1.236/30)
Подсеть 6 - 11000000.10101000.00000001.11110000 (сеть 192.168.1.240/28)

Задача разделения блока адресов на подсети выполнена.

Полученные подсети:

Подсеть A - 192.168.1.0/25
Подсеть B - 192.168.1.128/26
Подсеть C - 192.168.1.192/27
Подсеть D - 192.168.1.224/30
Подсеть E - 192.168.1.228/30
Подсеть F - 192.168.1.232/30

ВЫПОЛНЕНИЕ РАБОТЫ

Часть 1 Разбиение сети на подсети

В соответствии с индивидуальным вариантом табл. 1 выполните разбиение исходной сети на подсети А,Б и В. Предусмотрите также три дополнительных подсети для соединения маршрутизаторов, как в топологии рис. 5 и показанном примере расчета. Номер варианта — ваш номер в списке группы.

Таблица 1 – Варианты заданий

Вариант	Исходная сеть (блок адресов)	Количество компьютеров в отделах		
		А	Б	В
1	118.7.50.0 /25	7	9	27
2	39.221.98.0 /25	8	5	18
3	88.27.252.0 /23	30	9	46
4	81.104.216.0 /21	48	120	249
5	7.50.128.0 /19	267	176	678
6	89.151.32.0 /19	311	246	806
7	126.61.74.0 /23	8	61	17
8	36.121.96.0 /19	311	696	226
9	28.54.64.0 /19	957	153	274
10	67.253.1.0 /20	365	116	508
11	77.75.0.0 /18	338	830	1403
12	5.63.168.0 /21	119	61	226
13	85.123.72.0 /21	189	51	72
14	72.241.3.0 /25	12	7	3
15	87.228.68.0 /22	26	45	71
16	46.41.64.0 /18	384	1535	675
17	57.214.86.0 /23	63	9	21
18	74.30.128.0 /19	346	179	732
19	88.61.128.0 /20	366	77	130
20	10.58.180.0 /22	30	92	43
21	112.56.76.0 /22	23	114	60
22	2.78.160.0 /19	214	443	525
23	30.182.64.0 /18	624	1700	358
24	75.39.128.0 /19	625	219	372
25	98.115.89.37 /21	48	119	250
26	35.163.168.0 /21	119	60	224

Часть 2 Настройка маршрутизации для рассчитанных подсетей

В программе VirtualBox создайте три виртуальных машины:

Виртуальная машина Linux. Можно использовать компактный образ Damn Small Linux - Damn Small Linux, Download the ISO, правда этот Linux имеет довольно старое ядро 2.4, или

любой другой, более современный Linux, например этот https://lubuntu.net/ Виртуальная машина маршрутизатора Mikrotik с RouterOS. Загрузить ее можно с официального сайта https://download.mikrotik.com/routeros/6.48.6/chr-6.48.6.ova [4].

Еще одна машина с маршрутизатором Mikrotik. Ее можно создать методом связанного клонирования с генерацией новых MAC-адресов, Рис. 6.

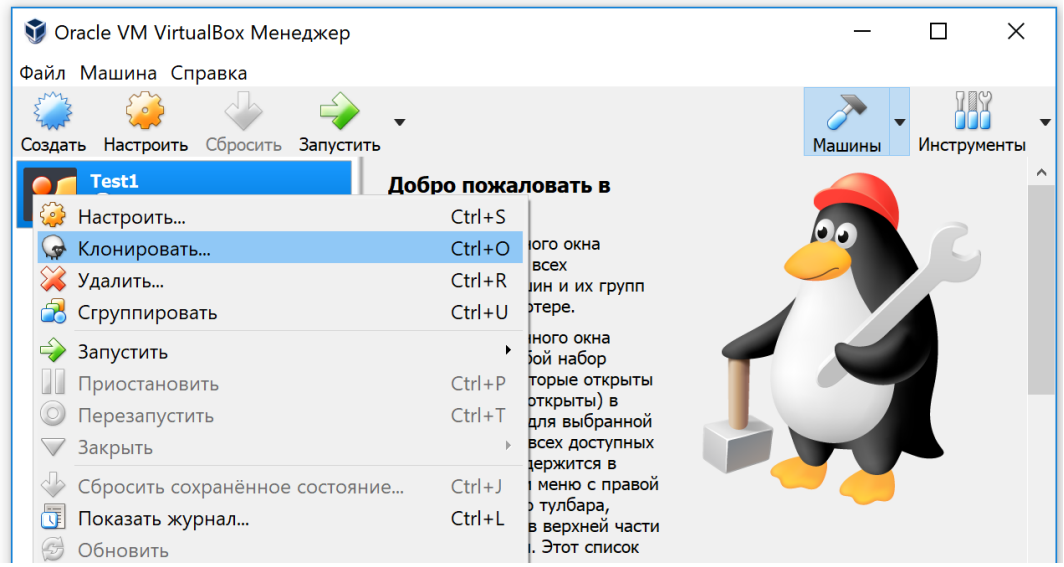


Рис. 6. Клонирование существующей виртуальной машины

Для маршрутизатора Mikrotik логин по умолчанию **admin** пароль пустой.

Образ DSL это Live CD, операционную систему можно запустить прямо с диска, без установки на жесткий диск виртуальной машины. После того, как скачали iso файл с загрузочным образом компакт диска, создайте новую виртуальную машину с пустым диском. Укажите в качестве названия DSL, тогда автоматически будет выбрано правильное ядро Linux, DSL использует достаточно старое ядро 2.4, рис. 7.

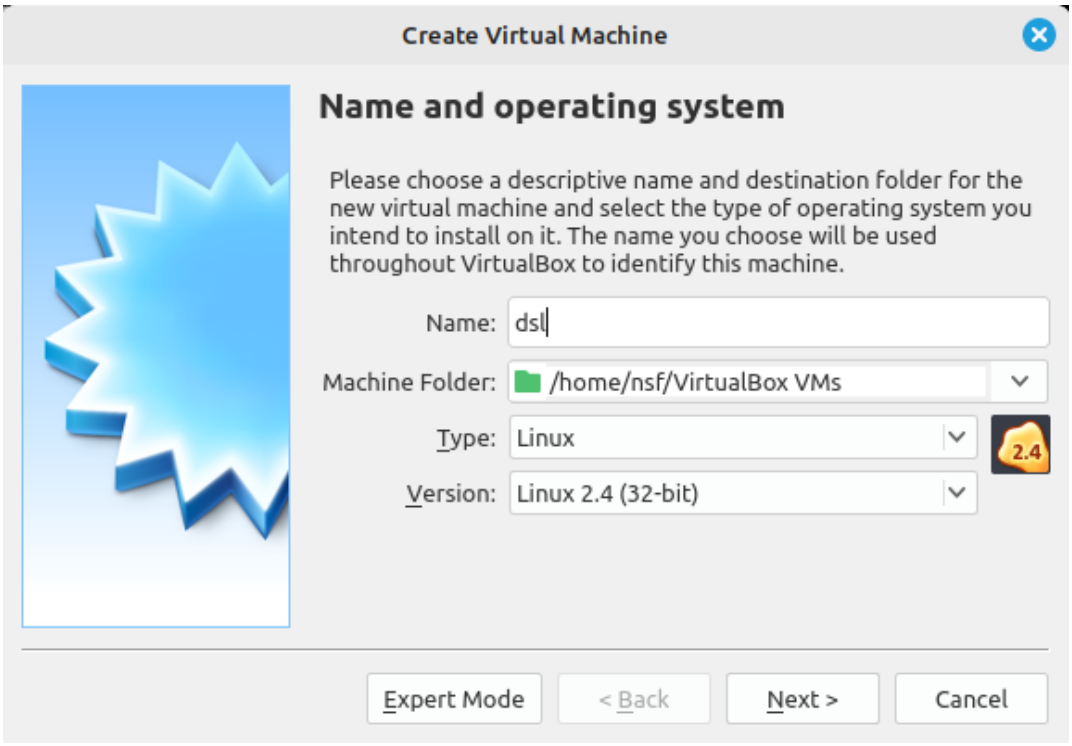


Рис. 7 Создание виртуальной машины для DSL

Несмотря на то, что эта версия Linux имеет графический интерфейс, она очень нетребовательна к аппаратным ресурсам. Объем оперативной памяти, предлагаемый Virtualbox по умолчанию вполне достаточен, рис. 8.

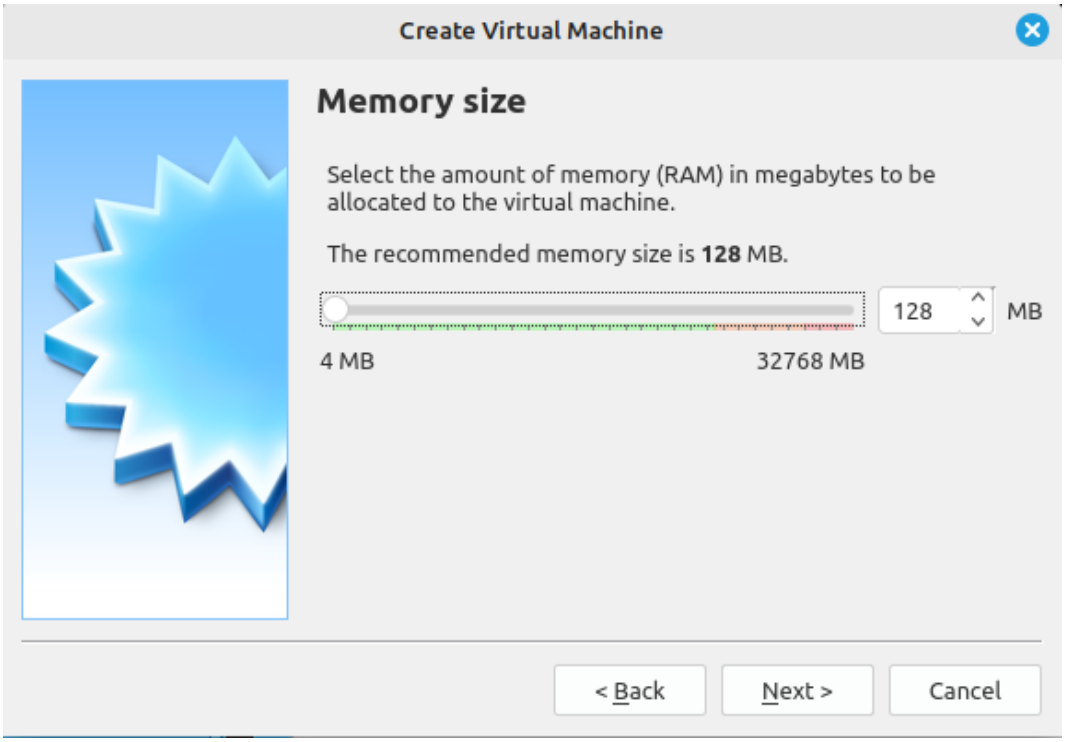


Рис. 8 Объем оперативной памяти для DSL

Необходимо создать новый жесткий диск, но каких-либо действий с ним не потребуется, рис. 9.

Рис. 9 Создание жесткого диска для виртуальной машины DSL

Формат VDI, предлагаемый по умолчанию, «родной» для Virtualbox, так что он подойдет, рис. 10.

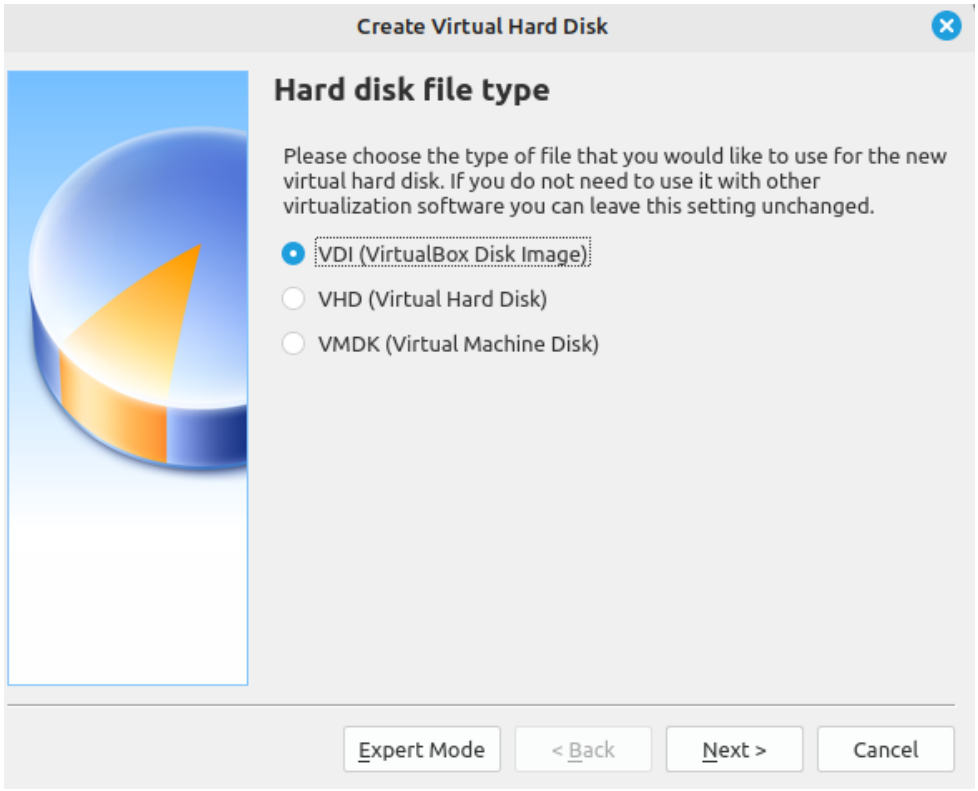


Рис. 10 Выбор формата жесткого диска

Динамически увеличивающийся диск это тип жесткого диска, который может существовать только в системах виртуализации. Вы задаете для него максимальный размер, реальный же размер будет зависеть от занятого файлами места на этом диске и может быть сильно меньше максимального. По мере установки программ и размещения файлов на этом диске, его размер будет постепенно расти, приближаясь к заданному максимальному размеру, рис. 11.

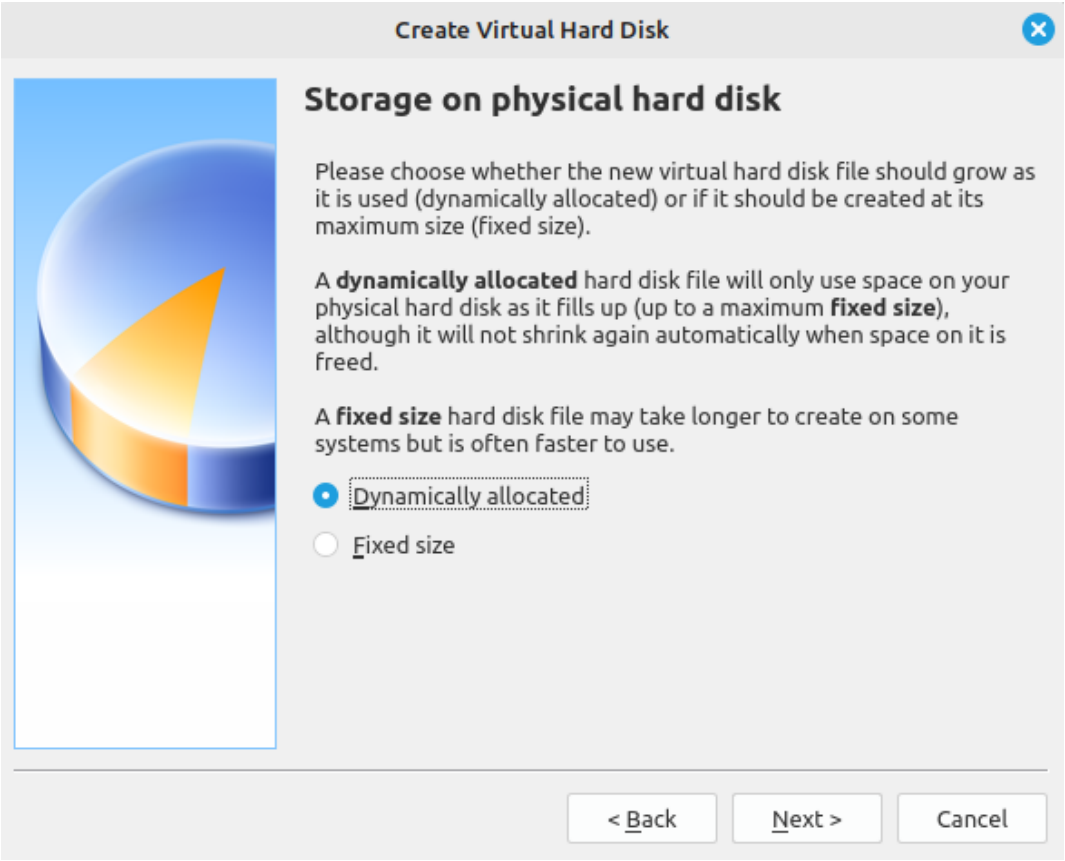


Рис. 11 Выбор типа жесткого диска — динамически расширяющийся

После создания виртуальной машины, добавьте в оптический дисковод файл iso образа, рис. 12.

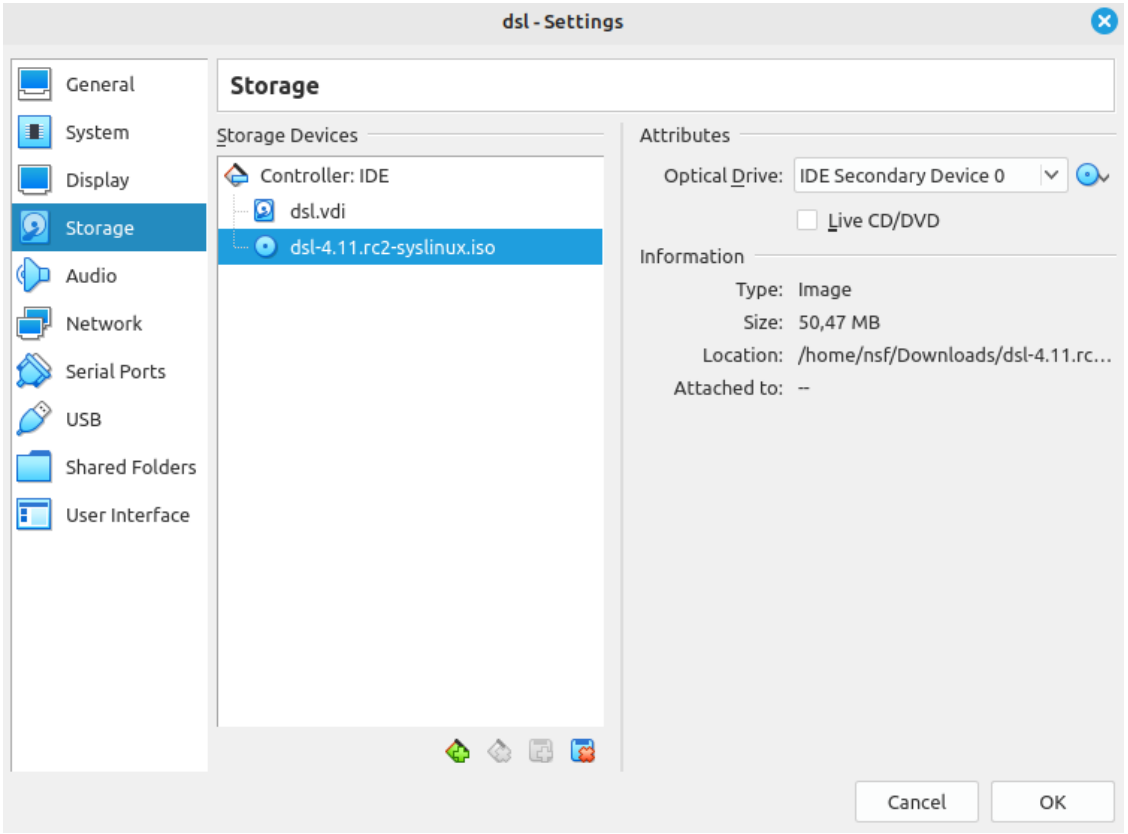


Рис. 12 Добавление образа диска DSL в оптический дисковод виртуальной машины.

При создании виртуальных машин убедитесь, что в каждой из них активны два сетевых адаптера. Включите эти адаптеры во внутреннюю сеть (Internal Network), рис. 13. Это специальный тип сетевого соединения, при котором разные виртуальные машины через сетевое соединение «видят» друг друга, если имя внутренней сети на них одинаковое.

Разумеется, настройки IP-адресов на этих машинах для соответствующего имени внутренней сети должна быть из одного адресного пространства. Задайте рассчитанное вами адресное пространство для трех подсетей, соединяющих маршрутизаторы.

Для рассчитанных вами подсетей А, Б и В, на каждом устройстве создайте дополнительный виртуальный интерфейс и настройте на нем адрес для соответствующей подсети. Например, на Linux для подсети А, на Микротик 1 для подсети Б, на Микротик 2 для подсети В.

Создать дополнительный интерфейс на Linux можно командой:

modprobe dummy

или **sudo modprobe dummy** если вы выполняете команду не в терминале администратора.

Выполнение команды включит модуль и автоматически создаст интерфейс dummy0. На него можно задать требуемый адрес.

В Микротик создать интерфейс можно командой:

/interface bridge add name=lobridge

Задать адрес на интерфейсе:

ip address add address=<ваш адрес>/ваш префикс interface=lobridge

Виртуальные интерфейсы будут изображать рассчитанные вами сети с десятками пользователей.

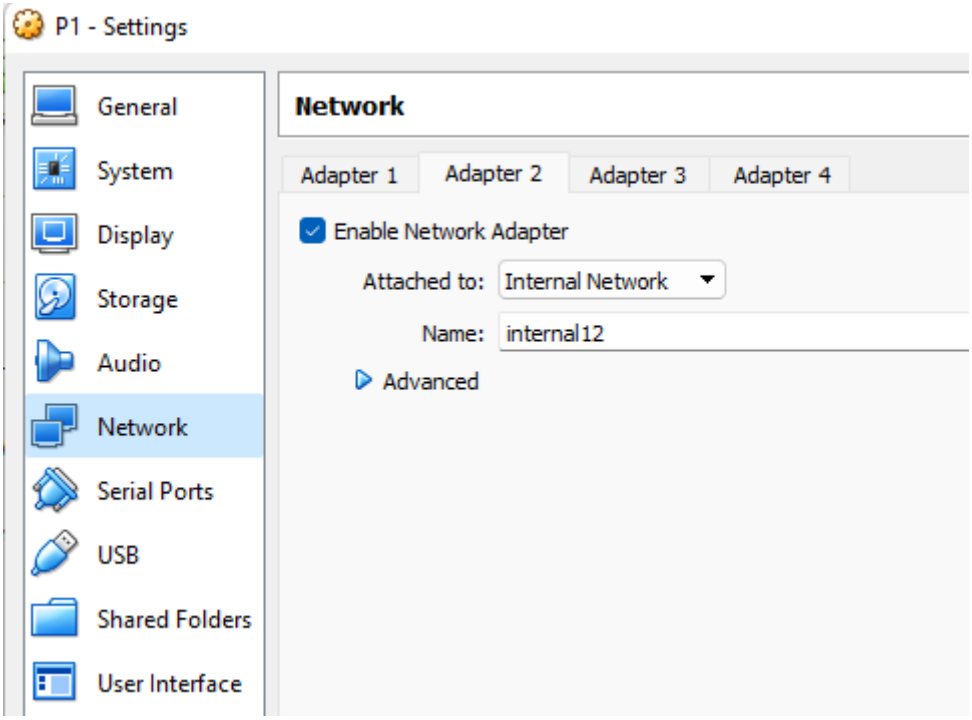


Рис. 13 Активация второго сетевого адаптера, настройка внутренней сети

В результате у вас должно появиться соединение всех трех виртуальных машин в виде треугольника, рис. 14. Здесь три внутренних сети VirtualBox с РАЗНЫМИ названиями и три соответствующих IP-подсети с разными адресными пространствами для соединения устройств и по дополнительной сети на виртуальном интерфейсе каждого устройства, так что топология будет похожа на рис. 5.

Тип соединения «внутренняя сеть» (Internal Network) в системе виртуализации позволяет как бы проложить физический кабель между устройствами, у которых сетевые адаптеры включены во внутреннюю сеть с одинаковым названием. Для соединения Linux ↔ Mikrotik_1 берем внутреннюю сеть с одним названием: «внутренняя сеть 2», для соединения Linux ↔ Mikrotik_2 с другим названием: «внутренняя сеть 3», для соединения Mikrotik_1 ↔ Mikrotik_2 с третьим названием: «внутренняя сеть 1». Именно разные названия внутренней сети позволяют нам создать три отдельных соединения, как показано на рис. 14.

Внутренняя сеть по умолчанию имеет название «intnet» не оставляйте это название без изменений! Если все интерфейсы будут включены в одну и ту же внутреннюю сеть, мы получим совершенно другую топологию!

Каждая отдельная внутренняя сеть это также отдельное адресное пространство, отдельная подсеть. Используйте для адресации устройств ваши личные подсети из практического задания. В показанной на рис. 14 топологии, каждый маршрутизатор включен в две сети своими «как бы» физическими интерфейсами и еще в одну виртуальным интерфейсом. Таким образом, каждое устройство «видит» три непосредственно подключенные сети, но не видит еще три.

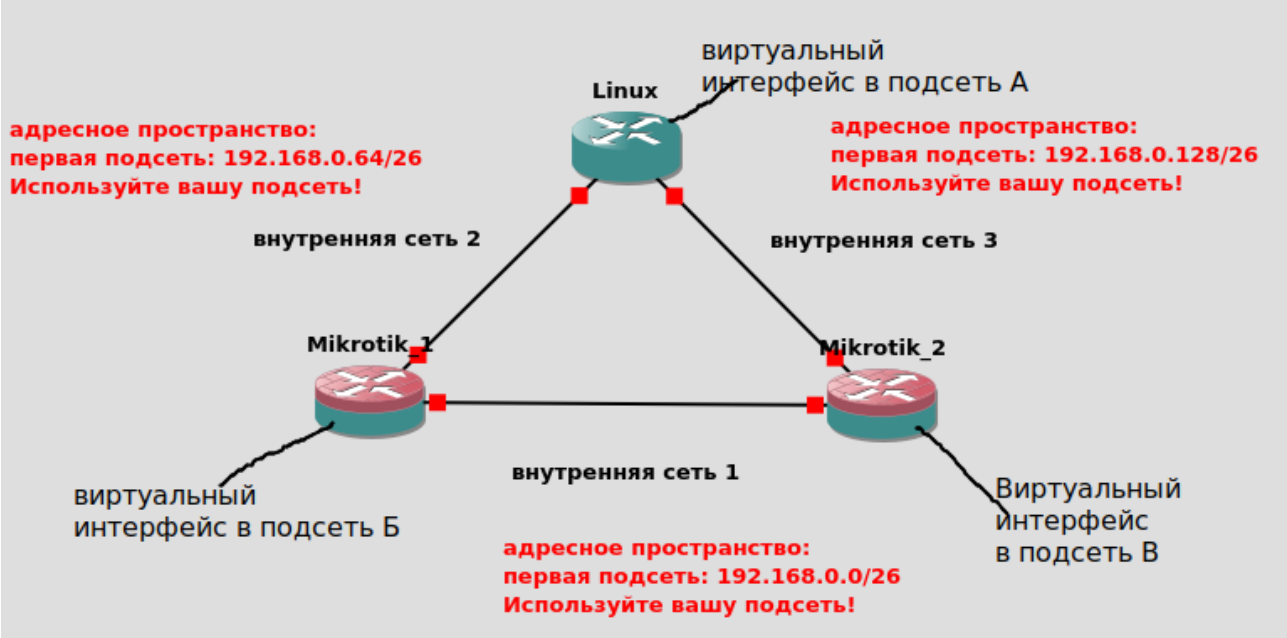


Рис. 14 Топология создаваемой сети

Мы не подключаем коммутатор и компьютеры пользователей, как на рис. 5, достаточно адреса на виртуальном интерфейсе из адресного пространства рассчитанной вами сети А, Б или В, чтобы в таблице маршрутизации появилась еще одна запись для этой сети. Ваша задача – настроить маршрутизацию таким образом, чтобы все шесть сетей были видны и доступны на всех устройствах.

В виртуальной машине Virtualbox без установленных дополнений на операционную систему не будет работать система бесшовной интеграции. Это значит, что «фокус внимания» операционной системы будет или на основной машине, или перейдет в виртуальную машину. Переключить фокус можно с помощью Host-клавиши, по умолчанию это правый Ctrl. Если вы работаете внутри виртуальной машины и хотите вернуться на основной компьютер, нажмите правый Ctrl. При щелчке мышью на окне виртуальной машины появится окно с предложением захватить фокус и если разрешить захват, то фокус перейдет внутрь виртуальной машины.

В графической среде DSL открыть окно командной строки можно нажав правой клавишей мышки в любом месте рабочего стола. Появится контекстное меню, в котором нужно выбрать строку Xshells и далее любую из строк. Transparent это полупрозрачное окно командной строки, Light с белым фоном и черным текстом, Dark наоборот, рис. 15. Если выбирате Root Access, там тоже те же три типа, только от имени суперпользователя. По умолчанию вы входите в систему как пользователь dsl, которому для выполнения команд настройки интерфейсов и модификации таблицы маршрутизации потребуется перед ними вводить дополнительную команду sudo.

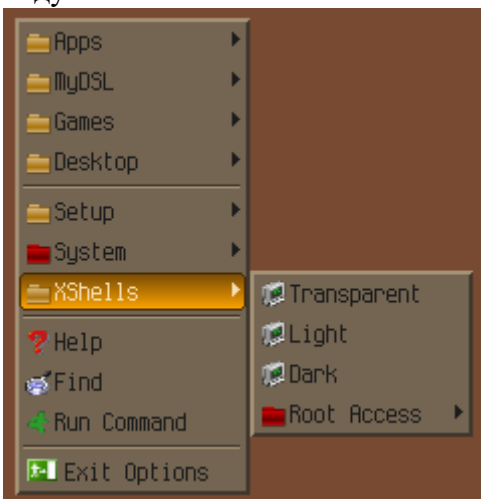


Рис. 15 Контекстное меню DSL, запуск командной строки

Для модификации или просмотра настроек сетевых интерфейсов в операционных системах Linux можно использовать команду:

ifconfig

Введенная без параметров, команда покажет имена и настройки всех активных, т. е. работающих, сетевых интерфейсов. Если хотите увидеть все сетевые интерфейсы, не только работающие, но и отключенные, используйте эту команду с ключом -a

ifconfig -a

Для настройки адреса на интерфейсе нужно указать его имя, а затем желаемый адрес:

ifconfig eth0 192.168.1.1

В таком виде команда присвоит указанный адрес и автоматически назначит маску по классовой схеме. Если интерфейс был выключен, команда назначения адреса включит его. Если требуется задать маску, отличную от классовой, используйте ключевое слово netmask:

ifconfig eth0 192.168.1.1 netmask 255.255.224.0

Если принудительно хотите включить интерфейс, используйте ключ up, для выключения down.

ifconfig eth0 192.168.1.1 netmask 255.255.224.0 up

Хотя, как уже было отмечено, интерфейсы переводятся в нерабочее состояние обычно потому что не имеют настроенного IP-адреса, и для включения интерфейса достаточно задать этот адрес.

Команда ifconfig часть программного пакета net-tools, в который входят также команды: arp, netstat, route. В современных версиях Linux этот пакет часто отсутствует, т. к. считается устаревшим. Вместо этого пакета используется iproute2, в составе которого большое разнообразие команд, позволяющих настроить и проверить сетевые интерфейсы. Н

апример, команда:

ip address, которая вводится часто в сокращенном виде

ip a

покажет ту же информацию по сетевым интерфейсам что и команда ifconfig -a немного в другом формате. Команда:

ip addr add 172.16.1.1/27 dev eth1

позволит назначить адрес на интерфейс. Просмотреть таблицу маршрутизации можно командой
route или

netstat -r

из пакета net-tools или командой:
ip route

из пакета iproute2. В DSL установлены оба пакета. Команда добавления маршрута показана на рис. 16. Указывать выходной интерфейс не обязательно, вместо него можно указать адрес следующего устройства по пути в сеть назначения:
route add -net 192.168.5.0/27 gw 192.168.5.5

С помощью iproute2 то же самое можно сделать так
ip route add 192.168.5.0/27 via 192.168.5.5

```
root@ubuntu:~# netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         192.168.142.2   0.0.0.0         UG        0 0          0 eno16777736
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 eno16777736
192.168.142.0   0.0.0.0         255.255.255.0   U         0 0          0 eno16777736
root@ubuntu:~#
root@ubuntu:~#
root@ubuntu:~# route add -net 192.168.5.0/24 dev eno16777736
root@ubuntu:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         192.168.142.2   0.0.0.0         UG    0      0        0 eno16777736
link-local      *               255.255.0.0     U     1000   0        0 eno16777736
192.168.5.0     *               255.255.255.0   U     0      0        0 eno16777736
192.168.142.0   *               255.255.255.0   U     0      0        0 eno16777736
root@ubuntu:~#
root@ubuntu:~# route del -net 192.168.5.0/24
root@ubuntu:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         192.168.142.2   0.0.0.0         UG    0      0        0 eno16777736
link-local      *               255.255.0.0     U     1000   0        0 eno16777736
192.168.142.0   *               255.255.255.0   U     0      0        0 eno16777736
```

Рис. 16. Добавление, удаление и просмотр сетевых маршрутов в Linux

На маршрутизаторе Mikrotik посмотреть число и наименование интерфейсов можно командой:

/interface print

Назначить адрес на интерфейс можно командой:

/ip address add address=192.168.1.1/24 interface=<имя интерфейса>

Проверить какие адреса на какие интерфейсы назначены:

/ip address print

Добавление маршрута:

/ip route add dst-address=192.168.2.0/24 gateway=172.16.1.2

Вывод таблицы маршрутизации:

/ip route print

Mikrotik маршрутизирует пакеты (перебрасывает их с интерфейса на интерфейс) по умолчанию без каких-либо дополнительных настроек.

В операционных системах для настольных компьютеров Linux и Windows, эта функция чаще всего отключена.

Для проверки включена ли маршрутизация на Linux выполните из командной строки:

sysctl net.ipv4.ip_forward

или *cat /proc/sys/net/ipv4/ip_forward*

В ответ мы получим текущий статус (1 – маршрутизация включена, 0 — выключена). Если маршрутизация выключена, включите ее командой:

sysctl -w net.ipv4.ip_forward=1

или echo 1 > /proc/sys/net/ipv4/ip_forward

После включения маршрутизации выполните проверку еще раз и убедитесь, что состояние изменилось.

Проверку маршрутизации выполните отправляя эхо-запросы командой ping в по адресам в тех подсетях, которые не подключены непосредственно к устройству.

ПРАКТИКА №5
АНАЛИЗ МЕХАНИЗМОВ УПРАВЛЕНИЯ ПОТОКОМ И ПЕРЕГРУЗКОЙ В ПРОТОКОЛЕ TCP

Цель работы: изучить механизмы управления потоком и предотвращения заторов в протоколе TCP, сравнить работу разных версий TCP.

ОБЩАЯ ИНФОРМАЦИЯ

Окно передачи данных в TCP

TCP (протокол управления передачей) — это протокол, ориентированный на соединение, что означает, что мы отслеживаем объем переданных данных. Отправитель передает некоторые данные, и получатель должен подтвердить это. Если мы не получим подтверждение вовремя, отправитель повторно передаст данные. TCP использует «окно управления», позволяющее передавать сразу несколько сегментов данных, которые получатель должен подтвердить. Когда мы запускаем TCP-соединение, хосты будут использовать буфер приема, в котором мы временно сохраняем данные, прежде чем приложение сможет их обработать. Когда получатель отправляет подтверждение, он сообщает отправителю, какой объем данных он может принять, прежде чем получатель отправит подтверждение. Мы называем это размером окна. По сути, размер окна указывает размер буфера приема. Обычно TCP-соединение начинается с небольшого размера окна, и каждый раз при успешном подтверждении размер окна увеличивается. Вот пример, рис. 1:

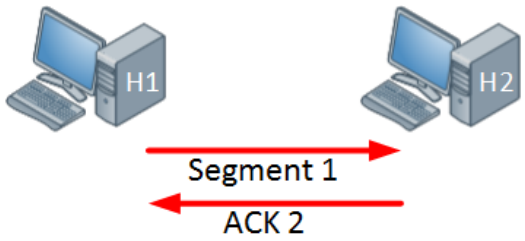


Рис. 1 Передача одного сегмента

Выше у нас есть два хоста: хост с левой стороны отправит один сегмент, а хост с правой стороны отправит в ответ подтверждение. Поскольку подтверждение получено, размер окна увеличится:

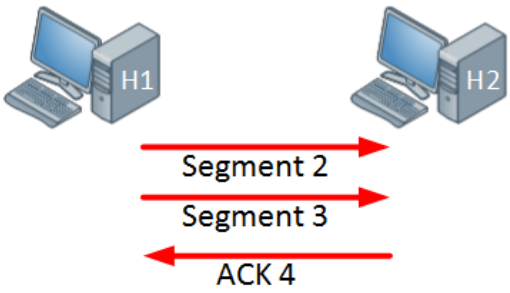
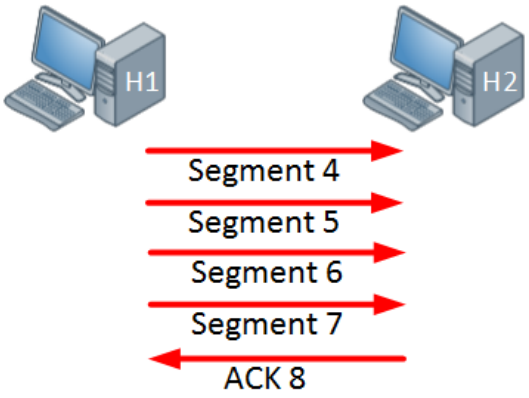


Рис. 2 Передача двух сегментов

Хост на левой стороне теперь отправляет два сегмента, а хост на правой стороне вернет одно подтверждение, подтверждающее их оба. Все работает нормально, поэтому

размер окна увеличится еще больше:

Рис. 3 Передача четырех сегментов



Хост теперь отправляет четыре сегмента, а хост на правой стороне отвечает одним подтверждением. В приведенном примере размер окна продолжает увеличиваться до тех пор, пока получатель отправляет подтверждения для всех наших сегментов или когда размер окна достигает определенного максимального предела. Если получатель не отправляет подтверждение в течение определенного периода времени (так называемого времени двойного оборота, Round Trip Time), размер окна будет уменьшен.

Когда интерфейс перегружен, IP-пакеты могут быть отброшены. Чтобы справиться с этой проблемой, в TCP имеется ряд алгоритмов управления перегрузкой. Один из них называется медленный старт. При медленном запуске TCP размер окна первоначально будет расти экспоненциально (размер окна удваивается), но как только пакет будет отброшен, размер окна уменьшится до одного сегмента. Затем он снова будет расти в геометрической прогрессии, пока размер окна не станет половиной того, который был на момент возникновения перегрузки. В этот момент размер окна будет расти линейно, а не экспоненциально, рис. 4.

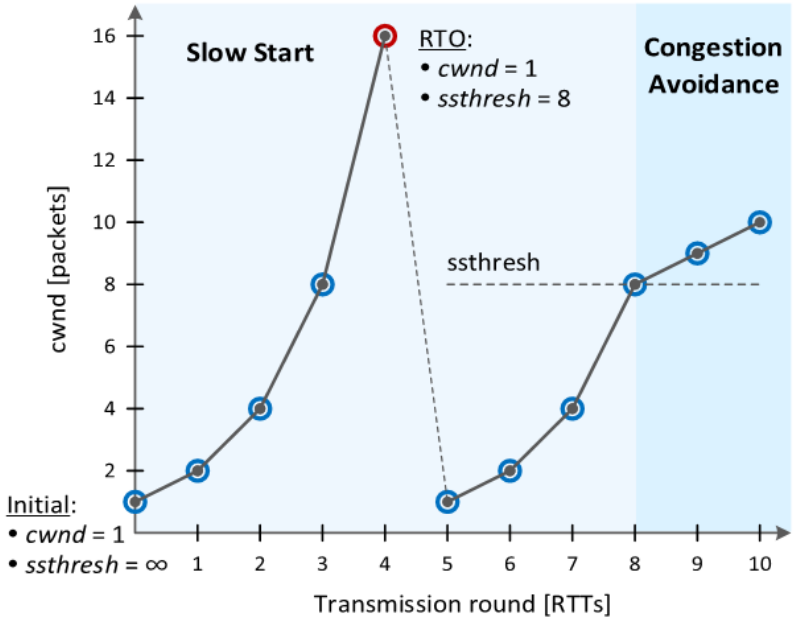


Рис. 4 Алгоритм регулирования размера окна и предотвращения перегрузки

Когда интерфейс перегружен, возможно, что все ваши TCP-соединения будут

испытывать медленный запуск TCP. Пакеты будут отбрасываться, и тогда все TCP-соединения будут иметь небольшой размер окна. Это называется глобальной синхронизацией TCP. Вот как это выглядит:

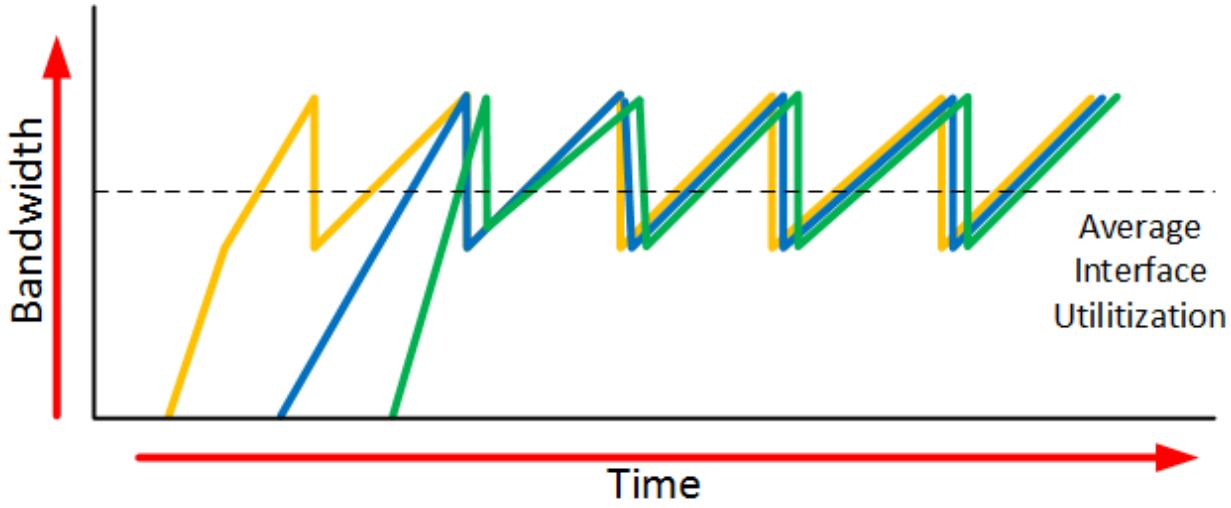


Рис. 5 Синхронизация протокола TCP из-за потерь сегментов на интерфейсе

Оранжевая, синяя и зеленая линии — это три разных TCP-соединения. Эти TCP-соединения запускаются в разное время, и через некоторое время интерфейс перегружается, и пакеты всех TCP-соединений отбрасываются. Что происходит, так это то, что размер окна всех этих TCP-соединений упадет до одного, и как только перегрузка интерфейса исчезнет, размеры всех их окон снова увеличатся. Затем интерфейс снова перегружается, размер окна снова падает до единицы, и история повторяется. В результате мы не используем всю доступную полосу пропускания, которую может предложить наш интерфейс. Если вы посмотрите на пунктирную линию, то увидите, что среднее использование интерфейса не очень высокое.

Программа Mininet

Работа выполняется в программе Mininet. Mininet — это эмулятор сети, который создает сеть из виртуальных компьютеров, коммутаторов, управляющих сетью контроллеров, при этом все они имеют реальные сетевые интерфейсы на основе стандартного сетевого программного обеспечения Linux.

Эмулятор Mininet широко используется в научных исследованиях, разработке, а также при обучении сетевым технологиям, создании прототипов, тестировании, отладке и других задачах, при решении которых необходимо наличие полной экспериментальной сети на ноутбуке или другом ПК. Мининет имеет интерфейс командной строки с поддержкой команд создания заданных топологий и протокола OpenFlow для отладки или запуска общесетевых тестов. Поддерживает произвольные пользовательские топологии и включает в себя базовый набор параметризованных топологий. Программа Mininet работает на платформе Linux. Для запуска ее на Windows мы будем использовать подготовленные образы виртуальной машины в формате OVF (Open Virtualization Format) — открытый стандарт для хранения и распространения виртуальных машин. Стандарт описывает открытый, переносимый, расширяемый формат для распространения образов виртуальных машин. Стандарт OVF не

привязан к какой-либо реализации гипервизора или аппаратной архитектуре. Все доступные релизы доступны в репозитории https://github.com/mininet/mininet/releases/

Скачайте последний актуальный релиз на основе Ubuntu 20.04 https://github.com/mininet/mininet/releases/download/2.3.0/mininet-2.3.0-210211-ubuntu-20.04.1-legacy-server-amd64-ovf.zip [1].

Логин и пароль для входа **mininet/mininet**

Программа Wireshark уже установлена в виртуальной машине, но для уменьшения размера образа не установлено графическое окружение рабочего стола, которое в Linux работает как отдельный сервис.

Для запуска Wireshark мы будем использовать внешний сервер графического режима. Такой сервер есть в составе популярного SSH-клиента **MobaXterm**

Загрузите его по ссылке:
https://download.mobatek.net/2212022060563542/MobaXterm_Portable_v22.1.zip [2].

Если мы хотим запускать программы на внешнем сервере графического окружения, то мы должны подключиться по протоколу SSH к виртуальной машине Mininet с указанием ключа перенаправления запросов выполнения графических программ на внешний сервер графического режима: -X или -Y, таким образом:

ssh -Y mininet@IPaddress_mininetVM

Здесь IP-адрес виртуальной машины это адрес сетевого адаптера, включенного в режиме «Сетевой мост».
(Если вы работаете с машиной в режиме NAT, по умолчанию, тогда можно настроить проброс портов в расширенных настройках сети и указывать адрес интерфейса сетевого адаптера VirtualBox HostOnly, обычно это 192.168.56.1).

Эмулятор запускается командой **mn** от имени администратора:

sudo mn

При подключении с пробросом X Window сессии через SSH соединения по сети на машине Mininet открывается сокет на порту 6000. Чтобы защитить это соединение в профиле пользователя, который подключился по SSH (в нашем случае это пользователь mininet) в специальном файле сохраняются MAGIC-COOKIE — уникальный цифровой отпечаток, разрешающий выполнение графических программ этому пользователю. Другим пользователям, у которых не будет этого цифрового кода, не будет и разрешения.
Просмотреть сохраненный код можно командой:

xauth list $DISPLAY

или просто **xauth list** так как других переменных не будет.

Так как подключение по SSH выполняется пользователем mininet, а запуск программы-эмулятора от имени администратора, то возникнет проблема с запуском

графических программ, таких как Wireshark в окне Mininet. Чтобы администратор мог запустить графические программы нужно скопировать в его профиль MAGIC-COOKIE из профиля пользователя mininet. Сделать это можно командой:

xauth add <вставить MAGIC-COOKIE>, скопированные из профиля пользователя mininet.

Таким образом, чтобы графические программы из оболочки эмулятора, который требует запуска от имени администратора, могли нормально запускаться через SSH соединение с пробросом X-сессии, открытое пользователем mininet, нужно:

1. После создания такого соединения посмотреть MAGIC-COOKIE из профиля пользователя mininet командой **xauth list**
2. Переключиться в профиль администратора командой: **sudo -i** или **sudo su -**
3. Скопировать MAGIC-COOKIE в профиль администратора командой: **xauth add** <вставить MAGIC-COOKIE>

При возникновении ошибки отсутствия файла для хранения MAGIC-COOKIE в профиле root-пользователя, создать этот файл с именем, которое выводится в сообщении ошибки, командой **touch**.

4. После этого можно запустить эмулятор sudo mn и выполнять в нем вызов графических программ, например, xterm.

ВЫПОЛНЕНИЕ РАБОТЫ

1. Запустите виртуальную машину Mininet. Проверьте IP-адрес на сетевом интерфейсе виртуальной машины и его доступность с основной машины. Для выведения информации об IP-адресах интерфейсов используйте команду:

ip address или

ifconfig

2. Запустите эмулятор Mininet командой

sudo mn

Эмулятор по умолчанию создает простую топологию из двух узлов с именами h1 и h2, подключенных к коммутатору. Командой net можно посмотреть структуру созданной сети, рис. 6.

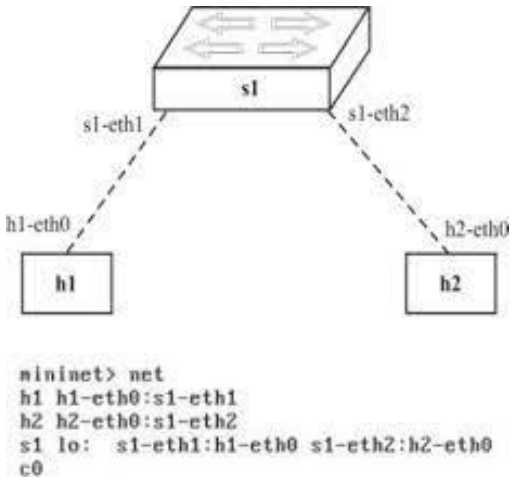


Рис. 6 Топология сети, создаваемой Mininet по умолчанию

3. Откройте окно командной строки на хостах эмулируемой сети командой:
xterm h1 h2

4. Запустите программу Wireshark для захвата трафика между h1 и h2 из командной строки mininet командой:
sh wireshark

5. Для сравнения разных алгоритмов регулирования скорости передачи и контроля перегрузки в TCP выполним два принципиально разных сценария: 1) отправка небольшого файла и 2) отправка значительного объема данных. При этом будем управлять алгоритмом работы TCP. Текущий алгоритм можно посмотреть командой:

sysctl net.ipv4.tcp_congestion_control

или

cat /proc/sys/net/ipv4/tcp_congestion_control

В операционной системе это будет скорее всего алгоритм **cubic.**

Отметьте, какой алгоритм вы будете использовать.

6. Выполним первый сценарий отправки небольшого файла. Для этого запустим на h1 мини-веб-сервер на Python командой:

python -m http.server 80 &

На h2 обратимся к веб-серверу и скачаем индексную страницу:

wget -o - 10.0.0.1

Сохраните захваченный Wireshark файл на всякий случай для последующего анализа с именем **web.pcapng**

Изучите, как изменялся размер окна со стороны клиента и со стороны сервера во время передачи файла. Размер окна указывается в поле Window заголовка TCP. Размер окна измеряется в байтах, а данные передаются сегментами, каждый из которых содержит

сколько-то байт. В современных сетях значение поля Window домножается на коэффициент, который передается в поле «Опции» и называется Windows Scale — масштаб окна. В этом поле передается степень двойки. Значение поля Window нужно умножить на 2 в этой степени, чтобы получить действующий размер окна.

Постройте графики для размера окна в последовательности сегментов, передаваемых от сервера к клиенту и в обратном направлении.

Откройте меню «Статистика» Wireshark, найдите внизу пункт «графики потока TCP» и в нем пункты Throughput — пропускная способность и Time/Sequence (Stevens). Изучите эти графики. Направление передачи в них можно переключить кнопкой в правом нижнем углу, рис. 7:

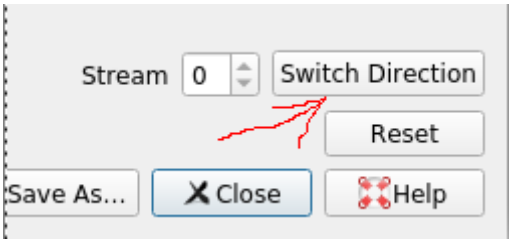


Рис. 7 Переключение направления передачи данных в окне статистики

В каком направлении передается основной объем данных?
Сколько потребовалось сегментов чтобы передать веб-страничку?
В скольки сегментах содержался основной объем данных?

7. Выполним второй сценарий отправки большого объема данных. Сначала остановим веб-сервер на h1. При запуске сервера с ключом & его процесс переходит в фоновый режим, освобождая нам командную строку и показывает идентификатор процесса. Если вы не запомнили идентификатор, можно вывести список всех процессов и отфильтровать его по ключевому слову, например так:

ps aux | grep http

Для остановки процесса http.server нужно подать команду с указанием номера процесса:

kill -9 1813

Здесь номер процесса показан только для примера, вы должны определить какой у вас номер процесса и использовать свой номер в этой команде.
После остановки веб-сервера запустим на h1 сервер iperf командой:

iperf -s

По умолчанию сервер iperf использует протокол TCP и прослушивает порт 5001.
На хосте h2 запустим клиент iperf командой:

iperf -c 10.0.0.1

Клиент по умолчанию передает максимально возможный объем данных на сервер, чтобы заполнить полосу пропускания и определить реально достижимую максимальную скорость соединения. Время в течении которого выполняется передача — 10 секунд.

Выведите те же графики Throughput — пропускная способность и Time/Sequence (Stevens) из меню «Статистика» Wireshark, для этого сценария. Изучите эти графики. Какие можно сделать выводы из полученной информации?

Используйте настраиваемый график из меню «Статистика»: IO Graphs. В этом графике вы можете задать фильтр, выводящий интересующую вас информацию. Создайте фильтр tcp.window_size_value Как меняется значение окна согласно этого графика?

8. Измените алгоритм управления потоком данных и перегрузками TCP командой

sudo sysctl net.ipv4.tcp_congestion_control=reno

Доступные в операционной системе алгоритмы можно посмотреть командой:

sysctl net.ipv4.tcp_available_congestion_control

Если вы хотите добавить алгоритм, это можно сделать командой:

/sbin/modprobe tcp_westwood

Здесь Westwood TCP — алгоритм, оптимизированный для линий с потерями. Версий TCP существует десятки. Дополнительную, хотя и не исчерпывающую информацию, можно найти в онлайн книге Ю.Семенова http://book.itep.ru/4/44/tcp.htm

После изменения алгоритма, повторите пункты 6 и 7 для алгоритма TCP Reno для двух рассмотренных сценариев. Ответьте на те же вопросы:

В каком направлении передается основной объем данных?

Сколько потребовалось сегментов чтобы передать веб-страничку?

В скольки сегментах содержался основной объем данных?

Какие изменения вы отмечаете при использовании алгоритма Reno?

Отчет с графиками результатов ваших экспериментов и их анализом загрузите для проверки.

ПРАКТИКА №6
НАСТРОЙКА АУТЕНТИФИКАЦИИ В БЕСПРОВОДНОЙ СЕТИ ПОД УПРАВЛЕНИЕМ КОНТРОЛЛЕРА

Теоретическая информация

Аутентификация в беспроводной сети имеет специфику. Здесь аутентифицируется не пользователь, а устройство, получающее доступ к сети. Стандарт IEEE 802.11 предусматривает два механизма аутентификации беспроводных абонентов: **открытую аутентификацию** (Open Authentication) и **аутентификацию с общим ключом** (Shared Key Authentication). В аутентификации в беспроводных сетях также широко используются два других механизма, выходящих за рамки стандарта *802.11*, а именно назначение идентификатора беспроводной локальной сети (Service Set Identifier - SSID) и аутентификация абонента по его MAC-адресу (MAC Address Authentication). Идентификатор беспроводной локальной сети (SSID) представляет собой конфигурационный атрибут беспроводной сети, позволяющий логически отличать сети друг от друга. В общем случае абонент беспроводной сети должен задать у себя соответствующий SSID для того, чтобы получить доступ к требуемой беспроводной локальной сети.

Процесс аутентификации абонента беспроводной локальной сети IEEE 802.11 состоит из следующих этапов (рис. 1):

Абонент (Client) посылает кадр Probe Request во все радиоканалы.

Каждая точка радиодоступа (Access Point - AP), в зоне радиовидимости которой находится абонент, посылает в ответ кадр Probe Response.

Абонент выбирает предпочтительную для него точку радиодоступа (обычно по уровню сигнала) и посылает в обслуживаемый ею радиоканал запрос на аутентификацию (Authentication Request).

Точка радиодоступа посылает подтверждение аутентификации (Authentication Reply).

В случае успешной аутентификации абонент посылает точке радиодоступа кадр ассоциации (Association Request).

Точка радиодоступа посылает в ответ кадр подтверждения ассоциации (Association Response).

Абонент может теперь осуществлять обмен пользовательским трафиком с точкой радиодоступа и проводной сетью.

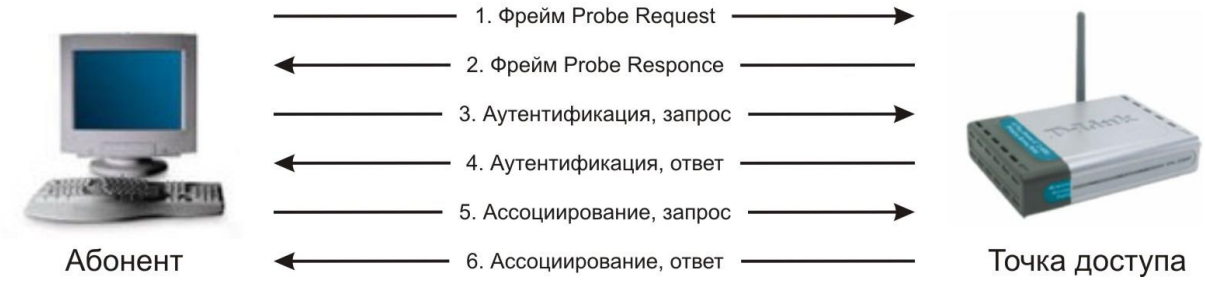


Рис. 1 Этапы аутентификации абонента

При активизации беспроводный абонент начинает поиск точек радиодоступа в своей зоне радиовидимости с помощью управляющих фреймов Probe Request. Фреймы Probe Request посылаются в каждый из радиоканалов, поддерживаемых абонентским радиоинтерфейсом, чтобы найти все точки радиодоступа с необходимыми клиенту идентификатором SSID и поддерживаемыми скоростями радиообмена. Каждая точка радиодоступа из находящихся в зоне радиовидимости абонента, удовлетворяющая запрашиваемым во фрейме Probe Request параметрам, отвечает фреймом Probe Response, содержащим синхронизирующую информацию и данные о текущей загрузке точки радиодоступа. Абонент определяет, с какой точкой радиодоступа он будет работать, путем сопоставления поддерживаемых ими скоростей радиообмена и загрузки. После того как предпочтительная точка радиодоступа определена, абонент переходит в фазу аутентификации.

Открытая аутентификация по сути не является алгоритмом аутентификации в привычном понимании. Точка радиодоступа удовлетворит любой запрос открытой аутентификации. На первый взгляд использование этого алгоритма может показаться бессмысленным, однако следует учитывать, что разработанные в 1997 году методы аутентификации IEEE 802.11 ориентированы на быстрое логическое подключение к беспроводной локальной сети. Вдобавок к этому многие IEEE 802.11-совместимые устройства представляют собой портативные блоки сбора информации (сканеры штрих-кодов и т. п.), не имеющие достаточной процессорной мощности, необходимой для реализации сложных алгоритмов аутентификации.

В процессе открытой аутентификации происходит обмен сообщениями двух типов:

- запрос аутентификации (Authentication Request);
- подтверждение аутентификации (Authentication Response).

Таким образом, при открытой аутентификации возможен доступ любого абонента к беспроводной локальной сети. Если в беспроводной сети шифрование не используется, любой абонент, знающий идентификатор SSID точки радиодоступа, получит доступ к сети. При использовании точками радиодоступа шифрования WEP сами ключи шифрования становятся средством контроля доступа. Если абонент не располагает корректным WEP-ключом, то даже в случае успешной аутентификации он не сможет ни передавать данные через точку радиодоступа, ни расшифровывать данные, переданные точкой радиодоступа (рис. 2).

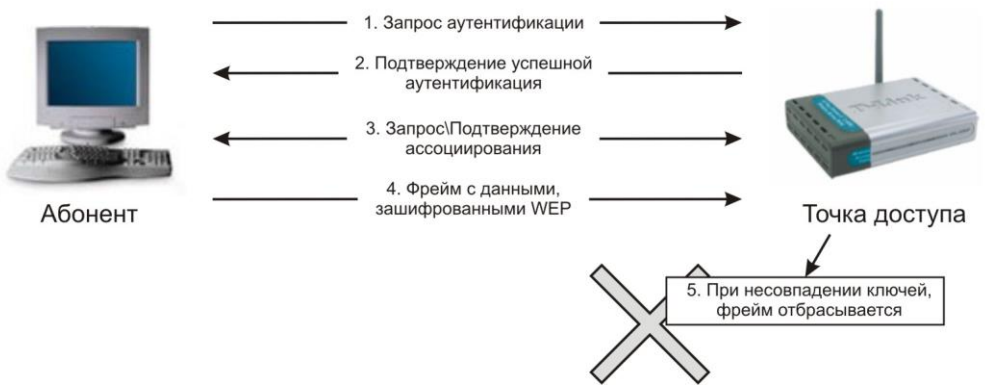


Рис. 2 Аутентификация с общим ключом

Аутентификация с общим ключом является вторым методом аутентификации стандарта IEEE 802.11. Она использует одинаковый ключ шифрования WEP, который

вводится в конфигурацию на точке беспроводного доступа и на клиенте. Процесс аутентификации иллюстрирует рис. 3:

1. Абонент посылает точке радиодоступа запрос аутентификации, указывая при этом необходимость использования режима аутентификации с общим ключом.

2. Точка радиодоступа посылает подтверждение аутентификации, содержащее Challenge Text.

3. Абонент шифрует Challenge Text своим статическим WEP-ключом и посылает точке радиодоступа запрос аутентификации.

4. Если точка радиодоступа в состоянии успешно расшифровать запрос аутентификации и содержащийся в нем Challenge Text, она посылает абоненту подтверждение аутентификации, таким образом предоставляя доступ к сети.

Рис. 3 Аутентификация WEP

Защита беспроводного соединения с помощью ключей WEP оказалась уязвимой к атакам «грубой силой», т.к. длина ключевой последовательности была недостаточно большой, а сам ключ не изменялся. Злоумышленник набрав некоторое число пакетов при прослушивании радиосети может подобрать ключ шифрования перебором всех возможных комбинаций. Поэтому позднее были предложены улучшенные методы защиты WPA, WPA2. Несмотря на улучшение алгоритмов защиты и смену ключей шифрования при каждой отправке нового пакета в беспроводную сеть, оба метода также основывались на аутентификации с общим ключом, англ. PSK – Pre Shared Key. PSK также известен, как четырёхэтапное установление связи, поскольку именно столько сообщений необходимо передать между точкой беспроводного доступа и подсоединяющимся устройством, чтобы подтвердить, что они договорились по поводу пароля, при том, что ни одна из сторон не сообщает его другой. До 2016 года PSK казался безопасным, а потом была открыта атака с переустановкой ключа (Key Reinstallation Attacks, KRACK). В 2018г был предложен новый стандарт защиты: WPA3, в котором применен новый метод аутентификации устройства, SAE, Simultaneous Authentication of Equals. Это разновидность dragonfly handshake [рукопожатия по методу стрекозы], поэтому можно встретить название WPA3 Dragonfly, рис. 4.

Рис. 4 Стандарты безопасности беспроводных сетей по времени введения

Несмотря на усложнение алгоритма для защиты от словарных атак и то что WPA3 пока не получил широкого распространения, в 2019 году исследователи показали слабости нового протокола в статье Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd.

Поэтому защита современной беспроводной сети никогда не строится только на аутентификации устройств, дополнительно применяются традиционные методы аутентификации пользователей.

При создании большой распределенной беспроводной сети настройка множества точек радиодоступа становится очень трудоемкой задачей. Для решения этой проблемы были созданы специальные устройства — контроллеры беспроводной сети, WLC - Wireless LAN Controller, а также «упрощенные» точки беспроводного доступа, LAP - Lightweight Access Point, специально предназначенные для подключения к контроллеру. Некоторые модели LAP могут работать самостоятельно, без подключения к контроллеру, но большинство не могут. Контроллер берет на себя большую часть функций управления, в том числе аутентификацию подключающихся клиентов. Между контроллером и LAP, подключенными в общую локальную сеть предприятия, создается защищенное туннельное соединение по специальному протоколу CAPWAP, рис. 5

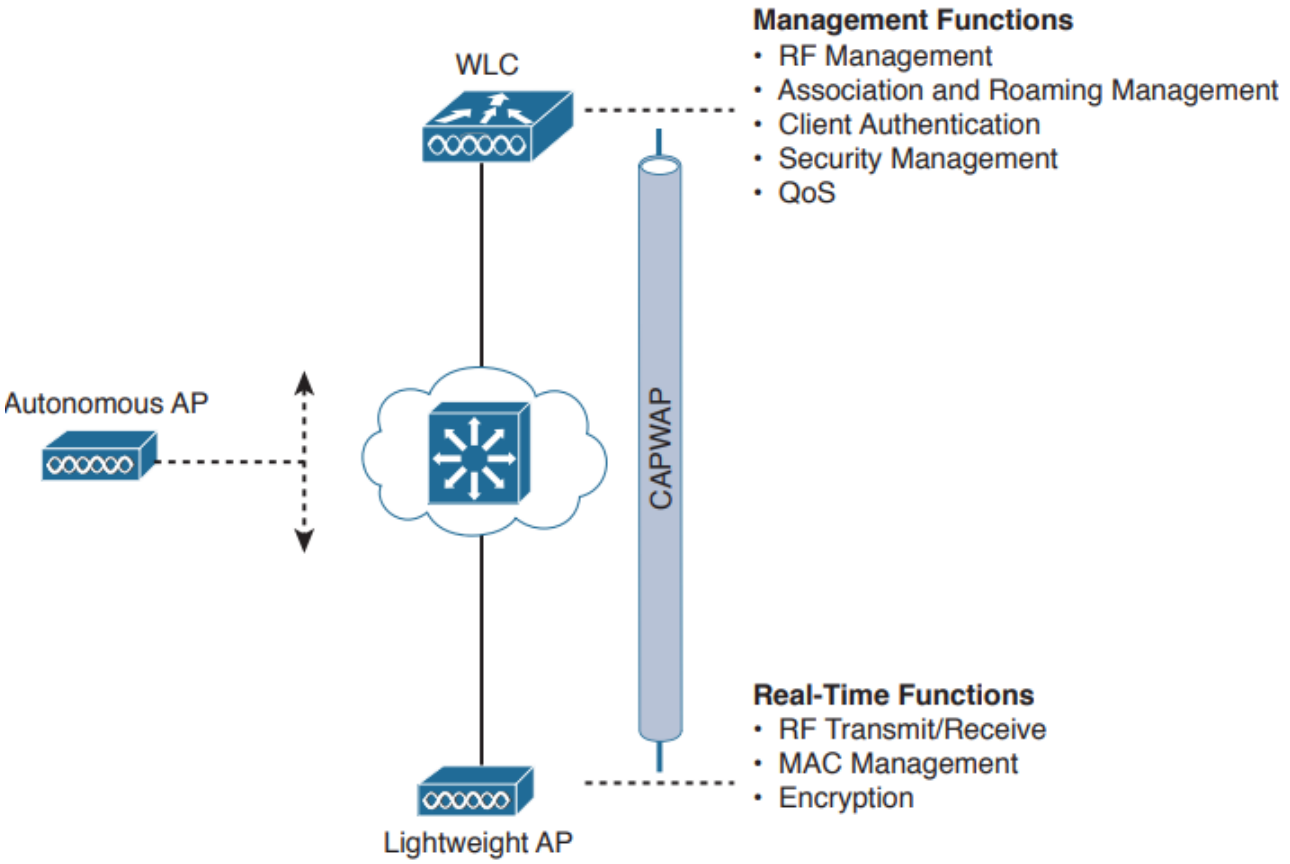


Рис. 5 Туннель CAPWAP между WLC и LAP

Контроллер может поддерживать одновременную работу с десятками точек радиодоступа.

Еще один шаг в развитии технологий аутентификации в беспроводной сети был сделан с развитием «облачных» сервисов. Можно сказать, что наиболее современным способом управления беспроводной сетью являются облачные сервисы подобные Cisco Meraki, FortiGate NGFW и др. Это комплексные решения, не ориентированные только на поддержку беспроводных сетей, но включающие их как один из элементов. Облачный сервис и специализированные устройства безопасности: security appliances, позволяют быстро развернуть защищенную информационную инфраструктуру предприятия, соединяющую географически распределенные подразделения криптографически защищенными VPN-туннелями. Настройка механизмов защиты, в том числе беспроводных сетей и аутентификации в них, выполняется через веб-интерфейс облачного сервиса, рис. 6.

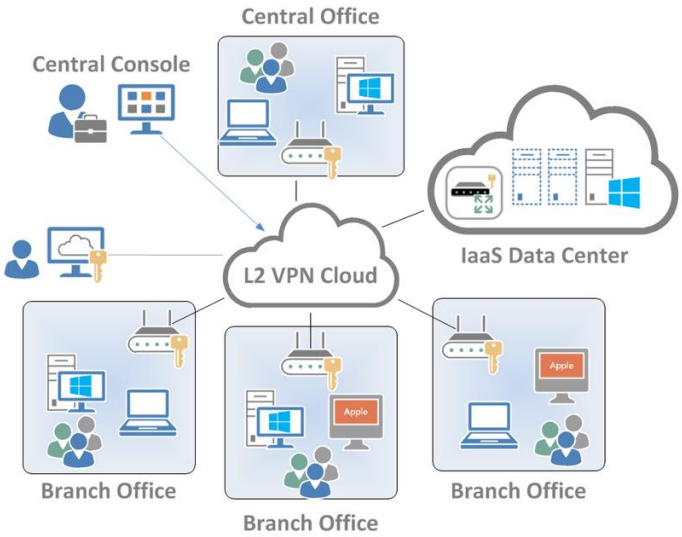


Рис. 6 Облачный сервис управления решениями безопасности

ВЫПОЛНЕНИЕ РАБОТЫ

Для выполнения работы нужно использовать версию Cisco Packet Tracer 8.2.1. Вы настраиваете локальную сеть с участком беспроводной сети под управлением локального контроллера WLC.

Создайте топологию, как показано на рис. 7

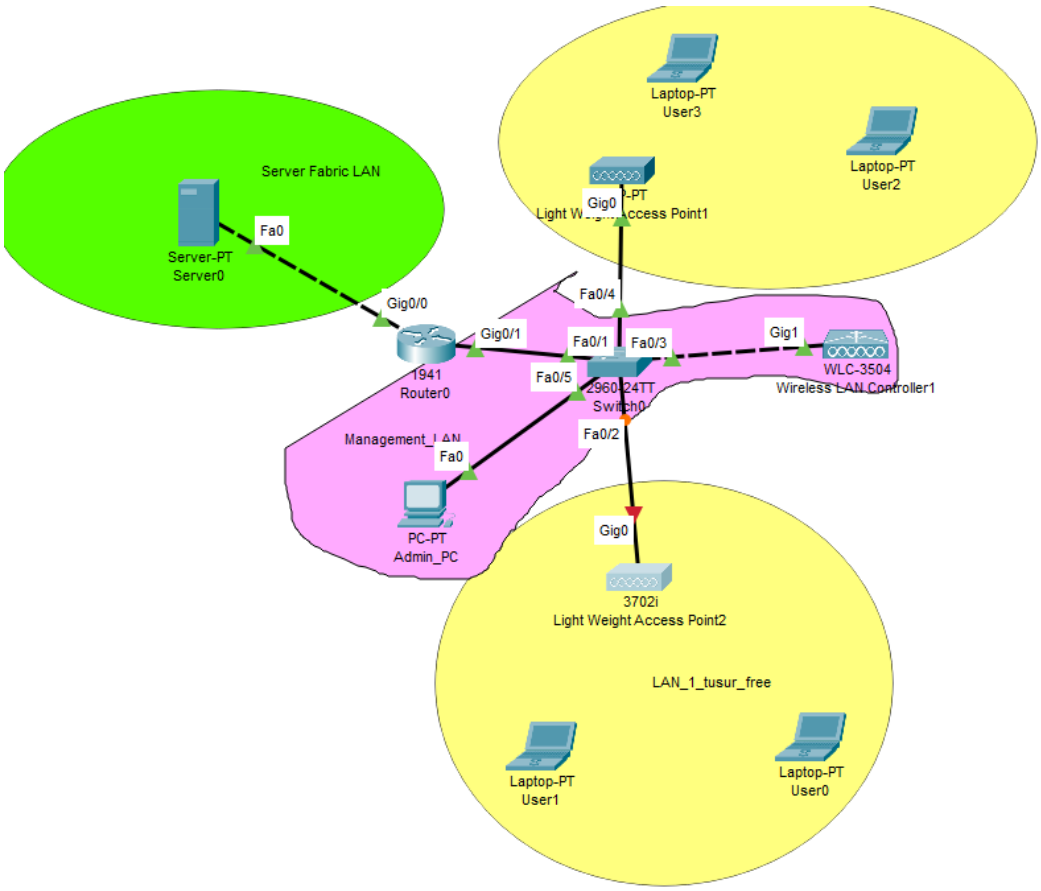


Рис. 7 Топология практической работы часть 1

Беспроводные устройства находятся в группе сетевого оборудования, рис. 8. В качестве контроллера беспроводной сети возьмите WLC-3504, а в качестве LAP – 3702i. Гигабитный порт контроллера №1 подключите к гигабитному порту коммутатора. Ко второму гигабитному порту подключите компьютер администратора. Это важно!

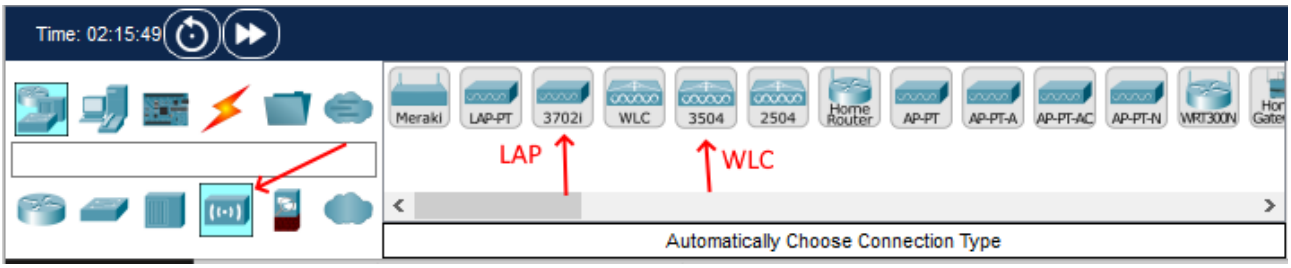


Рис. 8 Выбор беспроводных устройств

Точка беспроводного доступа не имеет встроенного источника питания и для ее включения нужно подключить внешний источник перетаскиванием его рисунка снизу на точку доступа на вкладке физических свойств, так чтобы разъем шнура подключения попал в разъем питания на устройстве, рис. 9.

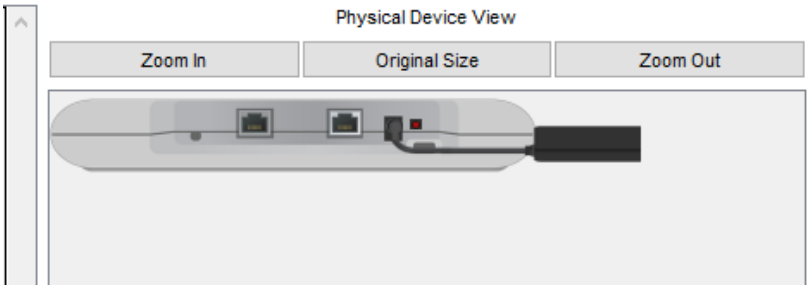


Рис. 9 Подключение источника питания к LAP

В топологии задействованы три VLAN:

1. VLAN-10 – для пользовательских компьютеров в беспроводной сети
2. VLAN-20 – то же, что VLAN-10
3. VLAN-99 – для управления

Каждой VLAN сопоставляется подсеть на сетевом уровне. Для VLAN10 и VLAN20 это две подсети из вашей практики №1, VLAN-99 использует подсеть 172.16.99.0 /24.
Кроме того, в топологии есть подсеть серверной фермы, представленной одним веб-сервером. В ней используется адресный блок 172.16.0.0 /24.

Настройте сетевые интерфейсы сервера и маршрутизатора для подсети серверной фермы.

Интерфейс маршрутизатора в подсети управления будет выполнять маршрутизацию между всеми VLAN. Создайте на нем соответствующие подынтерфейсы и задайте первый доступный адрес, например 172.16.99.1 для подынтерфейса, соответствующего VLAN-99.

Также на маршрутизаторе включите DHCP-сервис для абонентов беспроводной сети, подключающихся к точкам LAP в соответствующие VLAN. Это можно сделать командами в глобальной конфигурации маршрутизатора:
ip dhcp pool <вашеФИО>_vlan10

network <ваша сеть №1>
default-gateway <адрес шлюза в вашей сети №1>
dns-server 8.8.8.8 (можно указать как адрес шлюза)
ip dhcp excluded-address <первый и последний адрес диапазона, который вы хотите убрать из заданной сети, например 172.16.99.1 172.16.99.100, адреса начнут выдаваться с 172.16.99.101>. Исключите 50 адресов. На маршрутизаторе должно быть настроено два пула адресов: для VLAN-10 и VLAN-20.

Все интерфейсы на коммутаторе, к которым подключены маршрутизатор, контроллер WLC, точки LAP, компьютер администратора, переведите в режим транка командой:
switchport mode trunk в контексте настройки интерфейса.

По умолчанию контроллер беспроводной сети имеет настройки интерфейса управления, как показано на рис. 10. На нем нет никаких аккаунтов до первого подключения. Настройте интерфейс компьютера администратора (Admin_PC) на рис. 7 в ту же сеть, откройте вкладку Desktop и браузер для подключения к WLC. Введите в адресную строку адрес контроллера, нажмите Enter или кнопку Go. Подождите некоторое время. Можно нажать кнопку ускорения включения интерфейсов (двойная стрелка слева внизу). Быстрое моргание светодиодов на линиях подключения компьютера и контроллера говорит о нормальном процессе. В конце концов откроется окно создания нового аккаунта администратора, рис. 11.

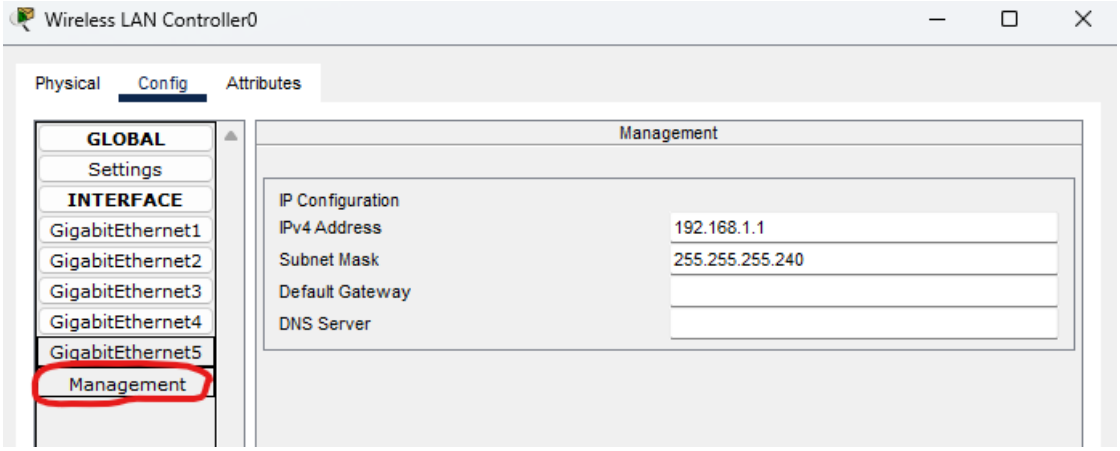


Рис. 10 Настройки интерфейса управления WLC по умолчанию

Введите в этом окне логин и пароль. Для пароля задана политика: обязательно заглавные и прописные символы, длина не менее восьми, например логин **admin**, пароль **Tusur123**.

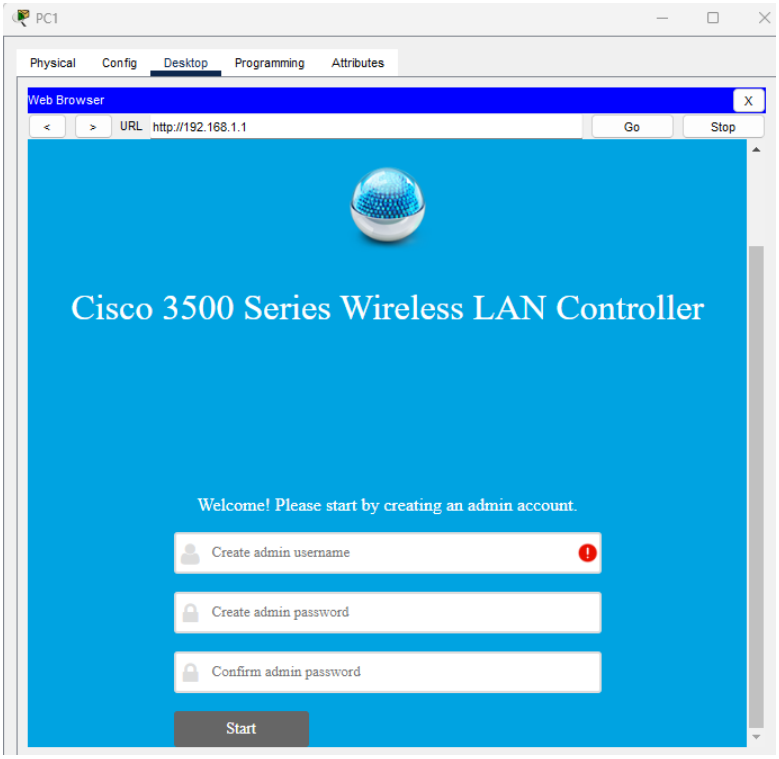


Рис. 11 Окно создания аккаунта администратора WLC

После нажатия клавиши Start откроется первое окно мастера настройки, рис. 12. Здесь необходимо задать имя: поле System Name, оно может быть любым. Адрес управления. Первоначально использовался адрес 192.168.1.1, но мы определили что VLAN управления будет 99 и для нее будет использоваться адресное пространство 172.16.99.0/24, поэтому адрес и маску нужно задать соответствующие. То же касается и шлюза по умолчанию. В поле Management VLAN ID тоже следовало бы поставить 99, но для Cisco Packet Tracer здесь должно быть только 1, поэтому замените 0 на 1. Нажмите клавишу Next. Откроется окно следующего шага, рис. 13, где необходимо создать первую WLAN, задав для нее имя, уровень защиты и пароль доступа. Задайте имя в виде <ваше ФИО>_vlan10, и пароль, например, 12345678. WPA2 Personal оставьте как есть и нажмите клавишу Next.

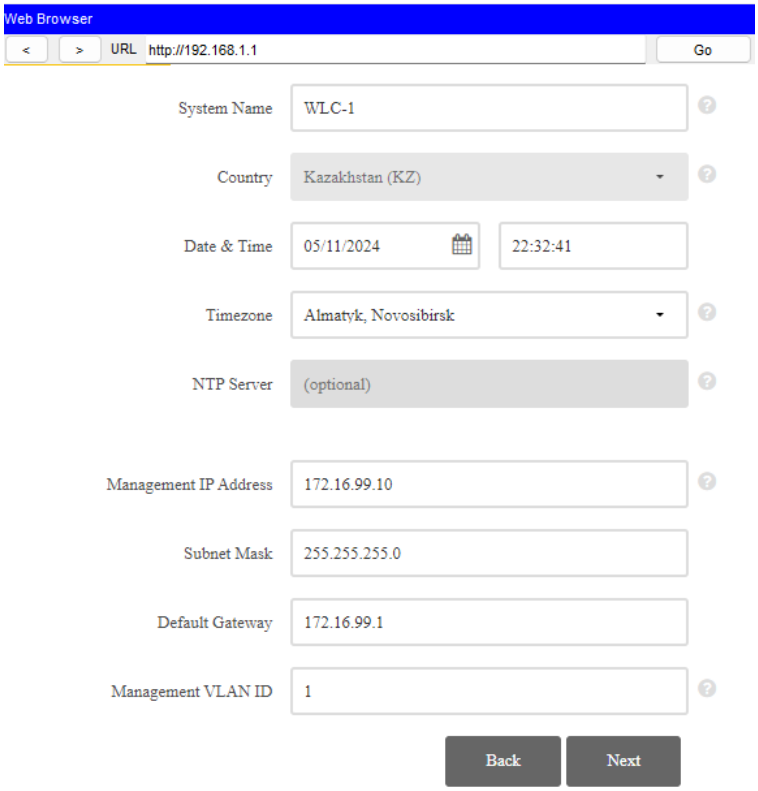


Рис. 12 Стартовое окно настройки WLC

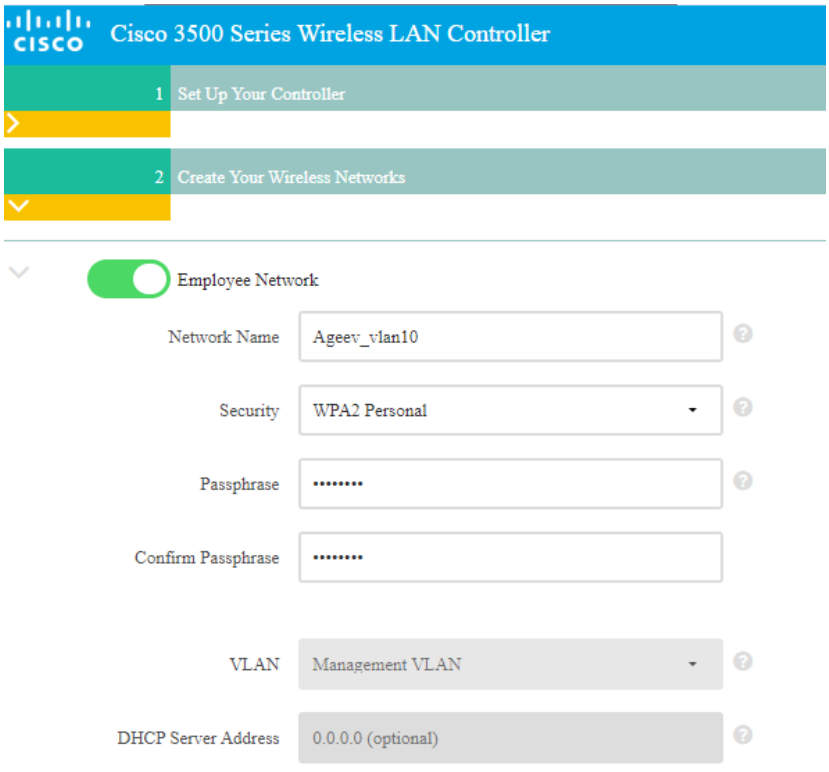


Рис. 13 Настройка первой беспроводной сети

На следующем и последнем окне мастера настройки не нужно ничего менять, просто нажмите Next. Откроется сводное окно, демонстрирующее все заданные настройки, вы можете их еще раз проверить и, если нет никаких ошибок, нажмите клавишу Apply. Выскочит еще одно окошко с подтверждением, в котором нужно нажать ОК. После чего

настройки применятся, но будет висеть крутящееся колесо с просьбой подождать. Ждать не нужно, нужно закрыть окно браузера, иначе это колесо будет крутиться до бесконечности. Так как мы задали новый адрес управления на WLC, нужно изменить адрес на компьютере администратора, задав его в сети 172.16.99.0/24.

Вновь откройте окно браузера на компьютере и введите заданный при настройке адрес управления, только сначала укажите протокол https в адресной строке, т. к. теперь, когда на контроллере есть аккаунт админа, разрешается только защищенное подключение, рис. 14.

Рис. 14 Повторное подключение к контроллеру в защищенном режиме

Нажмите кнопку Login и в открывшемся окне введите заданный ранее логин и пароль административного доступа. Откроется окно веб-интерфейса настройки контроллера, рис. 15. Мы видим здесь сводную информацию по устройству. Например, отмечается, что контроллер поддерживает до 150 точек доступа. Указана версия ПО, загрузка процессора, работа вентилятора, температура устройства и другая системная информация. На странице сводной информации есть информация о подключенных точках доступа. Вы можете видеть что сейчас их 0. Зайдите в раздел настройки точек доступа и укажите адрес управления WLC, рис. 16.

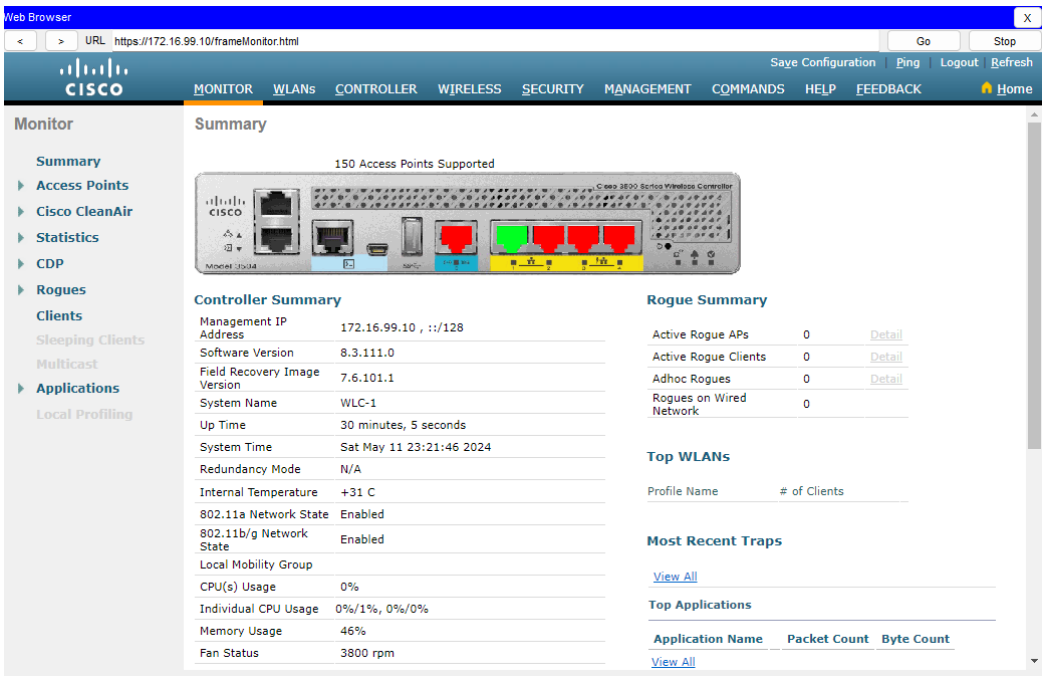


Рис. 15 Стартовая страница веб-интерфейса WLC

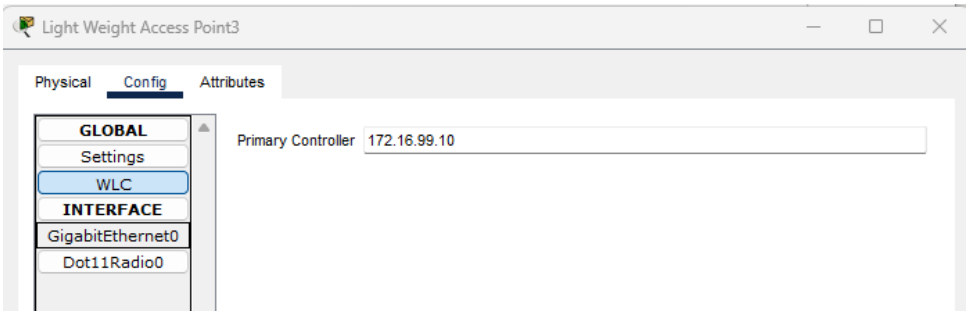


Рис. 16 Задание адреса WLC на точке доступа

Откройте пункт меню Controller в веб-интерфейсе управления контроллером. Выберите пункт Internal DHCP Server, рис. 17, выберите пункт DHCP Scope. Удалите дефолтовый диапазон адресов. Создайте новый с адресным пространством сети управления, рис. 18.

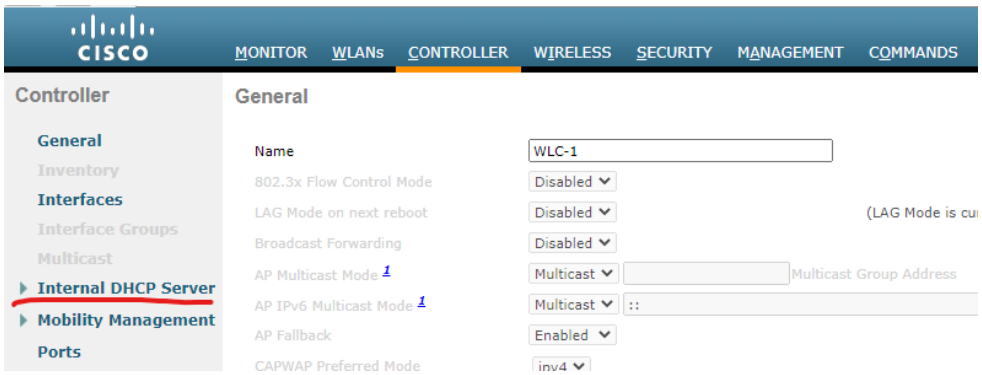


Рис. 17 Настройка встроенного DHCP-сервера на WLC

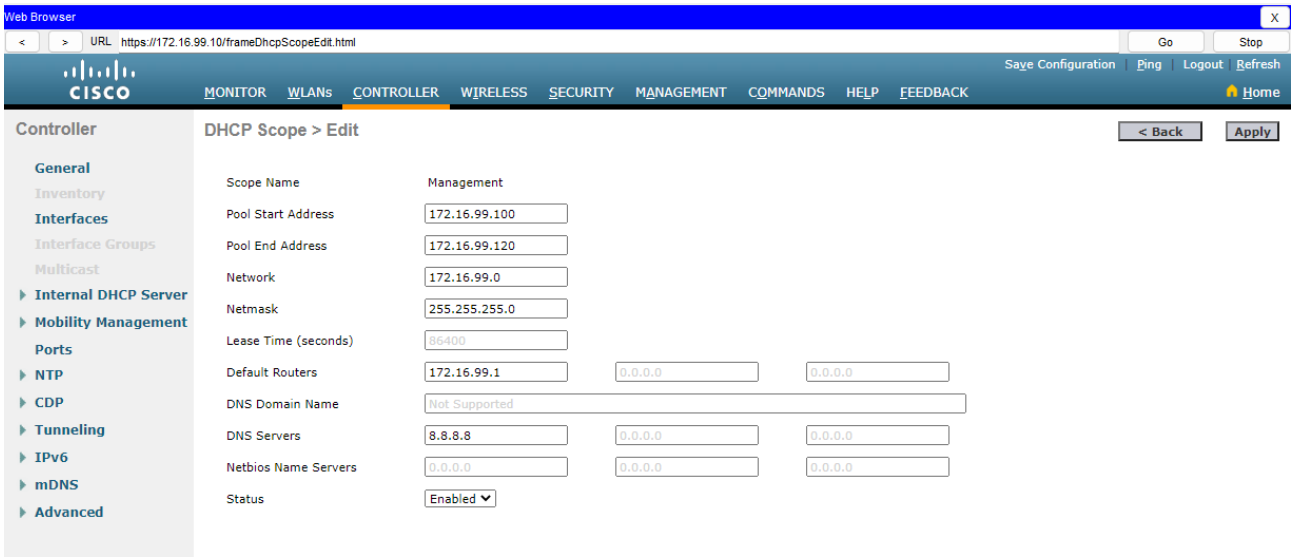


Рис. 18 Настройка адресного диапазона и других параметров DHCP

Нажмите клавишу Apply в правом верхнем углу. На точках LAP выберите получить конфигурацию IP по DHCP, рис. 19.

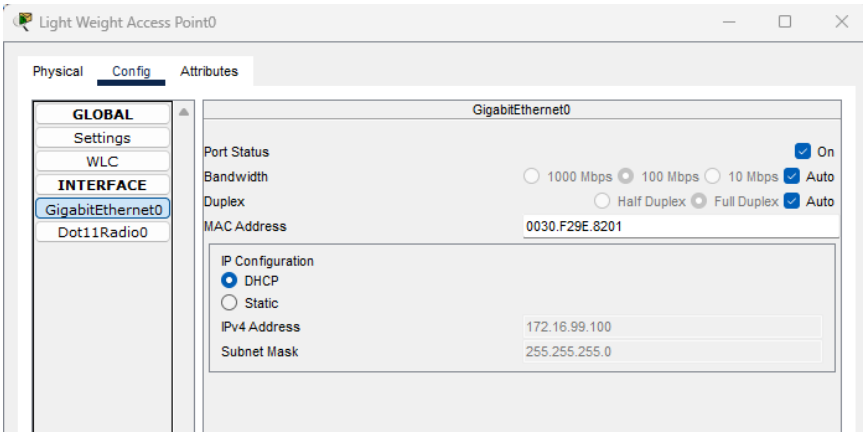


Рис. 19 Получение адреса на точках доступа от сервера DHCP на WLC

На вкладке Monitor главного меню управления вы теперь видите две подключенные точки, рис. 20.

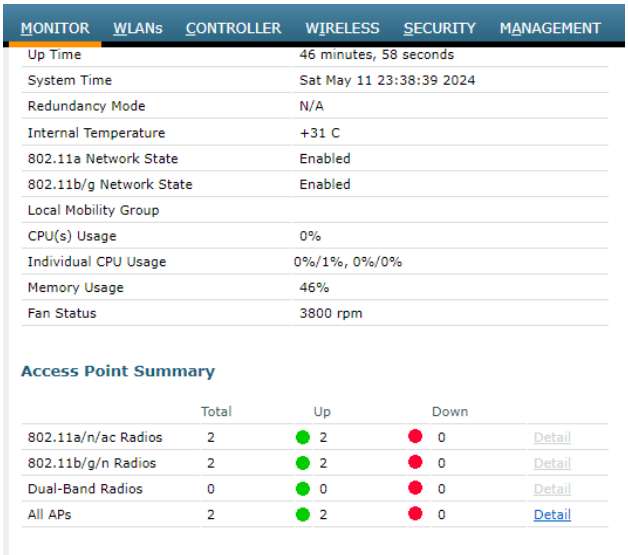


Рис. 20 В разделе Access Point Summary отображается число подключенных LAP

При переходе по ссылке Detail в этом разделе можно видеть имя и адрес точек доступа, рис. 21

Рис. 21 Подробная информация о подключенных точках доступа

Во всплывающем окне, появляющемся при наезде мышкой на LAP можно видеть что между точкой беспроводного доступа и WLC сформирован CAPWAP туннель, рис. 22.

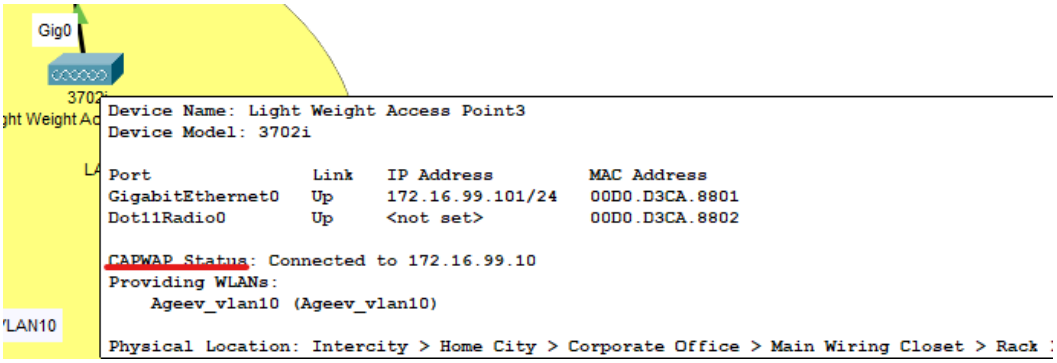


Рис. 22 Туннельное подключение LAP к WLC

Интерфейсы WLC (Wi-Fi контроллера) - это логические и физические порты, через которые контроллер взаимодействует с другими сетевыми устройствами и управляет беспроводными точками доступа. Основные интерфейсы WLC включают:

1. **Интерфейс управления** (Management Interface) - это логический интерфейс, через который контроллер «общается» с внешним миром (кроме точек доступа). Обычно настраивается статический IP-адрес для доступа к веб-интерфейсу и CLI контроллера.

2. **Динамические интерфейсы** (Dynamic Interfaces) - это VLAN-интерфейсы, которые используются для туннелирования клиентского трафика. Для каждого SSID настраивается свой динамический интерфейс с уникальным VLAN.

3. **Портал гостевого доступа** (Guest-Anchor) - специальный интерфейс для туннелирования трафика гостевых клиентов. Позволяет сегментировать гостевой трафик.

4. **Физические порты Ethernet** - используются для подключения контроллера к проводной сети. Например, порт 10/100/1000 Ethernet.

Важное значение имеет **виртуальный** интерфейс WLC, он используется для служебных функций, например, ретрансляции трафика DHCP, перевода подключающихся клиентов на страничку аутентификации и др. Откройте в меню Controller слева пункт Interfaces, рис. 23.

Создайте динамические интерфейсы для VLAN-10 и 20, чтобы обеспечить связь VLAN и соответствующих им WLAN. Нажмите кнопку New… и задайте имя интерфейса и соответствующую ему VLAN, рис. 24.

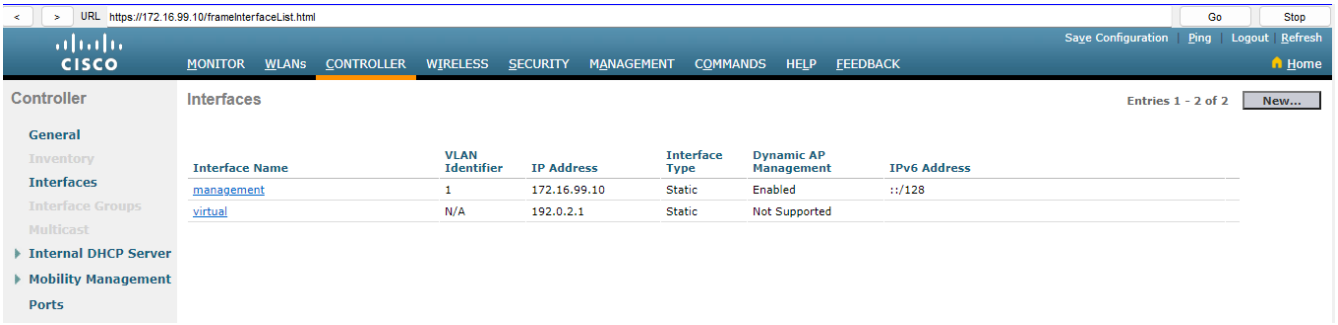


Рис. 23 Интерфейсы на WLC

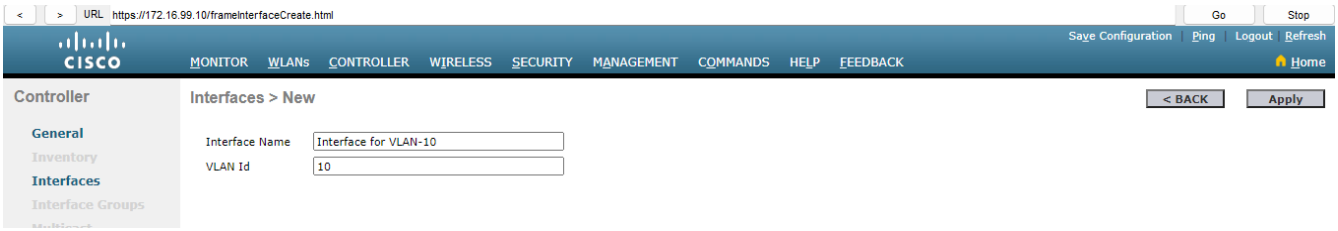


Рис. 24 Создание нового динамического интерфейса

На следующей страничке, которая открывается после нажатия кнопки Apply для создания интерфейса, нужно добавить конфигурационную информацию, рис. 25:

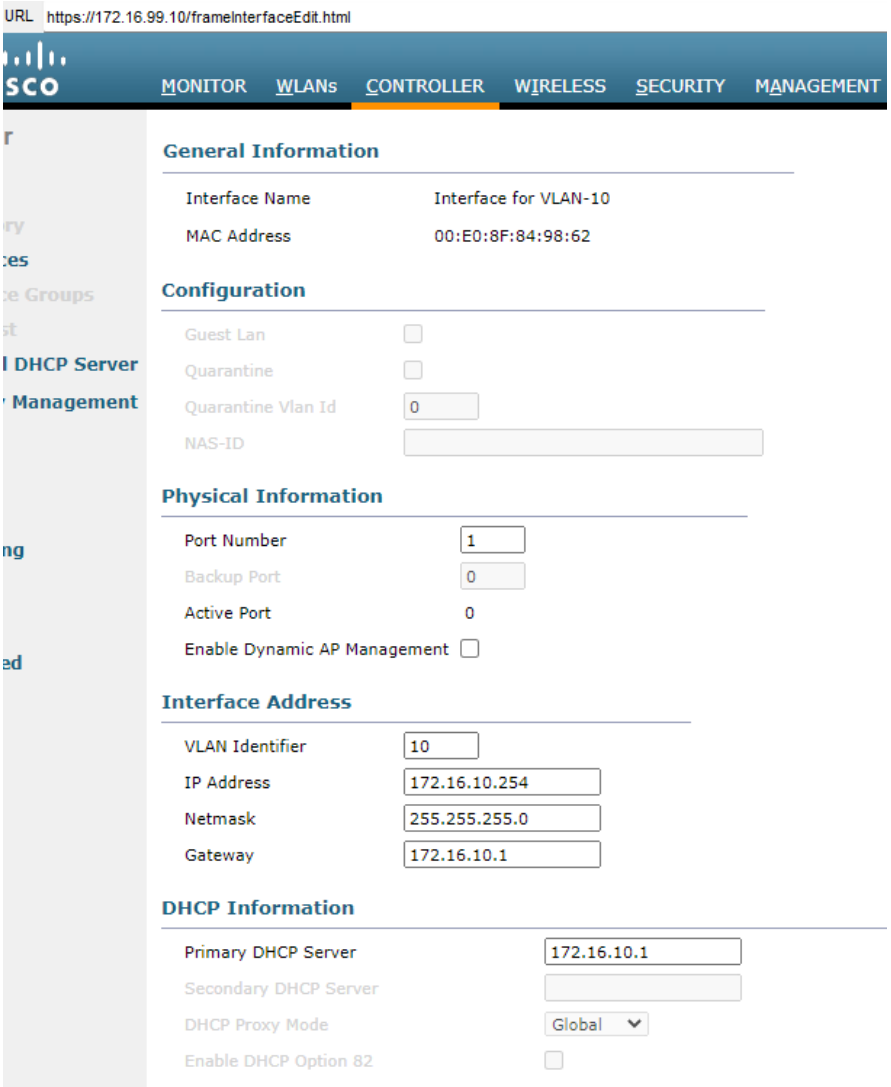


Рис. 25 Настройка созданного на WLC интерфейса

1. Номер порта в разделе физической информации. Это номер физического интерфейса на WLC, с которым будет связан создаваемый динамический интерфейс. Если вы выбрали для подключения WLC первый порт GigabitEthernet, то нужно ввести номер 1.
2. Адрес создаваемого интерфейса. Интерфейс будет находиться в той подсети, которую вы определили для VLAN-10. У вас это ваша персональная подсеть из практики 1. В примере для наглядности показана подсеть 172.16.10.0/24. Шлюзом в другие сети является подынтерфейс маршрутизатора, который вы создали для VLAN-10.
3. Главный DHCP сервер. Здесь мы указываем внешний DHCP сервер, чтобы продемонстрировать возможности использования как внутреннего, так и внешнего сервера DHCP. Этот сервер вы должны были настроить на маршрутизаторе для обоих VLAN. Поэтому указываем соответствующий VLAN интерфейс маршрутизатора.

Нажимаем клавишу Apply и подтверждаем внесение изменений. Аналогичным образом создаем второй интерфейс для VLAN-20. Теперь на вкладке Interfaces должны отображаться созданные интерфейсы, рис. 26.

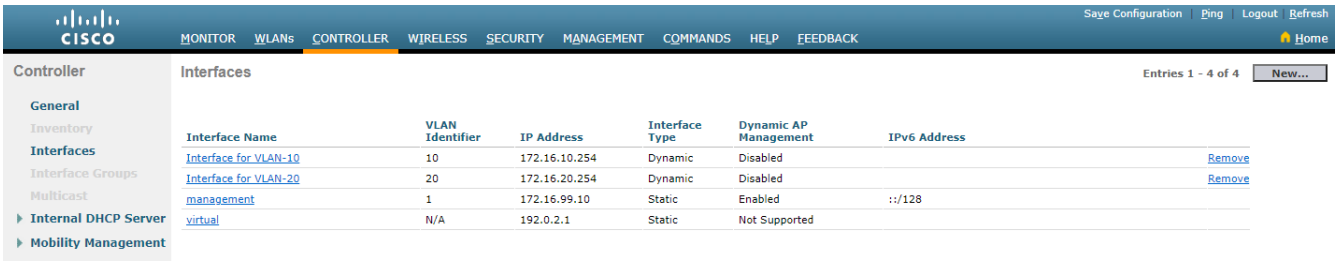


Рис. 26. Просмотр добавленных интерфейсов.

Перейдите в пункт меню WLANs. Здесь отображается первая беспроводная сеть, созданная в мастере первоначальной настройки, рис. 27

Рис. 27 Просмотр настроенных на WLC беспроводных сетей

Перейдите по ссылке с идентификатором первой WLAN в ее настройки и исправьте интерфейс с management на интерфейс для соответствующей VLAN, рис. 28. Дополнительно, перейдите на вкладку Advanced прокрутите вниз страничку и включите параметры **FlexConnect Local Switching** и **FlexConnect Local Auth**. Эти параметры позволяют более эффективно обрабатывать трафик в реальных условиях на реальных сетях. В данном случае это больше акцентирование внимания на этих параметрах для обучающегося и необходимое условие чтобы настроенная в Packet Tracer беспроводная сеть правильно работала.

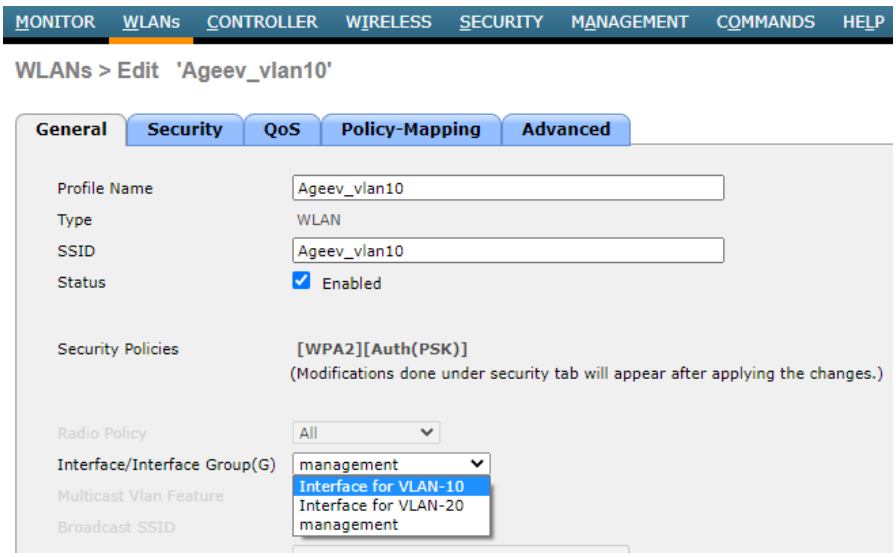


Рис. 28 Изменение интерфейса для WLAN

Вернитесь в главное меню WLANs и создайте еще одну беспроводную сеть для VLAN-20, нажатием на клавишу «Create New» **Go**. Задайте для нее такие же настройки как для VLAN-10.

На ноутбуках в сетевой топологии замените сетевой адаптер с проводного на беспроводной и откройте приложение PC Wireless на вкладке Desktop, рис. 29.

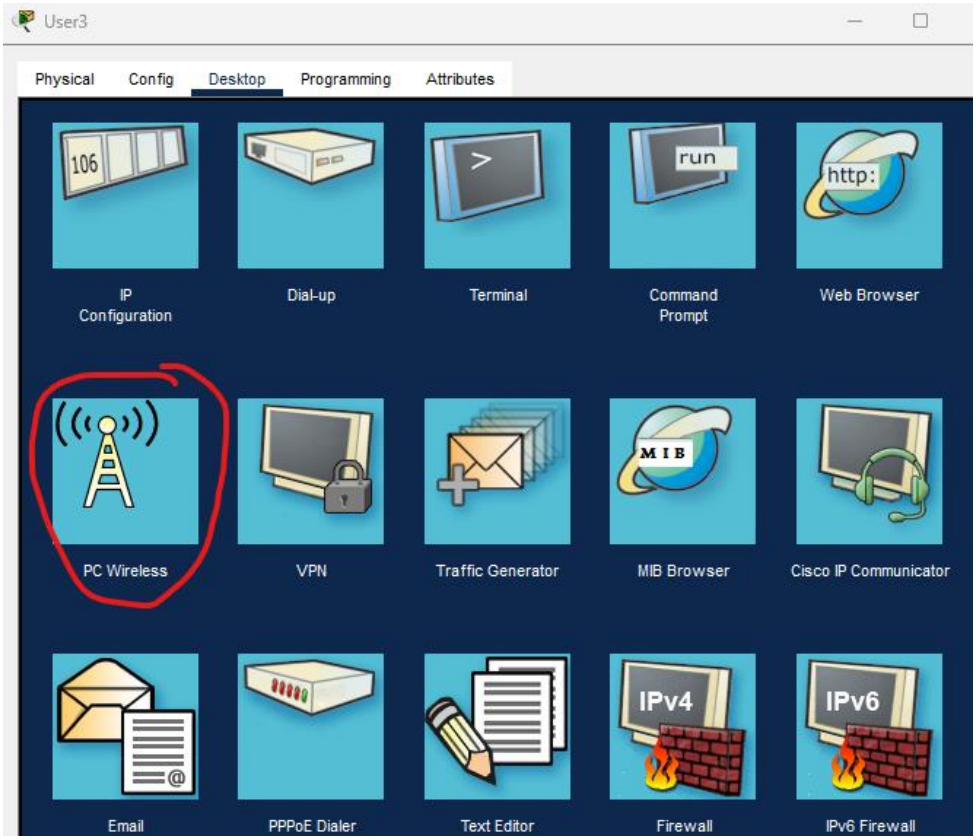


Рис. 29 Приложение для поиска и подключения к беспроводной сети на рабочем столе компьютера

Перейдите на вкладку Connect и дождитесь появления в окне списка доступных Wi-Fi сетей, подключитесь к сети, соответствующей заданной VLAN. В настройках сетевого адаптера задайте получение конфигурации по DHCP. Какой адрес получил компьютер?

Подключите к беспроводной сети все четыре ноутбука, рис. 30.

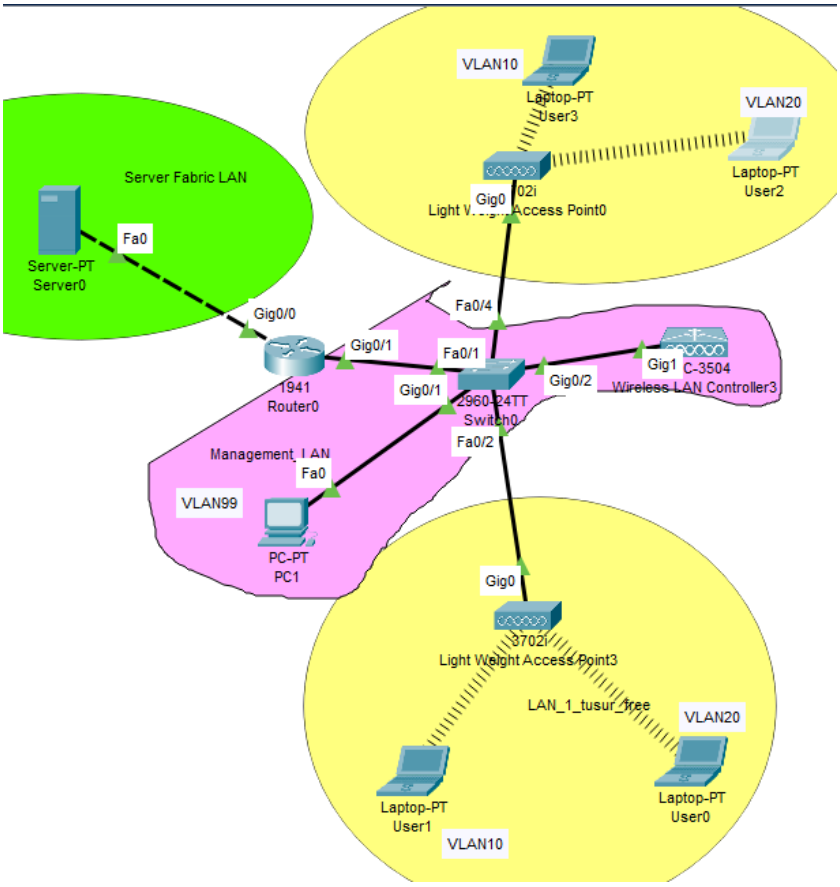


Рис. 30 Подключение ноутбуков к беспроводной сети

Проверьте, что ноутбуки могут обращаться к компьютерам в других VLAN с помощью эхо-запросов и все они могут открыть страничку на веб-сервере, рис. 31.

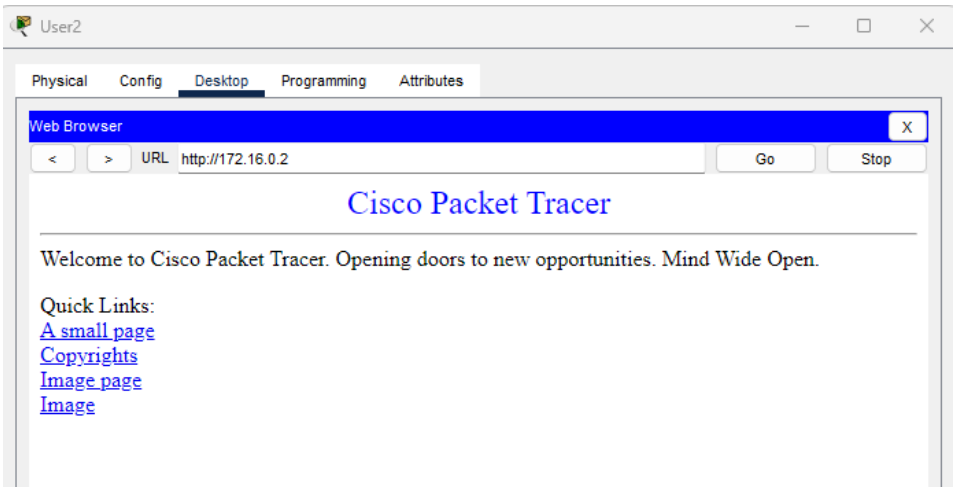


Рис. 31 Открытие индексной страницы на веб-сервере

Выполненную работу сохраните в файле с вашей фамилией и номером группы и загрузите на проверку. Отчета по работе не требуется, отчетом является файл Cisco Packet Tracer с настроенной топологией.

ПРАКТИКА №7
ПРОТОКОЛ РАСШИРЕННОЙ АУТЕНТИФИКАЦИИ, СТАНДАРТ IEEE 802.1X

Расширяемый протокол аутентификации (EAP) — это платформа аутентификации, часто используемая в сети и подключениях к Интернету. EAP определен в RFC 3748 и обновлен в RFC 5247. EAP считается платформой аутентификации, потому что он описывает общие функции и принципы аутентификации, называемые методами EAP. В настоящее время определено около 40 различных методов. Множество методов определено в RFC, а также существует ряд методов, специфичных для конкретных поставщиков. EAP не является именно протоколом, он определяет только информацию, касающуюся работы интерфейсов и форматов сообщений аутентификации. Каждый протокол, использующий EAP, определяет способ инкапсуляции сообщений EAP пользователя в сообщения этого протокола. EAP широко применяется на практике. Например, в IEEE 802.11 (WiFi) стандарты WPA и WPA2 приняли стандарт аутентификации IEEE 802.1X (с различными типами EAP) в качестве канонического механизма аутентификации.

Таким образом, стандарт IEEE 802.1X это конкретная реализация платформы расширенной аутентификации EAP в виде протокола. Стандарт IEEE 802.1X определяет контроль доступа на основе портов и протокол аутентификации, который запрещает не прошедшим аутентификацию и авторизацию рабочим станциям подключаться к локальной сети через общедоступные порты коммутатора. Сервер аутентификации проводит аутентификацию каждой рабочей станции, подключаемой к порту коммутатора, прежде чем предоставлять любые сервисы коммутатора или LAN. На рис. 1 показано, что в сценарии аутентификации 802.1X устройства в сети имеют определенные роли:

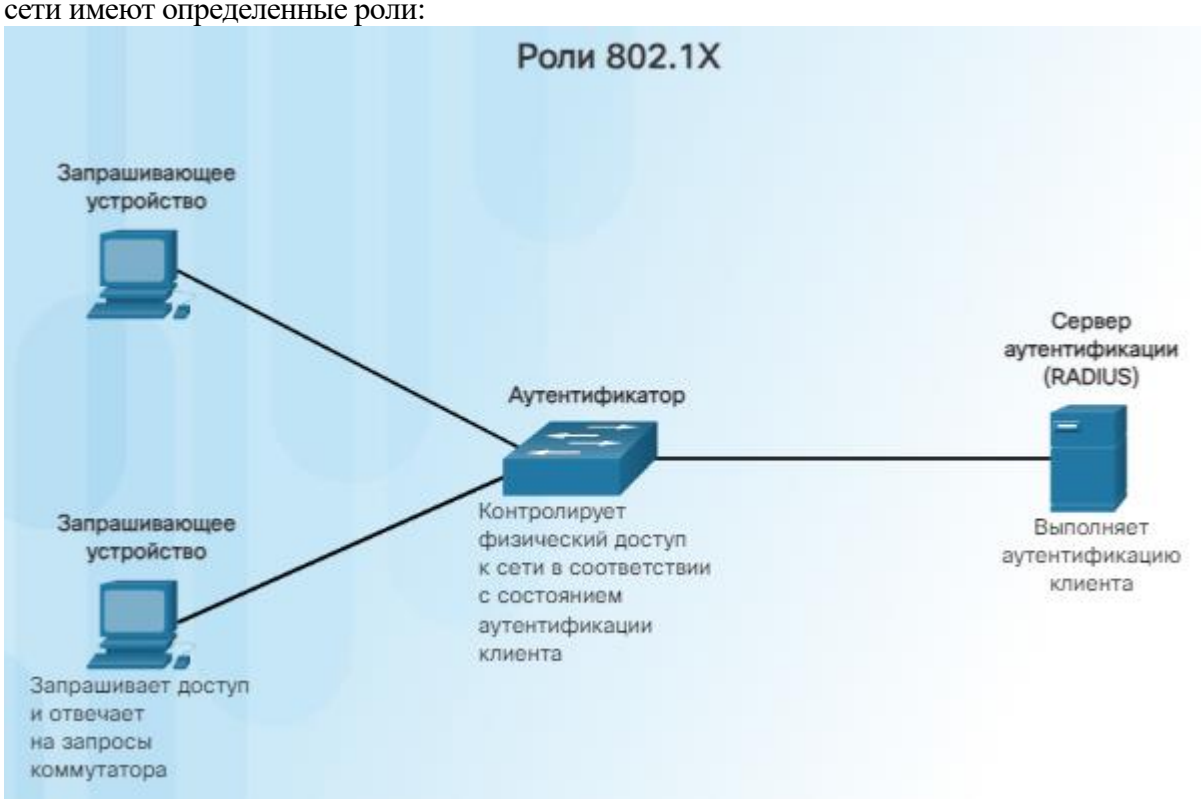


Рис. 1 Роли устройств в платформе расширенной аутентификации EAO

- **Запрашивающее устройство (англ. Supplicant)** - Устройство (рабочая станция), которое запрашивает доступ к сервисам LAN. На рабочей станции должно использоваться клиентское ПО, совместимое со стандартом 802.1X. (Порт коммутатора, к которому подключается клиент, относится к порту запрашивающего устройства согласно спецификации IEEE 802.1X.)

- **Аутентификатор (англ. Authenticator, в роли которого выступает коммутатор)** - Контролирует физический доступ к сети, руководствуясь состоянием аутентификации узла сети. Коммутатор выступает в роли посредника между клиентом и сервером аутентификации. Он запрашивает у клиента идентификационные данные, проверяет их на сервере и ретранслирует клиенту отклик сервера. В составе коммутатора имеется программный агент сервера централизованной аутентификации RADIUS, который отвечает за инкапсуляцию и декапсуляцию кадров расширяемого протокола аутентификации EAP, а также за взаимодействие с сервером аутентификации.
- **Сервер аутентификации (англ. Authentication server)** - Непосредственно выполняет аутентификацию клиента. Сервер аутентификации проверяет подлинность клиента и сообщает коммутатору, есть ли у клиента полномочия на доступ к сервисам LAN и коммутатора. Поскольку коммутатор выступает в качестве посредника, служба аутентификации для клиента прозрачна. Система безопасности RADIUS с расширениями EAP – единственный поддерживаемый сервер аутентификации.

До прохождения аутентификации рабочей станции, средства контроля доступа 802.1X пропускают через порт только трафик протокола расширяемой аутентификации по LAN (EAPOL). После успешной аутентификации разрешается пересылка через порт обычного трафика.
Состояние порта коммутатора определяет, предоставлен ли клиенту доступ в сеть. При настройке для аутентификации 802.1X на базе портов начальным состоянием порта является неавторизованное состояние. В этом состоянии порт запрещает весь входящий и исходящий трафик, кроме пакетов протокола 802.1X. После успешной аутентификации клиента порт переходит в авторизованное состояние, и весь трафик клиента может пересылаться обычным образом. Если коммутатор запрашивает идентификацию клиента и клиент не поддерживает 802.1X, порт остается в неавторизованном состоянии, и клиенту не предоставляется доступ к сети.

ВЫПОЛНЕНИЕ РАБОТЫ

Работа выполняется в программе Cisco Packet Tracer версии 8.2.1. Команды настройки аутентификации IEEE 802.1X не поддерживаются в версии 6.2. Установочный дистрибутив программы можно загрузить из курса. При установке программы и перед ее запуском отключите подключение Интернет. Программа попытается проверить наличие у вас зарегистрированной учетной записи в системе сетевых академий Cisco или на портале «Навыки для всех». При отсутствии подключения Интернет программа запускается и работает нормально. После запуска программы можно восстановить подключение Интернет. Повторных проверок не выполняется.
Откройте программу и постройте топологию в соответствии с рис. 2

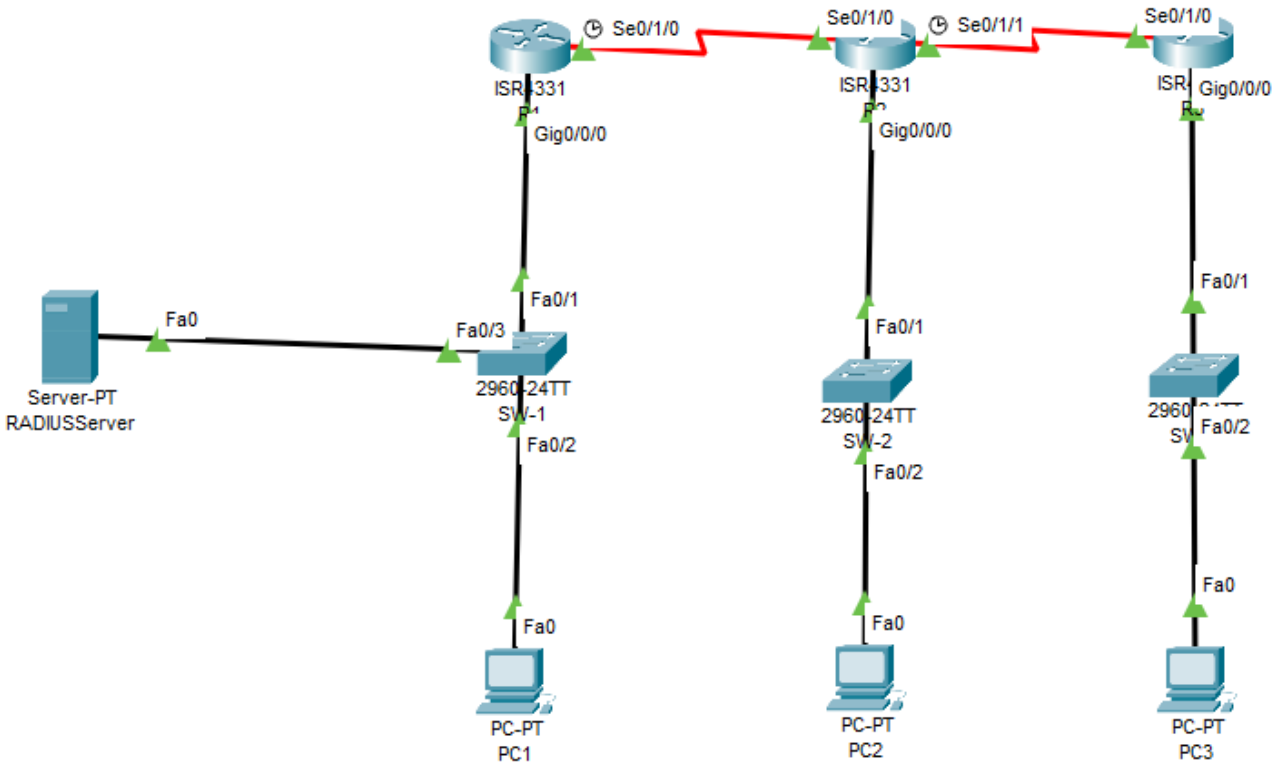


Рис. 2 Топология сети выполняемой работы

В качестве маршрутизаторов используйте маршрутизатор с интегрированными сервисами ISR4331. На вкладке физического представления маршрутизатора добавьте в него модуль поддержки синхронных последовательных соединений NM-2T, предварительно выключив питание. Как показано на рис. 3.

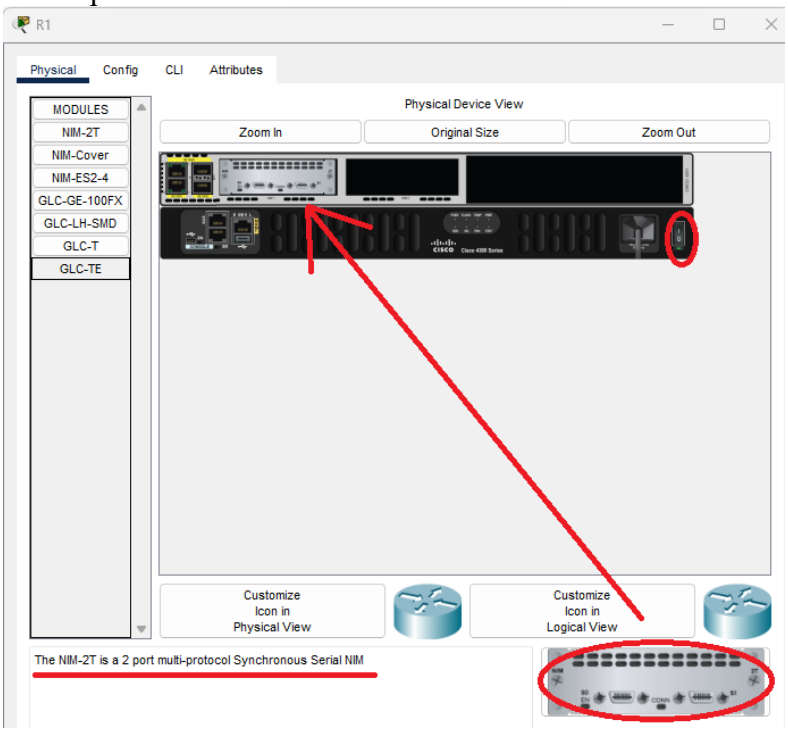


Рис. 3 Добавление модуля NM-2T в маршрутизатор ISR4331.

В качестве коммутаторов возьмите модель 2960. Среди конечных устройств выберите и добавьте в схему сети сервер и три настольных компьютера, как показано на рис. 2.

1. Настройка адресации

Табл. 1

Устройство	Интерфейс	IP-адрес	Маска подсети	Шлюз по умолчанию
RADIUS сервер	Fa0	192.168.10.1	255.255.255.0	192.168.10.254
SW-2	VLAN 1	192.168.20.1	255.255.255.0	192.168.20.254
SW-3	VLAN 1	192.168.30.1	255.255.255.0	192.168.30.254
R1	GigabitEthernet0/0/0	192.168.10.254	255.255.255.0	----------
R1	Serial0/1/0	10.0.0.1	255.255.255.252	----------
R2	Serial0/1/0	10.0.0.2	255.255.255.252	----------
R2	Serial0/1/1	11.0.0.1	255.255.255.252	----------
R2	GigabitEthernet0/0/0	192.168.20.254	255.255.255.0	----------
R3	Serial0/1/0	11.0.0.2	255.255.255.252	----------
R3	GigabitEthernet0/0/0	192.168.30.254	255.255.255.0	----------
PC1	Fa0	192.168.10.10	255.255.255.0	192.168.10.254
PC2	Fa0	192.168.20.10	255.255.255.0	192.168.20.254
PC3	Fa0	192.168.30.10	255.255.255.0	192.168.30.254

Настройте на интерфейсах устройств адреса согласно адресной таблице 1.

Настройка настольных компьютеров и сервера выполняется на вкладке Desktop в приложении «Настройка IP-конфигурации», рис. 4.

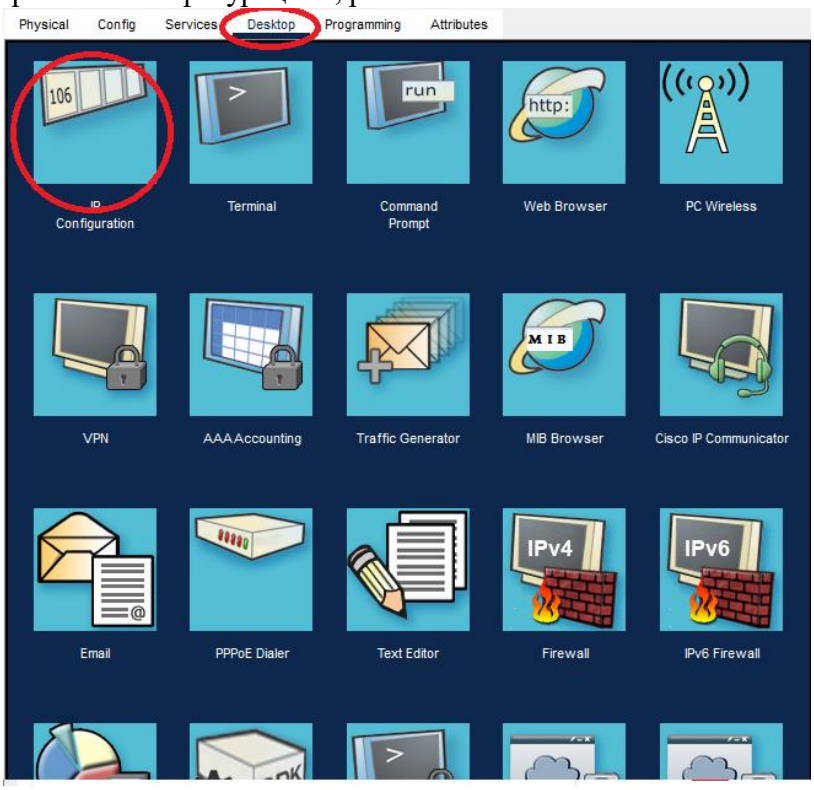


Рис. 4 Настройка IP-адреса, маски и адреса шлюза на компьютере или сервере

Для настройки адреса на интерфейсе VLAN 1 коммутатора необходимо перейти на вкладку интерфейса командной строки – CLI, рис. 5.

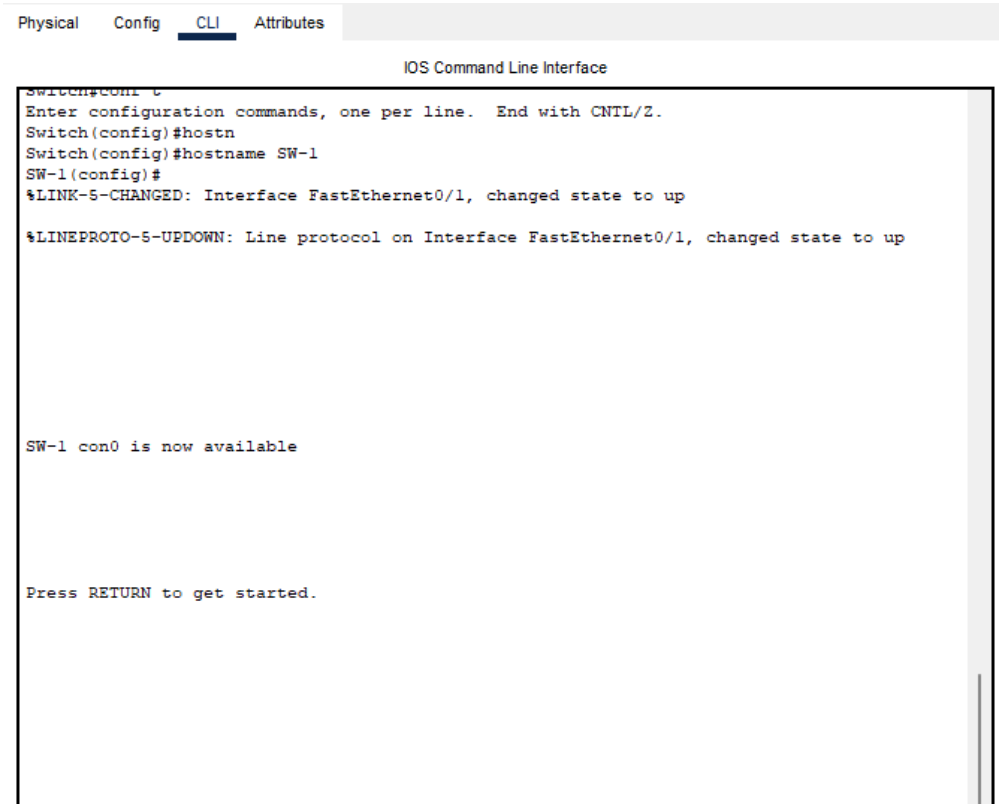


Рис. 5 Интерфейс командной строки на коммутаторе или маршрутизаторе в Cisco Packet Tracer

Нажать Enter для перемещения фокуса в окно командной строки. Ввести команду:
Switch0>**enable**

Для переключения в режим привилегированного доступа. Команду:
Switch0#**configure terminal**

Для переключения в режим глобальной конфигурации. Команду:
Switch0(config)#**interface VLAN 1**

Для переключения в режим настройки интерфейса VLAN 1, где командой:

Switch0(config-if)#**ip address A.B.A.B M.M.M.M**

Здесь A.B.A.B – IP-адрес в десятично-точечной нотации, M.M.M.M – маска в десятично-точечной нотации. Присвоить интерфейсу адрес и маску подсети. Включить интерфейс командой:
Switch0(config-if)#**no shutdown**

После чего выйти из режима настройки интерфейса в режим глобальной настройки командой:
Switch0(config-if)#**exit**

И настроить адрес шлюза по умолчанию командой:
Switch0(config)#**ip default-gateway A.B.A.B**

Изменить имя коммутатора можно в глобальной конфигурации командой:
Switch0(config)#**hostname SW-1**

Здесь SW-1 задаваемое новое имя. Можно использовать вкладку Config

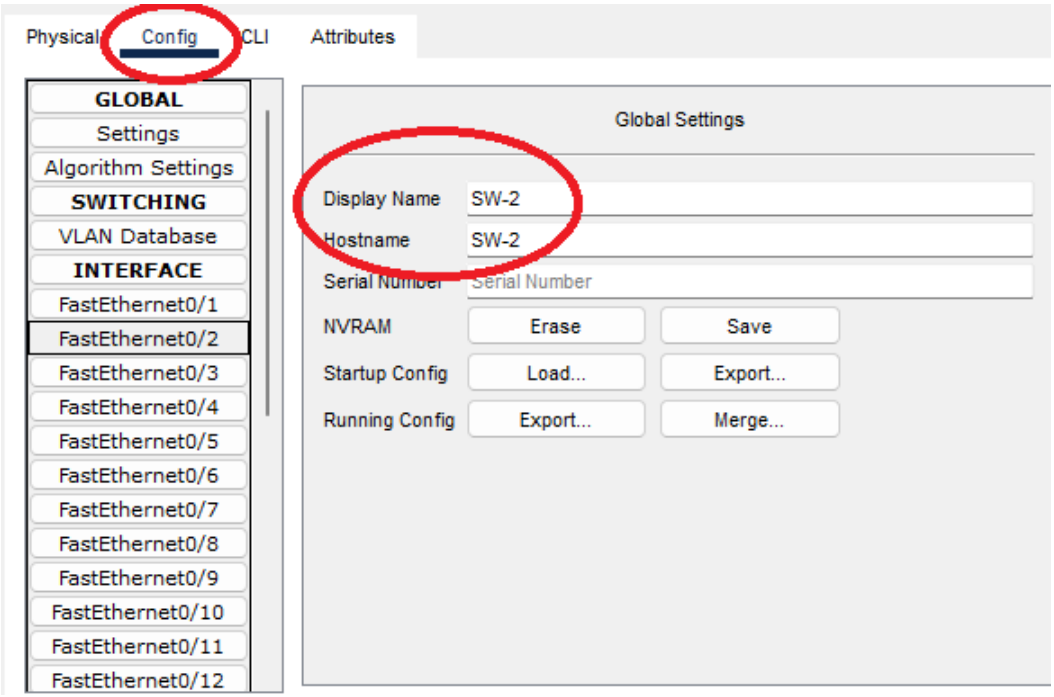


Рис. 6 Настройка имени устройства на вкладке Config

На маршрутизаторах для настройки адресов на интерфейсах можно использовать вкладку Config или командную строку.

2. Настройка маршрутизации

Для настройки маршрутизации на R1, R2 и R3 включите протокол динамической маршрутизации OSPF для одной области. Для этого перейдите в командную строку на маршрутизаторе, переключитесь в режим привилегированного доступа, а затем в режим глобальной конфигурации. В этом режиме введите команду:

Router0(config)#**router ospf 1**

Здесь цифра 1 идентифицирует процесс OSPF локально. Это может быть любое число в диапазоне 1 – 65535. Идентификаторы не обязательно должны совпадать на разных маршрутизаторах, но практически, для упрощения процедуры настройки, часто используют одни и те же идентификаторы. Команда переведет маршрутизатор в режим настройки OSPF. В этом режиме командой network нужно указать все сети, в которые включены интерфейсы маршрутизатора:

R1(config-router)#**network 192.168.10.0 0.0.0.255 area 0**

После указания адреса сети задается маска для этой сети в инвертированном виде, а затем указывается область, в которой работает OSPF. В нашей небольшой топологии будет только одна область с номером 0, которая считается в OSPF магистральной и обязательна к применению. Указав все сети для всех маршрутизаторов, из привилегированного режима проверьте таблицу маршрутизации командой:

R1#**show ip route**

В таблице должны присутствовать все сети, существующие в заданной топологии. Отправьте эхо-запрос с компьютера PC3 на RADIUS-сервер, запрос должен быть успешным.

3. Настройка RADIUS-сервера

Откройте окно управления сервером, перейдите на вкладку Services и выберите службу AAA, рис. 7:

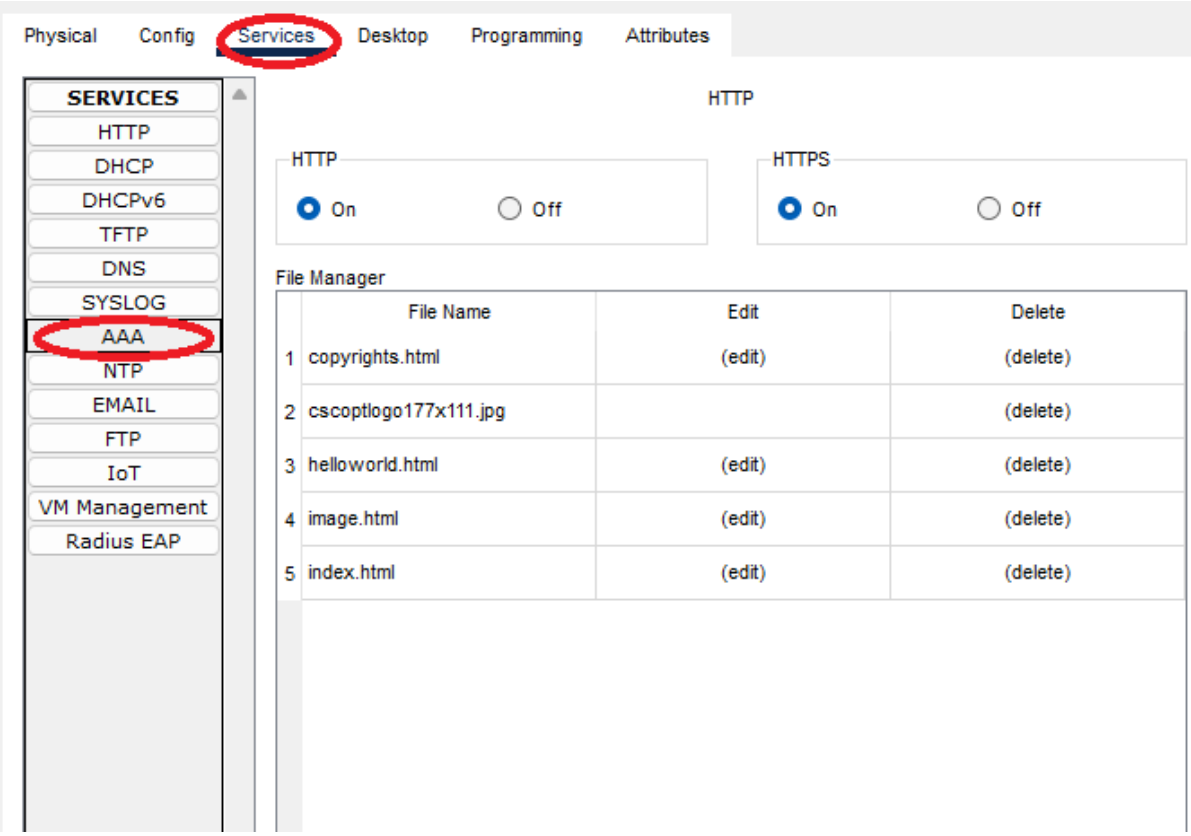


Рис. 7 Выбор службы AAA в перечне сервисов

1. Включите сервис радиокнопкой.
2. Задайте для коммутаторов SW-2 и SW-3, которые будут выступать в роли аутентификаторов, логин - имя устройства и пароль **tusur** на RADIUS сервере, а также укажите IP-адрес, настроенный на соответствующих интерфейсах VLAN 1. Тип протокола оставьте по умолчанию Radius, номер порта 1645 также не меняйте. Аутентификаторы лишь пересылают запросы от устройств, пытающихся войти в сеть, но сами при этом тоже проверяются в централизованном сервисе аутентификации.
3. Задайте логин и пароль для конечных устройств, в качестве которых будут выступать PC2 и PC3. На рис. 8 показан пример, для одного компьютера задан логин student и пароль PISH@123, для второго логин user и пароль TUSUR%23.

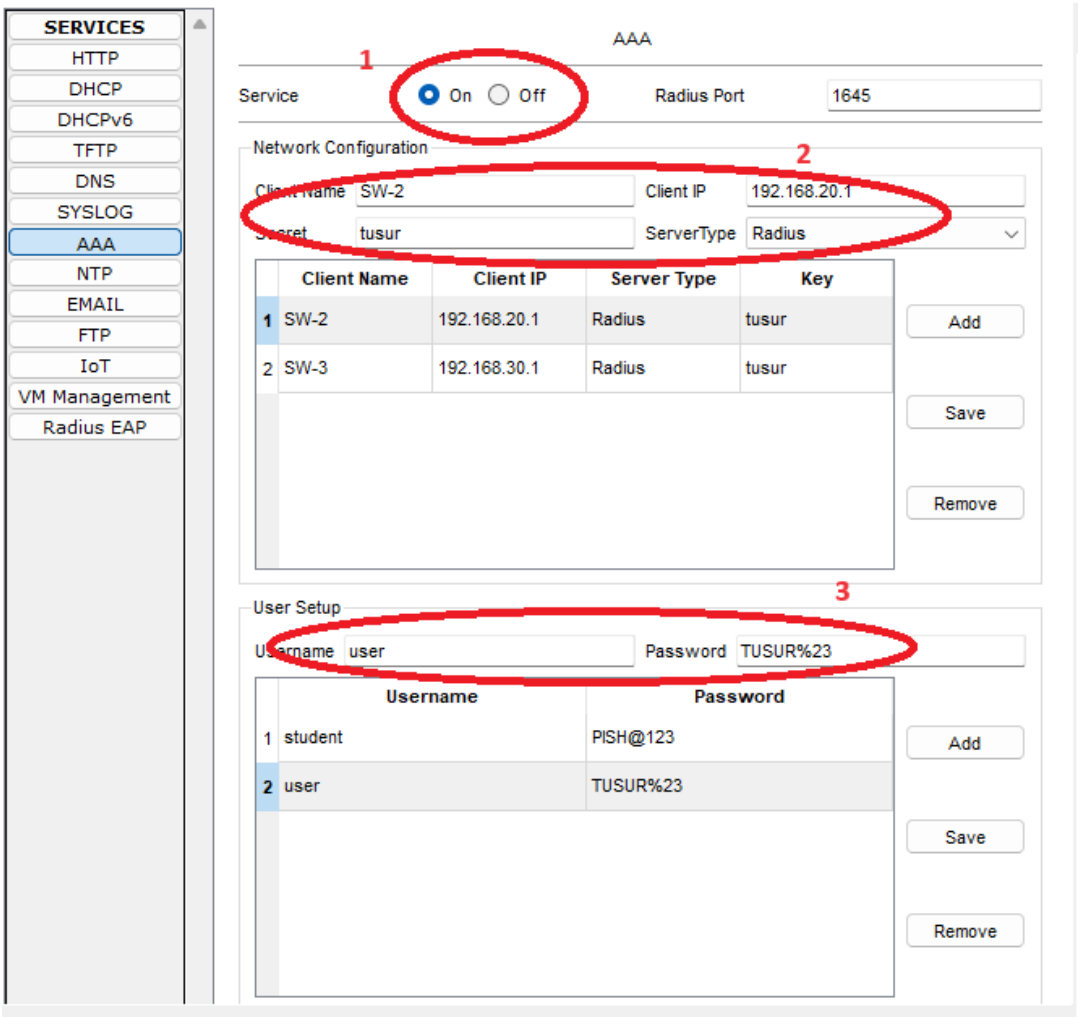


Рис. 8 Настройка параметров аутентификации 802.1X на RADIUS сервере.

Перейдите на пункт Radius EAP в списке сервисов и включите использование протокола EAP на RADIUS сервере, рис. 9:

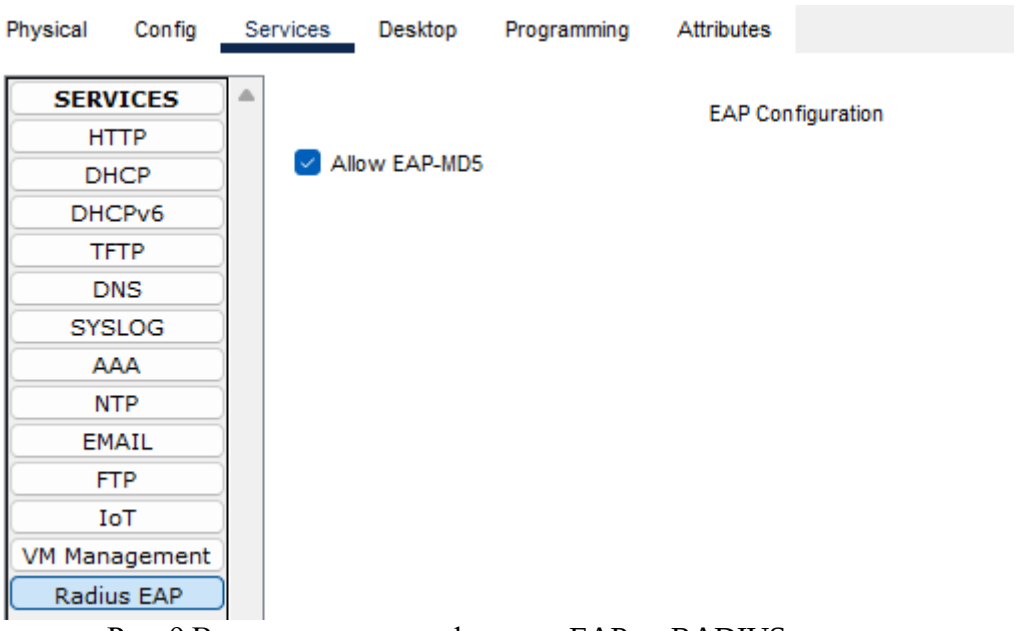


Рис. 9 Включение аутентификации EAP на RADIUS сервере

4. Настройка коммутаторов SW-2 и SW-3

В качестве аутентификаторов в нашей сети будут работать только два коммутатора: SW-2 и SW-3. Интерфейсы, на которых работает протокол IEEE802.1X, на Cisco коммутаторах необходимо перевести в режим доступа, т.к. по умолчанию они находятся в режиме dynamic auto – автоматической настройки режима транка и других параметров линии, которые согласуются с подключенным на другом конце кабеля партнером с помощью специального протокола. Работа этого протокола автоматического согласования параметров не совместима с работой протокола IEEE802.1X. Включение режима доступа на интерфейсе равносильно отключению протокола автоматического согласования. Проверьте, к какому интерфейсу коммутатора подключен компьютер. Перейдите в режим настройки этого интерфейса и выполните команду:

SW-2(config-if)#**switchport mode access**

Выйдите командой exit в режим глобальной конфигурации и включите на коммутаторе службу AAA командой:

SW-2(config)#**aaa new-model**

Настройте адрес RADIUS сервера и пароль для подключения к нему, как вы настроили его на Radius сервере:

SW-2(config)#**radius-server host 192.168.10.1 key tusur**

Укажите метод аутентификации для службы AAA – Radius сервер и протокол IEEE802.1X:

SW-2(config)#**aaa authentication dot1x default group radius**

Включите протокол IEEE802.1X на коммутаторе:

SW-2(config)#**dot1x system-auth-control**

Перейдите в режим настройки интерфейса, к которому подключен компьютер и включите аутентификацию по протоколу IEEE802.1X на этом интерфейсе:

SW-2(config-if)#**authentication port-control auto**

Задайте роль интерфейса коммутатора (Port Access Entity, PAE) в сценарии расширенной аутентификации:

SW-2(config-if)#**dot1x pae authenticator**

После выполнения этих настроек вы должны увидеть, что порты, к которым подключены PC2 и PC3 стали неактивными. Как отмечалось во вводной части, начальным состоянием порта является неавторизованное состояние, когда разрешено прохождение только кадров протокола IEEE802.1X. Никакой другой трафик не передается.

5. Настройка компьютеров PC2 и PC3

На компьютере откройте вкладку Desktop и приложение «Настройка IP-конфигурации». Активируйте раздел 802.1X и введите логин и пароль, установленный для соответствующего пользователя, рис. 10. Через некоторое время вы должны увидеть что интерфейс, которым компьютер подключен к коммутатору стал активным. Это значит, что была выполнена аутентификация. Для пользователя работа протокола прозрачна. Если настройкой интерфейса на его компьютере занимается администратор, то он может не знать, что его компьютер получает доступ в сеть только после прохождения проверки на сервере аутентификации.

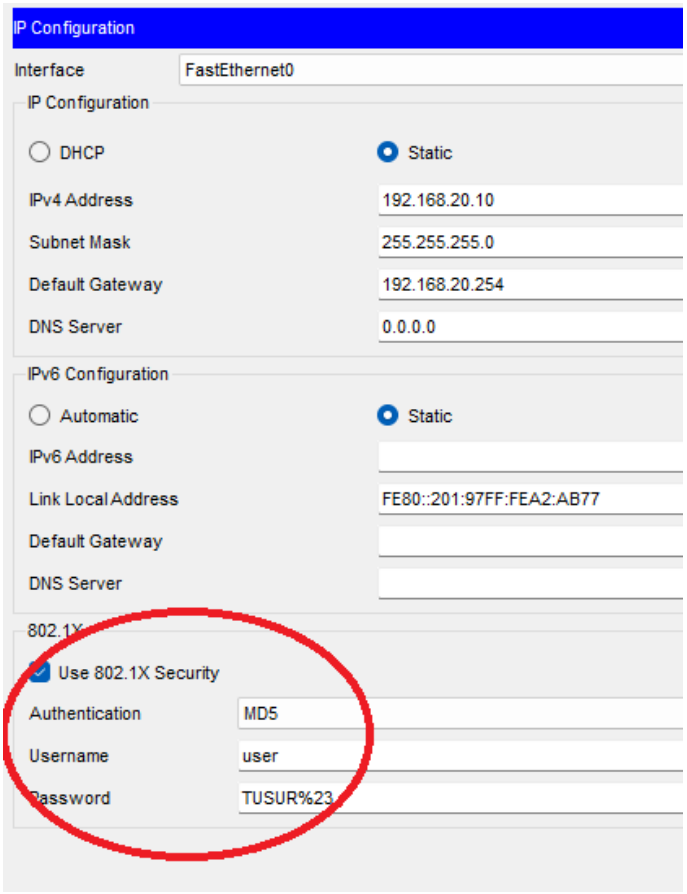


Рис. 10 Параметры аутентификации IEEE802.1X в настройках интерфейса компьютера

ПЕРЕЧЕНЬ ВОПРОСОВ ДЛЯ КОНТРОЛЯ

1. Перечислите функции каждого уровня модели OSI.
2. В модели сетевого взаимодействия TCP/IP какие протоколы соответствуют каким уровням модели.
3. Как соотносятся уровни модели TCP/IP и уровни модели OSI?
4. Какие адреса используются при сетевом взаимодействии?
5. Какие уровни регистрации событий предусмотрены службой Syslog?
6. Опишите принципы работы протокола NTP, какова иерархия серверов NTP, что значит запись в конфигурации сервера stratum 5?
7. Кодирование сигнала в компьютерной сети.
8. Связь полосы пропускания линии и метода кодирования.
9. Метод доступа к среде передачи Ethernet.
10. Halfduplex и Full-duplex.
11. MDI/MDIX.
12. Сетевой уровень.
13. Адресация в IP сетях.
14. Таблицы маршрутизации.
15. Алгоритм управления окном протокола TCP.
16. Работа системы доменных имен DNS.
17. Аутентификация клиента в беспроводной сети.

ЗАКЛЮЧЕНИЕ

Учебно-методическое пособие представляет собой ресурс, способствующий углубленному изучению основных аспектов построение инфокоммуникационных систем и сетей.

Цель данного практикума заключается в приобретении студентами навыков работы с сетевыми технологиями, построения компьютерных сетей различного уровня: локальных и глобальных компьютерных сетей. Практикум направлен на развитие практических навыков студентов в применении полученных теоретических знаний для решения задач, связанных с исследованиями современных инфокоммуникационных систем и сетей, а также расчета и проектирования компьютерных сетей.

В процессе изучения данного пособия студенты получат не только теоретические знания, но и практические навыки, которые пригодятся им в будущей профессиональной деятельности.

СПИСОК ИСТОЧНИКОВ

1. Ubuntu 20.04 https://github.com/mininet/mininet/releases/download/2.3.0/mininet-2.3.0-210211-ubuntu-20.04.1-legacy-server-amd64-ovf.zip. – Режим доступа: свободный (дата обращения 16.10.2023).
2. Внешний клиент MobaXterm. – URL: https://download.mobatek.net/2212022060563542/MobaXterm_Portable_v22.1.zip. – Режим доступа: свободный (дата обращения 16.10.2023).
3. Mikrotik. Образ маршрутизатора. – URL: https://download.mikrotik.com/routeros/6.49.10/chr-6.49.10.ova. – Режим доступа: свободный (дата обращения 16.10.2023).
4. Виртуальная машина маршрутизатора Mikrotik с RouterOS. – URL: https://download.mikrotik.com/routeros/6.48.6/chr-6.48.6.ova. – Режим доступа: свободный (дата обращения 16.10.2023)