ЛАБОРАТОРНЫЙ ПРАКТИКУМ

«Основы технологий сети Интернет»

Москва 2025

УДК 004.738.5(076.5)
ББК 16.26

Лабораторный практикум
Основы технологий сети Интернет

Авторы: В.А. Кирсанов, В.С. Малиночкин, С.А. Трусов, Е.С. Янин.

Лабораторный практикум по дисциплине «Основы технологий сети Интернет» предназначен для очного и дистанционного обучения студентов по программам бакалавриата, специалитета, магистратуры и дополнительного профессионального образования базовой кафедры общественно-государственного объединения «Ассоциация документальной электросвязи» «Технологии электронного обмена данными» в Московском техническом университете связи и информатики.

Рецензент: В.А. Ерёменко, к.т.н., доцент
ISBN 978-5-6054409-5-6

Содержание

Введение 5
Практикум № 1 Знакомство с терминалом и основными командами 6
Практикум № 2 Преобразование имен NSS 15
Практикум № 3 Знакомство с сетевыми интерфейсами 22
Практикум № 4 Настройка маршрутизации 28
Практикум № 5 Динамическая настройка сети 33
Практикум № 6 Организация сеанса 38
Практикум № 7 Преобразование имен на основе DNS 41
Практикум № 8 Передача нешифрованной и зашифрованной информации между компьютерами 51
Практикум № 9 Трансляция адресов (NAT) 58
Практикум № 10 Веб-сервер Apache 61
Практикум № 11 Использование пакетного менеджера RPM 75
Практикум № 12 Удалённый доступ по протоколу VNC 80
Практикум № 13 Статическая маршрутизация IPv4 83
Практикум № 14 Протокол динамической маршрутизации RIPv2 98
Практикум № 15 Протокол динамической маршрутизации OSPF 122
Практикум № 16 Протокол динамической маршрутизации BGP 150
Практикум № 17 Статическая маршрутизация IPv6 183
Практикум № 18 Протокол динамической маршрутизации OSPFv3 198
Практикум № 19 Настройка статической маршрутизации с использованием командной строки утилиты nmcli 228
Список литературы 232

Введение

Пособие для лабораторного практикума состоит из 19 практикумов по следующим направлениям:

- понимание реализации стека сетевых протоколов и принципов работы сети передачи данных на базе IP, организации адресации и маршрутизации, сетевых интерфейсов;
- понимание принципов построения и управления сетью Интернет, взаимодействия сетей связи и обмена трафиком;
- умение работать с сетевыми утилитами, анализаторами трафика и другими специальными инструментами.

Каждый практикум содержит: цель работы, задание на работу, термины и используемые команды, порядок выполнения работы.

По результатам выполнения лабораторной работы создается отчет, который оформляется в формате Open Doc, и сохраняется в заранее созданной со своим именем и группой папке на рабочем столе компьютера.

Отчет должен содержать следующий материал:

- титульный лист;
- цель работы;
- схему сети;
- скриншоты с результатами работы команд ОС Альт, анализатора протоколов и программ моделирования трафика;
- выводы по результатам выполнения лабораторной работы.

Авторы выражают благодарность за поддержку разработки настоящего лабораторного практикума Автономной некоммерческой организации «Координационный центр национального домена сети Интернет».

Практикум № 1
Знакомство с терминалом и основными командами

1. ЦЕЛЬ РАБОТЫ

Ознакомление с терминалом ОС Альт. Изучение основных команд терминала.

2. ЗАДАНИЕ НА РАБОТУ

1. Ознакомиться с терминалом ОС Альт;
2. Изучить основные команды терминала.

3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

3.1. Ознакомление с терминалом ОС Альт

Основная работа в курсе лабораторных будет проходить в терминале. Для его вызова необходимо нажать ПКМ на свободном от ярлыков, папок и панели задач месте на рабочем столе (рисунок 1). Далее выбрать «Открыть в Терминале», после чего будет открыт терминал (рисунок 2).

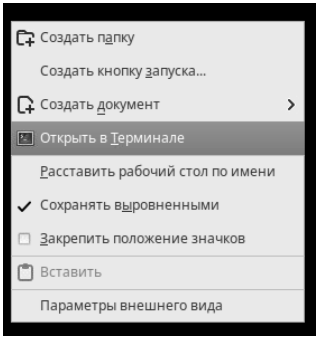


Рисунок 1. Диалоговое окно для открытия терминала

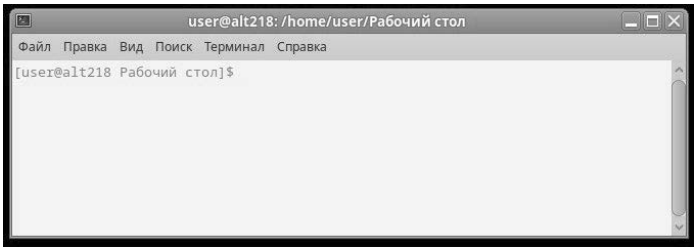


Рисунок 2. Открытый терминал

В терминале нас ожидает приглашение для ввода команд, которое изображено на рисунке 2.

Разберём из чего состоит приглашение:

`user` – имя пользователя, от имени которого происходит работа в терминале;

`@` - разделитель;

`alt218` – название машины;

`Рабочий стол` - текущая директория.

`$` - приглашение к выполнению команды от имени простого пользователя. Если пользователь имеет права администратора, то символ будет изменён на “#”, изображено на рисунке 3.

Для работы в терминале от имени администратора (суперпользователя) необходимо зайти под его именем, сменив пользователя. Для этого используется команда

```
su -
```

Далее будет предложено ввести пароль от профиля администратора (на всех компьютерах – 12345678). Процесс входа от имени суперпользователя изображён на рисунке 3.

```
[user@alt218 Рабочий стол]$ su -
Password:
[root@alt218 ~]#
```

Рисунок 3. Процесс перехода под суперпользователя

Как видно из рисунка 3, пользователь был изменён с `user` на `root`, а символ “$” на “#”. При этом домашняя директория так же была изменена с `/home/user/Рабочий стол` на `/root`.

Для выхода из режима суперпользователя или из-под любого другого пользователя, необходимо нажать `CTRL+D`.

3.2. Изучение основных команд

Все команды в терминале имеют общую структуру:

```
команда –ключ значение
```

`команда` – название исполняемого файла из каталогов, записанных в переменной $PATH (/bin, /sbin, /usr/bin, /usr/sbin, /usr/local/bin, /usr/local/sbin или полный путь к исполняемому файлу.

`ключ` – пишется после названия команды. У каждой команды свой набор ключей, подробнее о которых можно прочитать в инструкции к команде. С помощью ключей выставляются настройки, с использованием которых будет выполнена команда.

Ключи бывают короткие – состоящие из одного символа, например, -a, или же длинные, описывающие полное название ключа, например, -all. При этом ключ -a и -all, могут быть эквивалентными для какой-то команды, а для другой – нет. Короткие ключи можно перечислять подряд, например, `netstat -tpln`.

`Значение` – величина, которая будет передана при выполнении команды. Например, для команды `ping mtuci.ru` значением будет являться адрес `mtuci.ru`, который необходимо пропинговать.

Разберём основные команды, которые потребуются для работы в терминале:

`man <команда>` – показывает инструкцию по применению команды. Пример вызова команды изображён на рисунке 4, а полученная информация на рисунке 5. Для выхода из инструкции необходимо нажать клавишу `q`.

```
[user@alt218 Рабочий стол]$ man vim
```

Рисунок 4. Пример вызова команды `man`

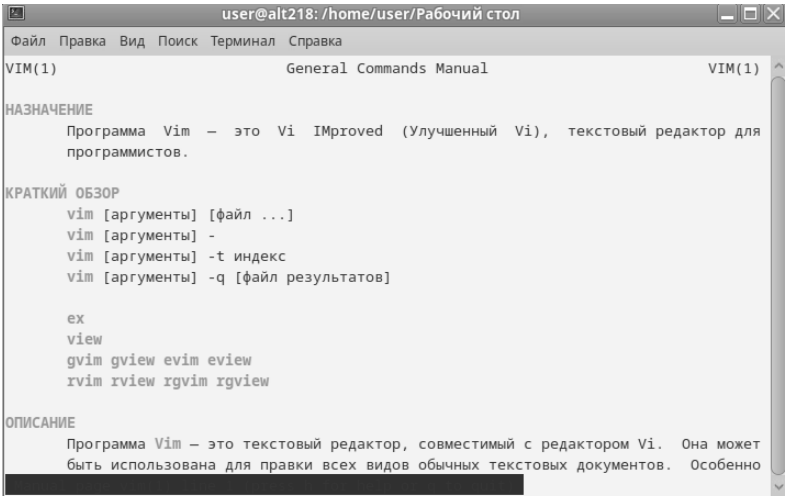


Рисунок 5. Пример полученной инструкции при вводе команды `man vim`

`cd <путь к директории>` – переход в указанную директорию. Без указания директории переходит в домашнюю директорию для пользователя, от которого осуществляется ввод команд.

Команда

```
cd .
```

Эквивалентна актуальному пути к каталогу, где находится пользователь. Вместо того, чтобы набирать весь путь к актуальной директории можно поставить символ "`.`" и писать путь после него. Например, в папке `/home/user` находится директория `.ssh`, перейти туда можно двумя способами (рисунок 6). Первый способ – полностью указать путь:

```
cd /home/user/.ssh
```

Второй способ:

```
cd ./.ssh
```

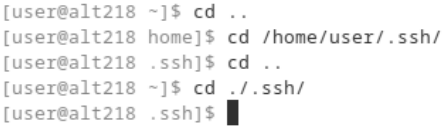


Рисунок 6. Два альтернативных способа вызова команды `cd`

При вводе команды

```
cd ..
```

Происходит переход в директорию, которая находится выше. Если до ввода команды активной директорией была `/home/user`, то после выполнения станет `/home`. Использование данной команды изображено на рисунке 7.

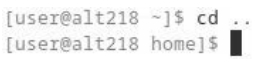


Рисунок 7. Использование команды `cd ..`, при нахождении в директории `/home/user`

`pwd` – выводит текущую директорию. Пример использования команды изображён на рисунке 8.

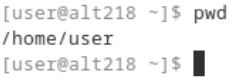


Рисунок 8. Пример использования команды `pwd`

`ls` – выводит список директорий и файлов в текущем каталоге. Информация, полученная при вводе команды `ls` для домашней директории пользователя изображена на рисунке 9.



Рисунок 9. Результат выполнения команды `ls`

`ip` – утилита, которая используется для работы с маршрутизацией, туннелями, устройствами. С помощью неё легко определить ip адрес устройства, для этого необходимо ввести команду

```
ip a
```

Результат выполнения команды изображён на рисунке 10.

```
[user@alt218 ~]$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp1s1f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state U
P group default qlen 1000
    link/ether 98:a7:b0:05:72:e0 brd ff:ff:ff:ff:ff:ff
    altname enx98a7b00572e0
    inet 10.10.21.218/24 brd 10.10.21.255 scope global noprefixroute enp1s1f0
       valid_lft forever preferred_lft forever
    inet6 fe80::9aa7:b0ff:fe05:72e0/64 scope link proto kernel_ll
       valid_lft forever preferred_lft forever
3: enp1s1f1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state
 DOWN group default qlen 1000
    link/ether 98:a7:b0:05:72:e1 brd ff:ff:ff:ff:ff:ff
    altname enx98a7b00572e1
5: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
```

Рисунок 10. Пример вызова команды `ip a`

Как видно из рисунка 10, у `enp1s1f0` есть ip адрес `10.10.21.218/24`

3.3. Использование WireShark

Для начала необходимо убедиться в том, что Wireshark установлен, для этого получим права суперпользователя, использовав команду

```
su -
```

Далее, введём команду

```
which wireshark
```

Если пакет не удалось обнаружить (рисунок 11), то установим его с помощью команды

```
apt-get install wireshark -y
```

```
[root@alt218 ~]# which wireshark
which: no wireshark in (/root/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/sbin:
/usr/local/bin:/usr/games)
[root@alt218 ~]#
```

Рисунок 11. Отсутствие Wireshark в системе

Для выбора сетевого интерфейса для анализа в WireShark, изначально, необходимо иметь права суперпользователя. Отключим это, введя команду

```
chmod u+s /usr/bin/dumpcap
```

Выйдем из режима суперпользователя, нажав `CTRL+D`.

Включение WireShark производится в консоли с помощью команды `wireshark`. При запуске потребуется выбрать сетевой интерфейс, с которого будет сниматься информация для последующего анализа. Для простоты можно выбрать `any`, что означает снятие со всех интерфейсов (рисунок 12).

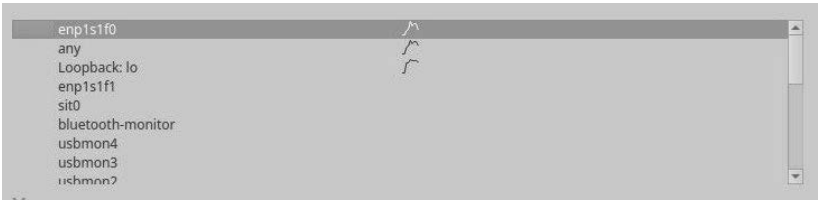


Рисунок 12. Список доступных интерфейсов

Для начала захвата пакетов воспользуемся кнопкой в виде синего плавника (рисунок 13) на панели инструментов. Для завершения захвата существует кнопка, смежная с плавником, в виде красного квадрата (рисунок 14).



Рисунок 13. Кнопка захвата пакетов



Рисунок 14. Кнопка остановки захвата пакетов

В основном поле программы (рисунок 15) находятся перехваченные пакеты, которые можно использовать для дальнейшего анализа.

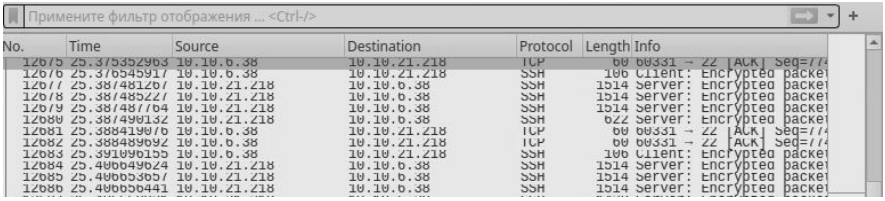


Рисунок 15. Фрагмент основного окна программы

Опишем основные элементы, отображённые на рисунке 19:

- `No.` – номер пакета;
- `Time` – время перехвата с момента включения режима перехвата пакетов;
- `Source` – источник пакета;
- `Destination` – назначение пакета;
- `Protocol` – протокол, по которому передаётся пакет;
- `Length` – длина пакета;
- `Info` – некоторая информация о пакете.

Отдельно следует упомянуть поле фильтрации, которое позволяет выводить пакеты, удовлетворяющие некоторым условиям.

При двойном клике на пакете откроется меню с его содержимым. Пример изображён на рисунке 16.

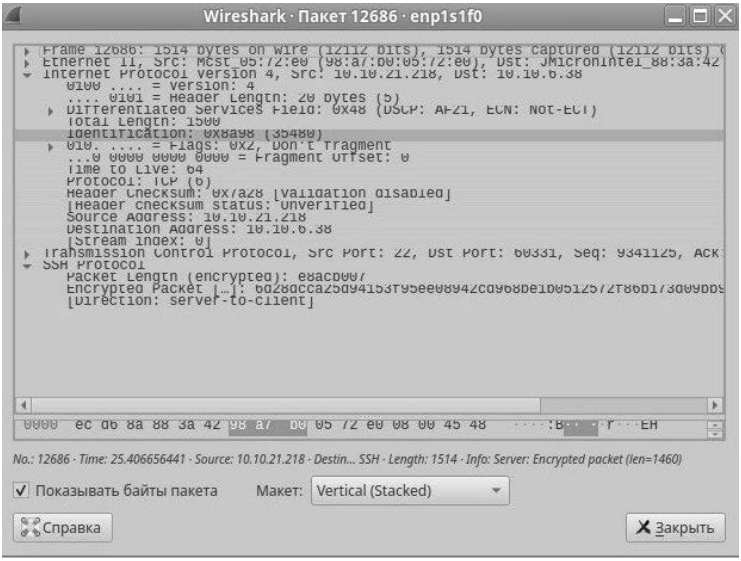


Рисунок 16. Пример содержания пакета

3.4. Использование утилиты iperf

Для генерирования трафика между двумя и более компьютерами можно использовать утилиту `iperf`. С её помощью можно легко проверить правильность настройки маршрутов между рабочими станциями, а также проводить анализ трафика. Ниже описано, как пользоваться утилитой.

Один из компьютеров должен выступать в качестве сервера. На нём необходимо выполнить команду (`-s` – сервер, `-p` – порт)

```
iperf -s -p port
```

Клиенту необходимо ввести

```
iperf -c ip -p port -n 20M
```

где

`ip` – ip-адрес сервера,

`n` – ограничение передачи данных,

`port` – порт сервера.

Если ip-адрес сервера – 10.10.21.219, а port, для примера, 12345, то получится следующее.

Для сервера

```
iperf -s -p 12345
```

Для клиента

```
iperf -c 10.10.21.219 -p 12345 -n 20M
```

Пример выполнения изображён на рисунках 17-18.

```
[user@alt219 ~]$ iperf -s -p 12345
------------------------------------
Server listening on TCP port 12345
TCP window size:  128 KByte (default)
------------------------------------
```

Рисунок 17. Активация iperf у сервера

```
[user@alt218 Рабочий стол]$ iperf -c 10.10.21.219 -p 12345 -n 20M
------------------------------------------------------------
Client connecting to 10.10.21.219, TCP port 12345
TCP window size: 85.0 KByte (default)
------------------------------------------------------------
[  3] local 10.10.21.218 port 52356 connected with 10.10.21.219 port 12345
```

Рисунок 18. Активация iperf у клиента

Практикум № 2
Преобразование имен NSS

1. ЦЕЛЬ РАБОТЫ

Ознакомиться с файлами, расположенными в каталоге `etc`: `nsswitch.conf`, `services`, `passwd`. Научиться создавать пользователей с паролями, определять их идентификатор, заходить в сеанс под этим пользователем и выполнять различные команды. Научиться создавать и удалять группы пользователей, а также определять идентификатор групп.

2. ЗАДАНИЕ НА РАБОТУ

1. Познакомьтесь с файлом `/etc/nsswitch.conf` и запишите основные базы данных для этого файла и источники информации для них.

2. Познакомьтесь с файлом `/etc/services` и заполните таблицу 1. соответствия служб и номеров портов (можно воспользоваться фильтром `grep`).

Таблица 1. Соответствие служб и номеров портов

Служба	Номер порта
	1
ftp	
ssh	
	23
	25
imap	
	631

3. Познакомьтесь с файлом `/etc/passwd` и заполните таблицу 2.

Таблица 2. Соответствие идентификаторов UID и имён пользователей

UID	Имя пользователя
0	
1	
8	
485	
500	

4. Создайте нового пользователя с именем `фамилия-группа`. Убедитесь в его присутствии в `/etc/passwd`. Задайте пароль для пользователя. Определите его UID.

Завершите сеанс суперпользователя и начните сеанс нового пользователя. Перейдите в его домашний каталог и создайте там файл test_file. Завершите сеанс пользователя.

5. Создайте новую группу `newgr`. Проверьте результат в `/etc/group`. Определите GID.

3. ТЕРМИНЫ И ИСПОЛЬЗУЕМЫЕ КОМАНДЫ

Термины

NSS (Name Service Switch – преобразование разрешенных имен) – установка, которая обеспечивает работу различных источников для общей конфигурации баз данных и механизмов разрешения имен. NSS позволяет системному администратору выбирать различные сервисы имен (поиск в текстовом файле, DNS, NIS, NIS+) для поиска в базе данных доменных имен, имен пользователей, имен групп пользователей и т.д.

Используемые команды

`cat` – команда для работы с файлами.

`grep` – команда для фильтрации информации из файла. Применяется как обёртка для других команд, например, для команды `cat`.

`useradd` – команда для создания нового пользователя.

`passwd` – команда для управления паролями пользователей.

`groupadd` – команда для создания новой группы.

`nano` – текстовый редактор.

`getent база элемент` – получает указанные элементы из указанной базы данных.

4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

4.1. Файл `nsswitch.conf`

Откроем файл, находящийся в каталоге `/etc/nsswitch.conf`, с помощью команды `cat` и посмотрим основные базы данных и источники информации для них.

Например, файл `passwd` является базой данных для файла `nsswitch.conf`. Источником информации для `passwd` являются локальные файлы. В таком же формате приведена информация для других баз данных (рисунок 1).

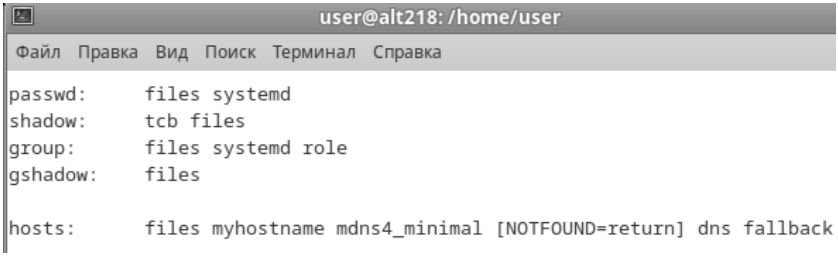


Рисунок 1. Файл `nsswitch.conf`

4.2. Файл `services`

Откроем файл `/etc/services` командой `cat` и ознакомимся с ним.

Узнаем из этого файла названия служб и портов, которые эти службы используют. Для этого воспользуемся фильтром `grep`.

```
cat /etc/services | grep 1
```

Результат выполнения команды изображён на рисунке 2. Определим службу, которая имеет порт с номером 1. В соответствии с командой выше, изменяя значение параметра после команды `grep`, необходимо получить службы и номера портов служб для заполнения в таблицу 1.

```
[user@alt218 ~]$ cat /etc/services | grep 1
# The Well Known Ports are those from 0 through 1023.
# The Registered Ports are those from 1024 through 49151.
# The Dynamic and/or Private Ports are those from 49152 through 65535.
tcpmux          1/tcp                           # TCP port service multiplexer
tcpmux          1/udp                           # TCP port service multiplexer
systat          11/tcp          users           # Active Users
systat          11/udp          users           # Active Users
daytime         13/tcp                          # Daytime (RFC 867)
daytime         13/udp                          # Daytime (RFC 867)
```

Рисунок 2. Использование `grep` для фильтрации информации из файла

Для поиска данных из файлов, которые описаны в nsswitch.conf, также существует специальная команда `getent`. С помощью данной команды, выполним поиск номера порта для службы ftp и название службы, которая использует порт 1.

```
getent services ftp
getent services 1
```

Результат выполнения представлен на рисунке 3.

```
[user@alt218 ~]$ getent services ftp
ftp                  21/tcp fsp fspd
[user@alt218 ~]$ getent services 1
tcpmux               1/tcp
```

Рисунок 3. Пример использования команды `getent`

4.3. Файл passwd

Откроем файл `/etc/passwd`, используя команду:

```
cat /etc/passwd
```

Используя команду `getent` (как в пункте 4.2), необходимо заполнить таблицу 2. Пример использования команды с фильтром изображен на рисунке 4.

```
[user@alt218 ~]$ getent passwd 0
root:x:0:0:System Administrator:/root:/bin/bash
```

Рисунок 4. Фрагмент файла `passwd`

Разберём одну строку для примера

```
root:x:0:0:System Administrator:/root:/bin/bash
```

Все поля разделены с помощью двоеточий и имеют следующий порядок:

- Имя пользователя (`root`);
- Необязательный зашифрованный пароль (`x`);
- Идентификатор пользователя UID (`0`);
- Идентификатор группы GID (`0`);
- Комментарий (`System Administrator`);
- Домашний каталог (`/root`);
- Используемая оболочка (`/bin/bash`).

Если в поле оболочки указан исполняемый файл, отличный от командного интерпретатора (например, `/sbin/nologin`, `/dev/null` или `/bin/false`), интерактивный вход пользователя в систему запрещён.

4.4. Создание нового пользователя

Для добавления нового пользователя, вам потребуются права суперпользователя. Для этого вводим команду

```
su -
```

Создать нового пользователя с именем `фамилия-группа` можно, набрав команду

```
useradd ivanov-btv2103
```

Переход под суперпользователя и создание нового пользователя изображены на рисунке 5.

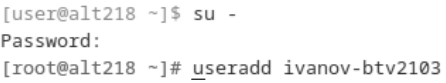


Рисунок 5. Создание нового пользователя

Убедимся, что он присутствует в `/etc/passwd`, набрав команду

```
cat /etc/passwd
```

Новый пользователь находится в конце файла (рисунок 6).

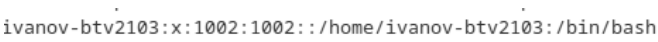


Рисунок 6. UID нового пользователя

Проверить можно и не открывая файл, для этого используем команду

```
getent passwd ivanov-btv2103
```

Команда `getent` будет искать пользователя `ivanov-btv2103` в базе данных `passwd`. Если он обнаружен, то будет выведен идентификатор (UID) этого пользователя.

Для задания или смены пароля существует команда

```
passwd пользователь
```

В нашей случае, имя пользователя – `ivanov-btv2103`. Команда будет иметь вид

```
passwd ivanov-btv2103
```

После ввода команды потребуется дважды ввести пароль (рисунок 7).

```
[root@alt218 ~]# passwd ivanov-btv2103
passwd: updating all authentication tokens for user ivanov-btv2103.

You can now choose the new password or passphrase.

A valid password should be a mix of upper and lower case letters, digits, and
other characters.  You can use a password containing at least 4 characters
from at least 3 of these 4 classes.
An upper case letter that begins the password and a digit that ends it do not
count towards the number of character classes used.

A passphrase should be of at least 3 words, 6 to 72 characters long, and
contain enough different characters.

Alternatively, if no one else can see your terminal now, you can pick this as
your password: "rat$meadow3riot".

Enter new password:
Weak password: based on a common sequence of characters and not a passphrase.
Re-type new password:
passwd: all authentication tokens updated successfully.
```

Рисунок 7. Создание пароля для нового пользователя

Обратим внимание, что при использовании команды `passwd` терминал выводит информацию о пожеланиях к паролю: пароль должен состоять не менее чем из трёх слов и содержать от 6 до 40 различных символов.

Выйдем из-под суперпользователя, нажав комбинацию клавиш `CTRL+D`, а затем войдём под именем пользователя `ivanov-btv2103`, введя команду (рисунок 8)

```
su – ivanov-btv2103
```

```
[user@alt218 Рабочий стол]$ su - ivanov-btv2103
Password:
[ivanov-btv2103@alt218 ~]$
```

Рисунок 8. Переход под нового пользователя

Зайдя под пользователем ivanov-btv2103, мы увидим, что текущая директория была изменена на домашнюю директорию пользователя. Для проверки введём команду

```
pwd
```

Теперь создадим файл test_file, используя текстовый редактор nano

```
nano test_file
```

Запишем туда какой-либо текст и, затем, сохраним его, нажав на CTRL+X, после y и далее enter. Чтобы удостовериться в том, что файл был создан, введён команду ls (рисунок 9).

```
[ivanov-btv2103@alt218 ~]$ pwd
/home/ivanov-btv2103
[ivanov-btv2103@alt218 ~]$ nano test_file
[ivanov-btv2103@alt218 ~]$ ls
test_file
[ivanov-btv2103@alt218 ~]$
```

Рисунок 9. Создание нового файла

4.5. Создание новой группы

Перейдём под суперпользователя, для чего в начале выйдем с пользователя ivanov-btv2103, нажав CTRL+D, а затем введём

```
su -
```

Для создания новых групп существует команда groupadd. С её помощью создадим группу группа-n, где n – порядковый номер в группе.

```
groupadd btv2103-5
```

Получим идентификатор группы, введя команду

```
cat /etc/group
```

Новая группа будет находиться в конце файла.

Практикум № 3
Знакомство с сетевыми интерфейсами

1. ЦЕЛЬ РАБОТЫ

Познакомиться с сетевыми интерфейсами и некоторыми командами для совершения операций над ними. Изучить возможности утилит netstat и ss.

2. ЗАДАНИЕ НА РАБОТУ

1. Узнать имена сетевых интерфейсов в системе и посчитать их количество.
2. Вывести локальный сетевой интерфейс, используя утилиту `grep`.
3. Включить и выключить сетевой интерфейс `enp1s1f0`. Убедиться, что ваши действия отображаются в состоянии интерфейса (UP или DOWN).
4. Запустить `ping` локального адреса. Перевести этот процесс в фоновый режим. Убедиться с помощью команды `pgrep` (или `ps`) в том, что действие выполняется. Завершить процесс командой `kill`.
5. С помощью команды `netstat` узнать список соединений типа TCP и приложений, связанных с ними.
6. Выполнить предыдущее задание с помощью команды `ss`.

3. ТЕРМИНЫ И ИСПОЛЬЗУЕМЫЕ КОМАНДЫ

Термины

Сетевой интерфейс – точка соединения между компьютером пользователя и частной или общественной сетью. Существуют внешние и внутренние (локальные) интерфейсы.

Локальный сетевой интерфейс – это локальная петля, (петля говоря простым языком – это когда посланный сигнал отправляется и возвращается обратно, то есть происходит движение по кругу) которая имеет IP-адрес 127.0.0.1 и предназначена для сетевого доступа к своему же компьютеру.

Фоновый режим – это режим, при котором определенные сервисы и программы работают автоматически, без участия человека.

Используемые команды

`ip` – команда для работы с сетевыми интерфейсами.

`ip link` – команда для управления состоянием сетевых интерфейсов и получения информации о них.
`ping` – команда для отправки пакетов по указанному сетевому адресу.
`grep` – команда для фильтрации информации из файла. Часто применяется как обёртка для других команд.
`pgrep` – команда, позволяющая просматривать активные процессы в системе и выдающая идентификаторы процессов, атрибуты которых соответствуют указанным в командной строке запросам.
`kill` – закрытие (“убийство”) процесса.
`ps` – тоже, что и `pgrep`.
`netstat` – выводит на дисплей состояние TCP-соединений (как входящих, так и исходящих), таблицы маршрутизации, число сетевых интерфейсов и сетевую статистику по протоколам. В настоящее время команда устаревшая и не рекомендована к применению.
`ss` – современный аналог команды `netstat`.

4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

4.1. Определение имён сетевых интерфейсов

Чтобы узнать имена сетевых интерфейсов в системе, воспользуемся утилитой `ip`, используя параметр `link` (рисунок 1).

```
[user@alt218 Рабочий стол]$ ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
 group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp1s1f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state U
P mode DEFAULT group default qlen 1000
    link/ether 98:a7:b0:05:72:e0 brd ff:ff:ff:ff:ff:ff
    altname enx98a7b00572e0
3: enp1s1f1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state
 DOWN mode DEFAULT group default qlen 1000
    link/ether 98:a7:b0:05:72:e1 brd ff:ff:ff:ff:ff:ff
    altname enx98a7b00572e1
5: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN mode DEFAULT group default
qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
```

Рисунок 1. Пример использования команды `ip link`

Имена сетевых интерфейсов находятся после цифр 1, 2, … и т.д. Чтобы посчитать количество интерфейсов, воспользуемся командой:

```
ip -o link | wc -l
```

Параметр `-o` (буква) в команде `ip link` выводит каждую запись на отдельной строке. Для подсчёта количества строк воспользуемся командой для подсчёта строк `wc -l` (буква).

4.2. Получение информации о локальном интерфейсе

Для вывода информации о локальном интерфейсе воспользуемся утилитой `grep`, которой обернём команду `ip link` (рисунок 2).

```
ip link | grep lo
```

```
[user@alt218 Рабочий стол]$ ip link | grep lo
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
```

Рисунок 2. Полученная информация о локальном интерфейсе

4.3. Работа с сетевым интерфейсом enp1s1f0

Для работы с сетевым интерфейсом необходимо войти в режим суперпользователя (пароль – 12345678)

```
su -
```

Для выключения сетевого интерфейса используем команду (рисунок 3)

```
ip link set enp1s1f0 down
```

Команда `ip link set` изменяет параметры указанного сетевого интерфейса.

```
[root@alt218 ~]# ip link set enp1s1f0 down
```

Рисунок 3. Выключение сетевого интерфейса enp1s1f0

Проверим, что интерфейс выключен, введя команду `ip a` (рисунок 3). Как видно из рисунка, сетевой интерфейс `enp1s1f0` перешёл в состояние DOWN. Значит, он выключен.

```
[root@alt218 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp1s1f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state DOWN group default qlen 1000
    link/ether 98:a7:b0:05:72:e0 brd ff:ff:ff:ff:ff:ff
    altname enx98a7b00572e0
    inet 10.10.21.218/24 brd 10.10.21.255 scope global noprefixroute enp1s1f0
       valid_lft forever preferred_lft forever
    inet6 fe80::9aa7:b0ff:fe05:72e0/64 scope link proto kernel_ll
       valid_lft forever preferred_lft forever
3: enp1s1f1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN group default qlen 1000
    link/ether 98:a7:b0:05:72:e1 brd ff:ff:ff:ff:ff:ff
    altname enx98a7b00572e1
5: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
```

Рисунок 3. Пример выполнения команды `ip a`

Теперь включим его обратно, воспользовавшись командой

```
ip link set enp1s1f0 up
```

Результат выполнения команды представлен на рисунке 4. Если на интерфейсе есть ip адрес, то будет написано UP, иначе UNKNOWN.

```
[root@alt218 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp1s1f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 98:a7:b0:05:72:e0 brd ff:ff:ff:ff:ff:ff
    altname enx98a7b00572e0
    inet 10.10.21.218/24 brd 10.10.21.255 scope global noprefixroute enp1s1f0
       valid_lft forever preferred_lft forever
    inet6 fe80::9aa7:b0ff:fe05:72e0/64 scope link proto kernel_ll
       valid_lft forever preferred_lft forever
3: enp1s1f1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN group default qlen 1000
    link/ether 98:a7:b0:05:72:e1 brd ff:ff:ff:ff:ff:ff
    altname enx98a7b00572e1
5: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
```

Рисунок 4. Включение сетевого интерфейса

4.4. Использование утилиты ping

Выйдем из-под суперпользователя, нажав `CTRL+D`. Воспользуемся утилитой `ping` и укажем ip адрес сетевого интерфейса, а затем переведём её в фоновый режим, нажав на `CTRL+Z` (рисунок 5).

```
ping 127.0.0.1
CTRL+Z
```

```
[user@alt218 Рабочий стол]$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.034 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.039 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.052 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.076 ms
^Z
[1]+  Остановлен    ping 127.0.0.1
```

Рисунок 5. Перевод процесса `ping` в фоновый режим

Используем команду `ps`, чтобы обнаружить процесс ping в фоновом режиме, а также для определения его идентификатора PID (рисунок 6).

```
[user@alt218 Рабочий стол]$ ps
    PID TTY          TIME CMD
2355947 pts/0    00:00:00 bash
2355964 pts/0    00:00:00 ping
2356277 pts/0    00:00:00 ps
```

Рисунок 6. Список процессов

Завершим процесс `ping` командой `kill -9` (-9 – номер сигнала для убийства процесса, который означает безусловное завершение без возникновения ошибок), введя команду:

```
kill -9 2355964
```

2355964 – номер процесса команда `ping`, который был получен в результате использования команды `ps` (поле PID).

Результат выполнения команды `kill` изображён на рисунке 7.

```
[user@alt218 Рабочий стол]$ kill -9 2355964
[user@alt218 Рабочий стол]$ [1]+  Убито              ping 127.0.0.1
```

Рисунок 7. Завершение процесса `ping`

4.5. Получение списка TCP соединений

Определим список всех слушающих сокетов типа TCP с помощью утилиты `netstat`

```
netstat -lt
```

Распишем ключи, использованные в команде:

`-l` – показывать только слушающие сокеты;

`-t` – отображать только TCP соединений.

Результат выполнения команды изображён на рисунке 8.

```
[user@alt218 Рабочий стол]$ netstat -lt
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address               Foreign Address             State
tcp        0      0 *:ssh                       *:*                         LISTEN
tcp        0      0 *:3389                      *:*                         LISTEN
tcp        0      0 localhost.localdomai:sunrpc *:*                         LISTEN
tcp        0      0 localhost.localdomain:ipp   *:*                         LISTEN
tcp        0      0 localhost.localdomain:39303 *:*                         LISTEN
tcp        0      0 localhost.localdomain:39302 *:*                         LISTEN
tcp        0      0 *:ssh                       *:*                         LISTEN
tcp        0      0 localhost6.localdoma:sunrpc *:*                         LISTEN
tcp        0      0 localhost6.localdomain:ipp  *:*                         LISTEN
tcp        0      0 localhost6.localdomai:39302 *:*                         LISTEN
```

Рисунок 8. Список TCP соединений, полученных с помощью утилиты `netstat`

4.6. Использование утилиты ss

Пункт 4.5. можно выполнить с помощью современной утилиты ss. Её синтаксис и флаги аналогичны команде `netstat`:

```
ss -lt
```

Результат выполнения представлен на рисунке 9.

```
[user@alt218 Рабочий стол]$ ss -lt
State      Recv-Q      Send-Q          Local Address:Port           Peer Address:Port
LISTEN     0           128                   0.0.0.0:ssh                 0.0.0.0:*
LISTEN     0           2                     0.0.0.0:3389                0.0.0.0:*
LISTEN     0           4096              127.0.0.1:sunrpc                0.0.0.0:*
LISTEN     0           4096              127.0.0.1:ipp                   0.0.0.0:*
LISTEN     0           128               127.0.0.1:39303                 0.0.0.0:*
LISTEN     0           128               127.0.0.1:39302                 0.0.0.0:*
LISTEN     0           128                    [::]:ssh                    [::]:*
LISTEN     0           4096                  [::1]:sunrpc                 [::]:*
LISTEN     0           4096                  [::1]:ipp                    [::]:*
LISTEN     0           128                   [::1]:39302                  [::]:*
LISTEN     0           128                   [::1]:39303                  [::]:*
```

Рисунок 9. Список TCP соединений, полученных с помощью утилиты `ss`

Практикум № 4
Настройка маршрутизации

1. ЦЕЛЬ РАБОТЫ

Познакомиться с понятиями публичных и частных сетей. Изучить механизмы функционирования маршрутизации в ОС Альт. Научиться настраивать статическую маршрутизацию.

2. ЗАДАНИЕ НА РАБОТУ

Компьютеры соединены по схеме, изображённой на рисунке 1.

Наименование интерфейсов указано для примера, на практике оно может отличаться. В нашем же случае `eth0 – enp1s1f0`, а `eth1 – enp1s1f1`.

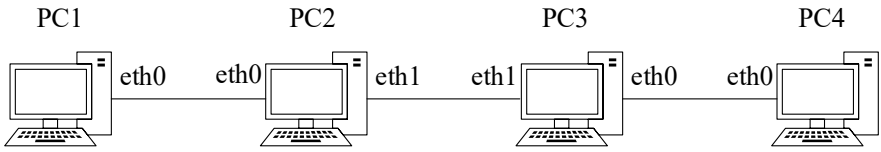


Рисунок 1. Схема соединения компьютеров

1. Настройка сетевых интерфейсов
   - Присвоение ip адресов;
   - Настройка конфигурационных файлов;
   - Разрешение передачи пакетов между сетевыми интерфейсами;
   - Настройка сетевых маршрутов.
2. Проверка правильности выполнения работы

3. ТЕРМИНЫ И ИСПОЛЬЗУЕМЫЕ КОМАНДЫ

Термины

Для большинства компьютеров локальным адресом считается адрес, расположенный в диапазоне 127.0.0.0, …,127.255.255.25, именно поэтому ip адреса, расположенные в данном диапазоне, нельзя использовать для узла локальной сети.

Использование локального адреса позволяет устанавливать соединение и передавать информацию для программ-серверов, работающих на том же компьютере, что и программа-клиент, независимо от конфигурации аппаратных сетевых средств компьютера.

Публичные IP-адреса – это те адреса, которые уникальны в пределах всего мира, иногда их еще называют белыми адресами. Можно было бы сказать, что публичные адреса – это все те адреса, которые не обозначены как частные, но это далеко не так, поскольку в протоколе IP есть еще целый ряд сетей, у которых особое назначение, и большая часть из них не находится в Интернете.

Частные IP-адреса – это такие адреса, которые не уникальны в пределах всего мира, но они должны быть уникальны в пределах локальной сети.

Используемые команды

`nano` – текстовый редактор.

`ip a` – команда для настройки сетевых интерфейсов.

`ip r` – команда для настройки маршрутизации.

`iperf` – утилита для генерации трафика.

4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

4.1. Настройка сетевых интерфейсов

Для выполнения всех команд необходимо зайти в режим суперпользователя (пароль – 12345678)

```
su -
```

Далее, в зависимости от того, какой сетевой интерфейс указан в задании и за каким компьютером находится студент, создадим файл `ipv4address` в директории `/etc/net/ifaces/enp1s1fX`, X – номер сетевого интерфейса, согласно рисунку 1 и номеру компьютера студента. Для примера возьмём интерфейс `enp1s1f0`, тогда команда будет иметь вид:

```
nano /etc/net/ifaces/enp1s1f0/ipv4address
```

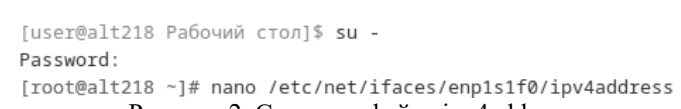


Рисунок 2. Создание файла ipv4address

В файле необходимо записать ip адрес. Определим, какой адрес необходимо указать для каждой из машин. На всех рабочих местах указан номер. Согласно этому номеру выбирается подсеть по следующему правилу – подсеть для двух компьютеров выбирается в соответствии с минимальным номером рабочего места. Общая формула выбора

```
10.10.XY.Z/24
```

где

`X` – номер ряда с рабочими местами;

`Y` – минимальный номер между двумя рабочими местами;

`Z` – номер рабочего места.

Предположим, что мы проводим настройку 1 ряда. Пусть схема подключения выглядит следующим образом

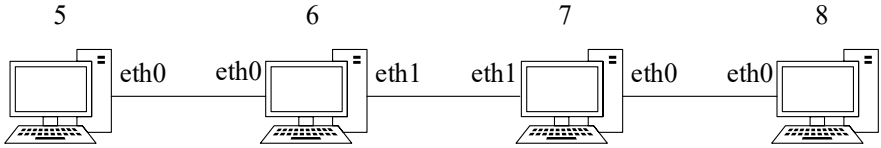


Рисунок 3. Пример схемы с указанием нумерации компьютеров

Согласно рисунку 3 и описанному выше правилу, компьютеры получат следующие ip адреса:

5:

- `10.10.15.5/24` на интерфейсе `enp1s1f0`

6:

- `10.10.15.6/24` на интерфейсе `enp1s1f0`
- `10.10.16.6/24` на интерфейсе `enp1s1f1`

7:

- `10.10.16.7/24` на интерфейсе `enp1s1f1`
- `10.10.17.7/24` на интерфейсе `enp1s1f0`

8:

- `10.10.17.8/24` на интерфейсе `enp1s1f0`

Далее, после записи адреса в файл, сохраним введённые данные с помощью комбинации `CTRL+O`, после чего нажмём `enter`, и выйдем из редактирования – `CTRL+X`.

Произведём настройку сетевых интерфейсов, для этого откроем файл `options` в директории `/etc/net/ifaces/enp1s1fX`, `X` - номер сетевого интерфейса, согласно рисунку 1 и номеру компьютера студента. К примеру, для интерфейса `enp1s1f0` команда будет выглядеть следующим образом

```
nano /etc/net/ifaces/enp1s1f0/options
```

В указанном файле обязательно должны присутствовать следующие строки с приведёнными ниже значениями:

`NM_CONTROLLED=no` – интерфейс не управляется NetworkManager

`DISABLED=no` – указывает, включён или отключён сетевой интерфейс

Также в файле могут быть указаны и другие параметры, если они отсутствуют, то прописывать их не требуется:

`BOOTPROTO=static` – способ получения IP-адреса

`ONBOOT=yes` – интерфейс поднимается автоматически при загрузке системы

`TYPE=eth` – тип сетевого интерфейса

`CONFIG_WIRELESS=no` – интерфейс не является беспроводным

`CONFIG_IPV4=yes` – используется протокол IPv4

На компьютерах PC2 (в примере – компьютер 6) и PC3 (в примере – компьютер 7) включим пересылку пакетов между сетевыми интерфейсами, изменив в файле `sysctl.conf` параметр `net.ipv4.ip_forward` на 1. Для этого откроем файл, введя команду

```
nano /etc/net/sysctl.conf
```

И изменим значение параметра `net.ipv4.ip_forward` на 1, если значение уже не равно единице (рисунок 4).

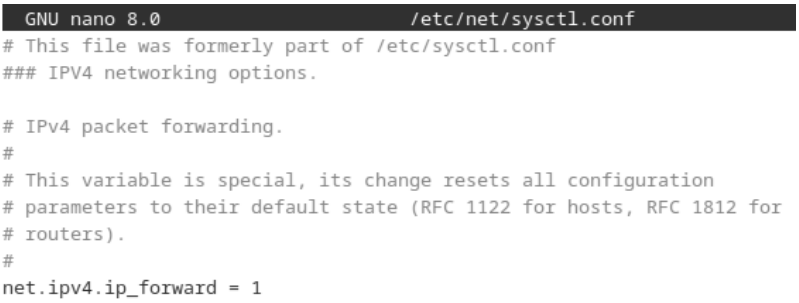


Рисунок 4. Фрагмент файла `sysctl.conf`

Перезапустим сетевую службу на всех компьютерах для применения параметров

```
service network restart
```

Пропишем следующие маршруты в консоли для каждого из компьютеров в соответствии с примером из рисунка 3 (Для других номеров маршруты будут отличаться от приведённых ниже)

5:

- `ip r add 10.10.16.0/24 via 10.10.15.6 dev enp1s1f0`
- `ip r add 10.10.17.0/24 via 10.10.15.6 dev enp1s1f0`

6:

- `ip r add 10.10.17.0/24 via 10.10.16.7 dev enp1s1f1`

7:

- `ip r add 10.10.15.0/24 via 10.10.16.6 dev enp1s1f1`

8:

- `ip r add 10.10.16.0/24 via 10.10.17.7 dev enp1s1f0`
- `ip r add 10.10.15.0/24 via 10.10.17.7 dev enp1s1f0`

4.2. Проверка правильности выполнения работы

Для проверки правильности выполнения работы воспользуемся утилитой `iperf`, указав сервером машину с номером 5, а клиентом – машину с номером 8. Подробнее об использовании утилиты написано в практикуме № 1.

Практикум № 5
Динамическая настройка сети

1. ЦЕЛЬ РАБОТЫ

Целью лабораторной работы является знакомство с механизмом динамической маршрутизации. Приобретение практических навыков автоматизации получения компьютерами сетевых настроек.

2. ЗАДАНИЕ НА РАБОТУ

Компьютеры в классе соединяются схеме, указанной на рисунке 1.

Наименование интерфейсов указано для примера, на практике оно может отличаться. В нашем же случае `eth0 - enp1s1f0`.

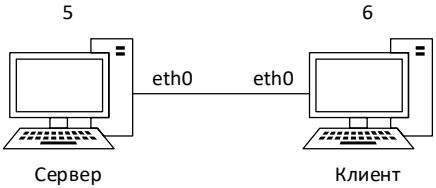


Рисунок 1. Схема подключения компьютеров

1. Настройка DHCP-сервера
    - Присвоить ip адрес серверу;
    - Настроить dnsmasq;
    - Активировать DHCP-сервер.
2. Получение ip адреса
    - Настроить сетевой интерфейс для получения адреса по DHCP;
    - Отключить сетевой менеджер;
    - Запустить DHCP-клиент.

3. ТЕРМИНЫ И ИСПОЛЬЗУЕМЫЕ КОМАНДЫ

Термины

DHCP (англ. Dynamic Host Configuration Protocol — протокол динамической настройки узла) — сетевой протокол, позволяющий сетевым устройствам автоматически получать IP-адрес и другие параметры, необходимые для работы

в сети TCP/IP. Данный протокол работает по модели «клиент-сервер». Для автоматической конфигурации компьютер-клиент на этапе конфигурации сетевого устройства обращается к так называемому серверу DHCP, и получает от него нужные параметры. Сетевой администратор может задать диапазон адресов, распределяемых сервером среди компьютеров. Это позволяет избежать ручной настройки компьютеров сети и уменьшает количество ошибок. Протокол DHCP используется в большинстве сетей TCP/IP.

Используемые команды

`nano` – текстовый редактор.

`service служба restart` – перезапускает указанную службу.

`service служба stop` – отключает указанную службу.

`ip a` – выводит все сетевые интерфейсы и информацию о них.

`mkdir директория` – создание указанной директории.

`killall имя` – закрытие всех процессов с указанным именем.

`dhcpcd` – DHCP-клиент.

`journalctl` – журнал всех системных событий.

`grep` – команда для фильтрации информации из файла. Применяется как обёртка для других команд.

4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

4.0. Предварительная настройка

Все команды в лабораторной работе выполняются с правами суперпользователя. Подробнее о том, как получить права суперпользователя написано в практикуме № 1.

Для примера выберем, что компьютеры имеют порядковые номера, как изображено на рисунке 2.

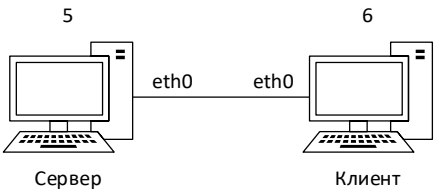


Рисунок 2. Пример схемы соединения с указанием номером рабочих мест

4.1. Настройка DHCP

Произведём настройку сервера (в примере – рабочее место с номером 5). Пропишем ip адрес `10.10.N.1` (N – номер рабочего места) на интерфейс `enp1s1f0`, предварительно убедившись, что у этого интерфейса нет адреса, введя команду

```
ip a
```

Если адрес присутствует (рисунок 3), то удалим его, введя команду

```
ip a del addr dev enp1s1f0
```

где `addr` – ip адрес у сетевого интерфейса (подчёркнут на рисунке 3).

```
[root@alt218 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp1s1f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group
default qlen 1000
    link/ether 98:a7:b0:05:72:e0 brd ff:ff:ff:ff:ff:ff
    altname enx98a7b00572e0
    inet 10.10.21.218/24 brd 10.10.21.255 scope global noprefixroute enp1s1f0
       valid_lft forever preferred_lft forever
    inet6 fe80::9aa7:b0ff:fe05:72e0/64 scope link proto kernel_ll
       valid_lft forever preferred_lft forever
3: enp1s1f1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN gr
oup default qlen 1000
    link/ether 98:a7:b0:05:72:e1 brd ff:ff:ff:ff:ff:ff
    altname enx98a7b00572e1
5: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
```

Рисунок 3. Информация о сетевых интерфейсах

Для примера будем считать, что N равно 5. В таком случае пропишем адрес для сервера

```
ip a add 10.10.5.1/24 dev enp1s1f0
```

Произведём настройку сервера для работы на интерфейсе `enp1s1f0`. Для этого зайдём в файл `/etc/dnsmasq.conf`

```
nano /etc/dnsmasq.conf
```

Перейдём в конец файла (для быстрого перехода можно зажать `CTRL+V`) и запишем туда следующие строки

```
dhcp-range=10.10.5.2,10.10.5.254,12h
interface= enp1s1f0
```

где

`10.10.5.2` – адрес, обозначающий начало диапазона, с которого будут ‘выдаваться’ адреса клиентам этого DHCP-сервера;

`10.10.5.254` – адрес, обозначающий конец диапазона;

`12h` – время, на которое выданный ip адрес действителен и не потребуется его подтверждение. Время можно указывать в секундах (10s), в минутах (32m) или часах (12h) или ‘infinite’ – бесконечность. Минимальное время – 2m;

`interface=enp1s1f0` – сетевой интерфейс, на котором будет работать DHCP-сервер.

Для записи файлов журналирования работы DHCP-сервера необходимо создать директорию `/var/lib/dhcp`. Для этого используем команду

```
mkdir /var/lib/dhcp
```

После выполнения всех команд, перезапустим утилиту dnsmasq, отвечающую за DHCP, введя команду

```
service dnsmasq restart
```

4.2. Получение ip адреса

На машинах-клиентах зайдём в файл настройки сетевого интерфейса `enp1s1f0`

```
nano /etc/net/ifaces/enp1s1f0/options
```

В нём пропишем следующие строки

```
BOOTPROTO=dhcp
NM_CONTROLLED=no
DISABLED=no
ONBOOT=yes
```

Если каких-либо строк нет, их необходимо дописать. Если в файле присутствует строка `TYPE=eth`, то удалить её.

Отключим сетевой менеджер с помощью команды

```
service NetworkManager stop
```

Данная служба отвечает за автоматическое подключение к сети. Если её не выключить, то ip адрес будет получен автоматически.

Если на сетевом интерфейсе `enp1s1f0` присутствует ip адрес, то его необходимо удалить. Пример удаления описан в пункте 4.1.

Отключим все процессы, связанные с DHCP-клиентом

```
killall dhcpcd
```

Включим новый процесс DHCP-клиента

```
dhcpcd
```

Убедиться в получении адреса можно, введя команду

```
ip a
```

Вся информация по получению ip адреса отображена в журнальном файле. Получим всю информацию по аренде сетевого адреса

```
journalctl | grep dhcpcd
```

Практикум № 6
Организация сеанса

1. ЦЕЛЬ РАБОТЫ

Познакомиться с программой Wireshark. Разобраться с помощью wireshark с трехзвенным рукопожатием и фактом разрыва соединения.

2. ЗАДАНИЕ НА РАБОТУ

1. Установите соединение с помощью утилиты `iperf` между двумя компьютерами и проследите передачу пакетов по протоколу TCP.
2. Зафиксируйте трехзвенное рукопожатие. Перечислите пакеты, участвующие в установке соединения вместе с флагами.
3. Зафиксируйте факт разрыва соединения. Перечислите пакеты, участвующие в разрыве соединения вместе с флагами.

3. ТЕРМИНЫ И ИСПОЛЬЗУЕМЫЕ КОМАНДЫ

Используемые команды

`which утилита` – команда для обнаружения установленных утилит.
`apt-get install утилита` – команда для установки указанной утилиты.
`iperf` – утилита для генерации трафика.

4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

4.0. TCP протокол

Откроем Wireshark, введя команду `wireshark`. Если он не установлен, то необходимо установить его в соответствии с порядком, описанном в практикуме № 1.

В Wireshark выберем сетевой интерфейс `enp1s1f0`. В строку фильтра пропишем `tcp` и включим сбор трафика (рисунок 1). После чего воспользуемся утилитой `iperf` для генерации tcp-трафика. Для этого один компьютер будет сервером, а другой клиентом. Подробнее об использовании `iperf` и `wireshark` написано в практикуме № 1.

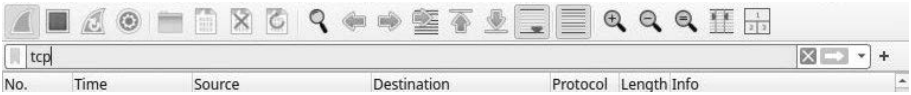


Рисунок 1. Фильтр по протоколу tcp

После некоторого времени работы Wireshark, остановим захват трафика. Из полученного трафика определим, когда произошло трёхзвенное рукопожатие. На рисунке 2 оно изображено на пакетах с номерами 1975-1077.

`[SYN] seq=0`

`[SYN,ACK] Seq=0, Ack=1`

`[ACK] Seq=1, Ack=1`

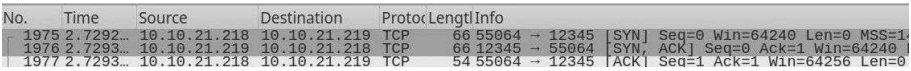


Рисунок 2. Пример tcp-трафика

4.1. Установление соединения

Теперь увидим трёхзвенное рукопожатие при входе на какой-либо сайт. Для этого в фильтре введём `ip.addr==185.71.67.49`, вы можете набрать ip-адрес любого другого сайта (для определения ip адреса другого сайта можно воспользоваться утилитой `dig домен`), затем нажимаем на синий плавник, который расположен в верхнем левом углу (рисунок 3).

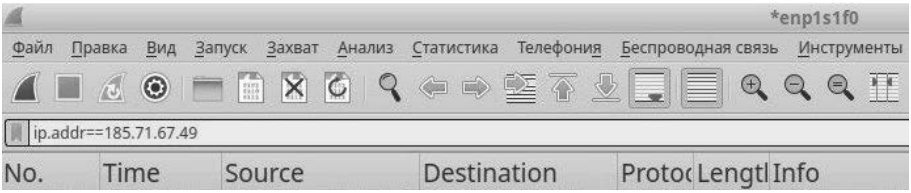


Рисунок 3. Пример введённого фильтра

Затем нажимаем на синюю стрелочку, расположенную справа в адресной строке Wireshark, тогда будут отображаться только пакеты, принадлежащие указанному ip-адресу.

После чего откроем браузер Mozilla Firefox через терминал, набрав команду `firefox`. Набираем в адресной строке браузера тот сайт, ip которого был выбран, и переходим на него, дожидаясь полной загрузки. В нашем случае это сайт mtuci.ru (рисунок 4).



Рисунок 4. Открытие страницы сайта mtuci.ru

Возвращаемся в Wireshark, останавливаем передачу пакетов, нажав на красный квадрат (расположен слева вверху) и видим трехзвенное рукопожатие (рисунок 5). Из рисунка видно, что в соединении участвуют порты 40878 и 443 (порт https), пакеты с ip-адресами 10.10.21.218 (наш ip) и 185.71.67.49 (ip сайта).

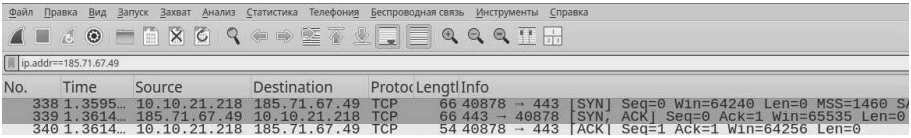


Рисунок 5. Пример трёхзвенного рукопожатия

4.2. Разрыв соединения

Факт разрыва соединения, можно зафиксировать, когда в процессе разрыва участвуют флаги и пакеты с ip-адресами 10.10.21.218 и 185.71.67.49 (рисунок 6).

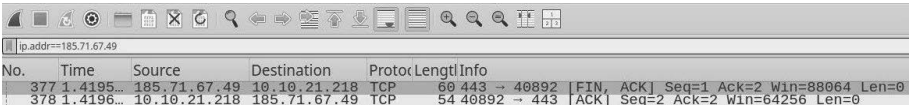


Рисунок 6. Процесс разрыва соединения

Практикум № 7
Преобразование имен на основе DNS

1. ЦЕЛЬ РАБОТЫ

Ознакомиться с файлом настроек /etc/resolv.conf. Научиться определять адреса сайтов; локального, почтового, FTP и поискового сервера. Научиться узнавать количество серверов корневой зоны и их название, а также находить сервер, ответственный за корневую зону и его адрес.

2. ЗАДАНИЕ НА РАБОТУ

1. Ознакомиться с содержимым файла настроек `/etc/resolv.conf`. Определить адрес локального сервера(серверов), заполнить таблицу 1.

Таблица 1. Локальный сервер

Локальный	

2. Определить адреса 4 случайных сайтов с помощью утилиты `dig` и заполнить таблицу 2.

Таблица 2. Адреса четырёх случайных сайтов

Имя	Адрес	Команда

3. Определить адреса почтового сервера и сервера FTP basealt.ru и записать их в таблицу 3.

Таблица 3. Почтовые сервера и FTP сервер basealt.ru

Имя	Адрес	Команда

4. Определите все сервера, принадлежащие mail.ru, и занести их в таблицу 4

Таблица 4. Сервера, принадлежащие mail.ru

Имя	Адрес	Команда

5. Узнать название сетевых ресурсов по их адресам

Таблица 5. Адреса сетевых ресурсов

Адреса	Сетевые ресурсы
5.255.255.242	
104.154.127.47	
91.201.52.134	
92.255.60.103	

6. Определить количество серверов корневой зоны и сервер, ответственный за корневую зону.

7. Определить количество серверов зоны su и сервер, ответственный за эту зону.

8. Определить количество серверов arpa.

Таблицы необходимо создать с помощью текстового редактора LibreOffice Writer и сохранить на рабочем столе. Название файла должно содержать номер группы и фамилию студента.

3. ТЕРМИНЫ И ИСПОЛЬЗУЕМЫЕ КОМАНДЫ

Используемые команды

`dig` - утилита (DNS-клиент), предоставляющая пользователю интерфейс командной строки для обращения к системе DNS. Позволяет задавать различные типы запросов и запрашивать произвольно указываемые сервера.

`nslookup` — утилита, предоставляющая пользователю интерфейс командной строки для обращения к системе DNS (проще говоря, DNS-клиент). Позволяет задавать различные типы запросов и опрашивать произвольно указываемые сервера.

`cat` – команда для работы с файлами.

`host` – команда, позволяющая преобразовывать ip адрес в название сетевого ресурса и наоборот.

4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

4.1. Изучение файла resolv.conf

Используем команду `cat` для открытия файла `resolv.conf`

```
cat /etc/resolv.conf
```

Результат выполнения изображён на рисунке 1. Из рисунка видно, что в файле находится название локального (nameserver) сервера. Полученные данные необходимо занести в таблицу 1.

```
[user@alt218 Рабочий стол]$ cat /etc/resolv.conf
# Generated by resolvconf
# Do not edit manually, use
# /etc/net/ifaces/<interface>/resolv.conf instead.
nameserver 8.8.8.8
```

Рисунок 1. Содержимое файла resolv.conf

4.2. Определение адресов различных сайтов

Для определения адреса сайта можно воспользоваться командой `dig` (рисунок 2) или командой `nslookup` (рисунок 3). Например, получим адрес сайта ya.ru

```
dig ya.ru
nslookup ya.ru
```

Необходимо определить адреса четырёх различных сайтов и заполнить таблицу 2.

```
[user@alt218 Рабочий стол]$ dig ya.ru

; <<>> DiG 9.18.41 <<>> ya.ru
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 24423
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;ya.ru.                         IN      A

;; ANSWER SECTION:
ya.ru.                  45      IN      A       5.255.255.242
ya.ru.                  45      IN      A       77.88.55.242
ya.ru.                  45      IN      A       77.88.44.242

;; Query time: 16 msec
;; SERVER: 8.8.8.8#53(8.8.8.8) (UDP)
;; WHEN: Tue Dec 23 00:18:02 MSK 2025
;; MSG SIZE  rcvd: 82
```

Рисунок 2. Применение команды `dig` для определения адреса сайта

Из рисунка 2 видно, что ya.ru имеет несколько адресов.

```
[user@alt218 Рабочий стол]$ nslookup ya.ru
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
Name:   ya.ru
Address: 5.255.255.242
Name:   ya.ru
Address: 77.88.55.242
Name:   ya.ru
Address: 77.88.44.242
Name:   ya.ru
Address: 2a02:6b8::2:242
```

Рисунок 3. Применение команды `nslookup` для определения адреса сайта

4.3. Определение почтового и FTP серверов basealt.ru

Для того чтобы узнать адрес почтового сервера и сервера FTP можно использовать либо `dig`, либо `nslookup`. Обе команды работают примерно одинаково, для примера рассмотрим вариант с командой `nslookup`. Для определения почтового сервера используем команду (рисунок 4)

```
nslookup -type=mx basealt.ru
```

где

mx – mail exchange (почтовый обмен) – это запись на сервере имен (DNS), которая указывает на сервер почты для данного домена.
-type — тип информации, которую хотим получить.

```
[user@alt218 Рабочий стол]$ nslookup -type=mx basealt.ru
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
basealt.ru      mail exchanger = 10 air.basealt.ru.

Authoritative answers can be found from:
```

Рисунок 4. Почтовый сервер basealt.ru

Чтобы узнать адрес FTP сервера, используем команду `dig`. Команда выглядит следующим образом (рисунок 5)

```
dig ftp.basealt.ru
```

```
[user@alt218 Рабочий стол]$ dig ftp.basealt.ru

; <<>> DiG 9.18.41 <<>> ftp.basealt.ru
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4471
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;ftp.basealt.ru.                        IN      A

;; ANSWER SECTION:
ftp.basealt.ru.         21600   IN      CNAME   ftp.altlinux.org.
ftp.altlinux.org.       21600   IN      A       62.152.55.238

;; Query time: 30 msec
;; SERVER: 8.8.8.8#53(8.8.8.8) (UDP)
;; WHEN: Tue Dec 23 00:21:18 MSK 2025
;; MSG SIZE  rcvd: 89
```

Рисунок 5. Определение ftp сервера

Из полученных данных заполните таблицу 3.

4.4. Определение серверов сайта mail.ru

Чтобы узнать все сервера, потребуется несколько команд:

```
nslookup mail.ru
nslookup ftp.mail.ru
nslookup mx.mail.ru
nslookup –type=ns mail.ru
```

Команда `nslookup mail.ru` выводит адреса серверов сайта mail.ru (рисунок 6).

Команда `nslookup ftp.mail.ru` выводит адреса ftp серверов сайта mail.ru (рисунок 7).

Команда `nslookup mx.mail.ru` выводит почтовые сервера сайта mail.ru (рисунок 8).

```
[user@alt218 Рабочий стол]$ nslookup mail.ru
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
Name:   mail.ru
Address: 89.221.239.1
Name:   mail.ru
Address: 90.156.232.4
Name:   mail.ru
Address: 185.180.201.1
```

Рисунок 6. Адреса серверов сайта mail.ru

```
[user@alt218 Рабочий стол]$ nslookup ftp.mail.ru
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
Name:   ftp.mail.ru
Address: 217.69.139.87
Name:   ftp.mail.ru
Address: 94.100.180.87
```

Рисунок 7. Адреса ftp серверов сайта mail.ru

```
[user@alt218 Рабочий стол]$ nslookup mx.mail.ru
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
Name:   mx.mail.ru
Address: 217.69.139.87
Name:   mx.mail.ru
Address: 94.100.180.87
```

Рисунок 8. Адреса почтовых серверов сайта mail.ru

Определим, сколько ns-серверов у сайта mail.ru, введя команду

```
nslookup –type=ns mail.ru
```

Получим несколько серверов в ответе. Для каждого из них используем команду

```
nsX.mail.ru
```

где X – номер сервера.

Результат выполнения команд представлен на рисунке 9.

```
[user@alt218 Рабочий стол]$ nslookup -type=ns mail.ru
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
mail.ru nameserver = ns1.mail.ru.
mail.ru nameserver = ns2.mail.ru.

Authoritative answers can be found from:

[user@alt218 Рабочий стол]$ nslookup ns1.mail.ru
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
Name:   ns1.mail.ru
Address: 217.69.139.112
Name:   ns1.mail.ru
Address: 2a00:1148:db00::2
```

Рисунок 9. Список ns-серверов и их адреса для сайта mail.ru

4.5. Определение имени сетевого ресурса

Для преобразования адресов в названия ресурсов воспользуемся утилитой host

```
host –t ptr ip
```

Параметр `-t` указывает на тип записи. Запись `ptr` связывает ip адрес с именем хоста. Для примера выполним команду `host –t ptr 5.255.255.242` (рисунок 10).

```
[user@alt218 Рабочий стол]$ host -t ptr 5.255.255.242
242.255.255.5.in-addr.arpa domain name pointer ya.ru.
```

Рисунок 10. Пример выполнения команды `host`

4.6. Изучение серверов корневой зоны

Чтобы узнать количество серверов корневой зоны, воспользуемся командой (рисунок 11)

```
nslookup –type=ns .
```

Обратите внимание: точка “`.`” в конце команды означает корневую DNS-зону. Без точки команда будет искать доменное имя, а не корневые DNS-сервера.

```
[user@alt218 Рабочий стол]$ nslookup -type=ns .
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
.       nameserver = d.root-servers.net.
.       nameserver = g.root-servers.net.
.       nameserver = j.root-servers.net.
.       nameserver = a.root-servers.net.
.       nameserver = m.root-servers.net.
.       nameserver = k.root-servers.net.
.       nameserver = i.root-servers.net.
.       nameserver = c.root-servers.net.
.       nameserver = f.root-servers.net.
.       nameserver = h.root-servers.net.
.       nameserver = e.root-servers.net.
.       nameserver = l.root-servers.net.
.       nameserver = b.root-servers.net.

Authoritative answers can be found from:
```

Рисунок 11. Список серверов корневой зоны

Для определения сервера, ответственного за корневую зону воспользуемся командой (рисунок 12)

```
nslookup -type=soa .
```

Обратите внимание: точка “.” в конце команды означает корневую DNS-зону. Без точки команда будет искать доменное имя, а не корневые DNS-сервера.

После определения этого сервера, получим его ip адрес, используя команду `nslookup`.

```
[user@alt218 Рабочий стол]$ nslookup -type=soa .
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
.
        origin = a.root-servers.net
        mail addr = nstld.verisign-grs.com
        serial = 2025122200
        refresh = 1800
        retry = 900
        expire = 604800
        minimum = 86400

Authoritative answers can be found from:

[user@alt218 Рабочий стол]$ nslookup a.root-servers.net
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
Name:   a.root-servers.net
Address: 198.41.0.4
Name:   a.root-servers.net
Address: 2001:503:ba3e::2:30
```

Рисунок 12. Сервер, ответственный за корневую зону и его ip адрес

4.7. Изучение серверов, ответственных за зону su

Для выполнения этого пункта будем использовать команды, которыми пользовались в пункте 4.5, но с одним отличием – поменяем “.” на “su”

```
nslookup -type=ns su
```

Определим сервер, ответственный за зону su и его адрес

```
nslookup -type=soa su
```

4.8. Определение корневых серверов зоны arpa

Для получения всех корневых серверов зоны arpa воспользуемся командой

```
nslookup –type=ns arpa
```

Результат выполнения команды представлен на рисунке 12.

```
[user@alt218 Рабочий стол]$ nslookup -type=ns arpa
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
arpa    nameserver = c.ns.arpa.
arpa    nameserver = f.ns.arpa.
arpa    nameserver = a.ns.arpa.
arpa    nameserver = e.ns.arpa.
arpa    nameserver = m.ns.arpa.
arpa    nameserver = i.ns.arpa.
arpa    nameserver = g.ns.arpa.
arpa    nameserver = h.ns.arpa.
arpa    nameserver = k.ns.arpa.
arpa    nameserver = l.ns.arpa.
arpa    nameserver = d.ns.arpa.
arpa    nameserver = b.ns.arpa.

Authoritative answers can be found from:
```

Рисунок 12. Список корневых серверов arpa

Практикум № 8
Передача нешифрованной и зашифрованной информации между компьютерами

1. ЦЕЛЬ РАБОТЫ

Изучение и сравнение зашифрованных и незашифрованных соединений с помощью Wireshark. Ознакомление с утилитой ssh в дистрибутиве ОС Альт.

2. ЗАДАНИЕ НА РАБОТУ

1. Подключиться к консоли соседнего компьютера с помощью утилиты ssh;
2. Запустить удалённое графическое приложение на компьютере соседа;
3. Сравнить с помощью Wireshark сеансы netcat и ssh. Зафиксировать особенности этих соединений (порт отправителя, порт получателя, протокол передачи данных, указать какое из соединений шифруется);
4. Сравнить с помощью Wireshark HTTP и HTTPS соединения. Зафиксировать особенности таких соединений (протокол передачи данных, указать какое из соединений шифруется).

3. ТЕРМИНЫ И ИСПОЛЬЗУЕМЫЕ КОМАНДЫ

Используемые команды

`ssh` – утилита для регистрации на удаленной машине и выполнения на ней команд.

`ssh-keygen` – создание, обслуживание и преобразование аутентификационных ключей.

`ssh-copy-id` – копирование вашего публичного ключа для ssh соединения на указанную машину для указанного пользователя.

`netcat` – утилита для передачи текстовых сообщений по протоколу tcp между двумя машинами.

`netstat` – утилита для отображения содержания различных структур данных, связанных с сетью.

`wireshark` – приложение для “захвата” трафика с целью его последующего анализа.

`firefox` – веб-браузер.

4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

4.1. Генерация ключа и подключение к соседнему компьютеру

Изначально для выполнения соединения по протоколу SSH потребуется создать SSH-ключ. Для его создания воспользуемся следующей командой

```
ssh-keygen
```

При её вводе консоль предложит создать новый SSH-ключ, состоящий из публичной и приватной частей. Публичная часть хранится в `/$HOME/.ssh/id_rsa.pub`, приватная часть в: `/$HOME /.ssh/id_rsa`, где `$HOME` – домашний каталог пользователя, под которым студент находится в терминале.

Без указания специальных параметров-ключей будет создан ключ для шифрования по алгоритму `rsa`.

Результат выполнения команды представлен на рисунке 1.

```
[user@alt218 Рабочий стол]$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/user/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/.ssh/id_ed25519
Your public key has been saved in /home/user/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:iRosq8OBCv6lenGKjtu58SakJG5S0iVlwVC1KnPnfYw user@alt218
The key's randomart image is:
+--[ED25519 256]--+
|  .+oo.          |
|    +  .         |
|   o  .          |
|  .... . .       |
|..+o= o S        |
|=o+B * . o       |
|O== =.. E o      |
|*O.*+    .       |
|B=B*.            |
+----[SHA256]-----+
```

Рисунок 1. Создание нового ssh ключа

После создания ключа, скопируем его на удалённую машину соседа, чтобы, в дальнейшем, подключаться по этому ключу без ввода пароля. Для этого используем следующую команду

```
ssh-copy-id user@ipaddr
```

Выше в команде `user` – пользователь, под которым мы будем находиться на удалённой машине, `ipaddr` – ip адрес машины соседа. Для примера примем, что имя пользователя – `user`, а ip адрес – `10.10.21.219`. Таким образом команда пример следующий вид

```
ssh-copy-id user@10.10.21.219
```

При вводе команды копирования SSH-ключа на удалённую машину потребуется ввести пароль от того пользователя, который был указан в команде `ssh-copy-id`. После выполнения всех манипуляций подключение к машине с ip адресом `10.10.21.219` под пользователем `user` можно производить без ввода пароля.

На рисунке 2 представлен процесс использования команды `ssh-copy-id user@10.10.21.219`.

```
[user@alt218 Рабочий стол]$ ssh-copy-id user@10.10.21.219
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/user/.ssh/id_ed255
19.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any
 that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now
it is to install the new keys
user@10.10.21.219's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'user@10.10.21.219'"
and check to make sure that only the key(s) you wanted were added.
```

Рисунок 2. Передача ключа на удалённую машину

Для подключения используем команду (рисунок 3)

```
ssh user@10.10.21.219
```

```
[user@alt218 Рабочий стол]$ ssh user@10.10.21.219
Last login: Mon Dec 22 23:34:34 2025 from 10.10.21.218
[user@alt219 ~]$
```

Рисунок 3. Подключение к удалённой машине

На рисунке 3 можно увидеть успешное подключение к удалённой машине с адресом 10.10.21.219 – в терминале пользователь сменился на `user@alt219`.

4.2. Подключение в качестве графической оболочки

В первом пункте уже было установлено ssh-соединение. Его необходимо закрыть, нажав комбинацию клавиш `CTRL+D`.

Для запуска удалённого графического приложения на компьютере соседа, необходимо подключиться к нему с помощью следующей команды:

```
ssh -YC user@10.10.21.219
```

При подключении не потребуется ввод пароля, так как манипуляции в первом пункте избавили нас от этой надобности.

Для проверки работы удалённого графического приложения откроем графический текстовой редактор `pluma` на удалённой машине `10.10.21.219`. На рисунке 4 можно увидеть удачный запуск приложения на удалённой машине.

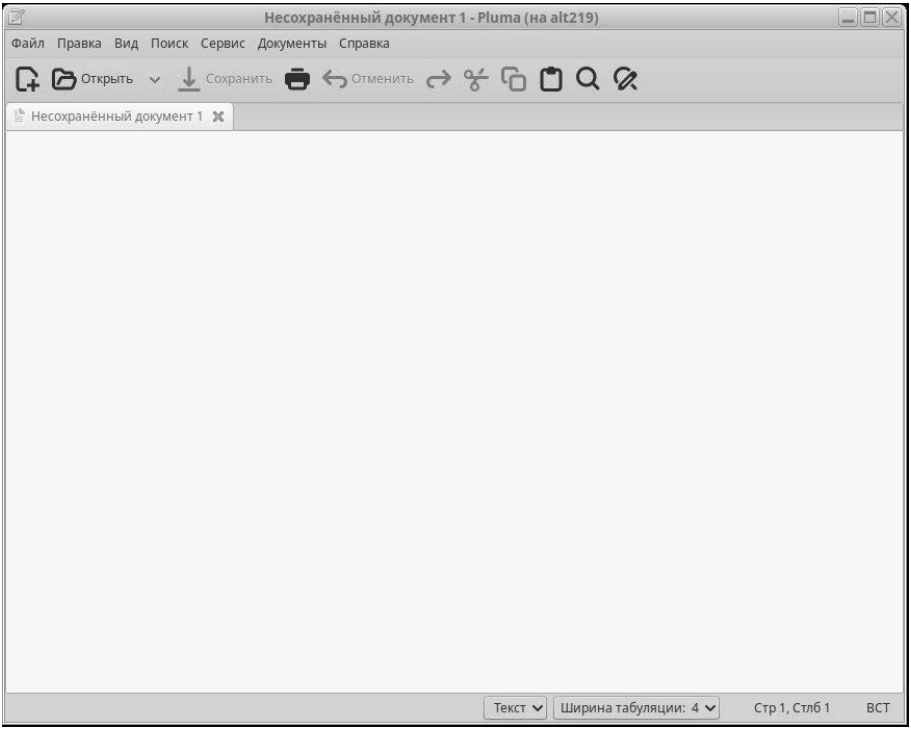


Рисунок 4. Запущенное приложение `pluma` на удалённой машине

4.3. Анализ трафика netcat и ssh

Для выполнения этого пункта необходимо включить Wireshark. Воспользуемся терминалом, введя в него команду

```
wireshark
```

На одном из компьютеров требуется активировать утилиту `netcat` на «прослушивание» всех входящих запросов по указанному порту

```
netcat -l port
```

Выше, `port` – номер порта «прослушивания», по которому ожидается поступление запроса. Например, `netcat -l 12345`.

На другом компьютере требуется подключиться к компьютеру, где была активирована утилита netcat на прослушивание входящих запросов

```
netcat ipaddr port
```

Выше, `ipaddr` – ip адрес машины, на которой активен `netcat`, `port` – номер порта, по которому ожидается поступление запроса. Для примера, как и ранее, примем, что ip адрес машины – `10.10.21.219`, а порт – `12345`. Получим следующие команды:

- Для удалённой машины

```
netcat -l 12345
```

- Для локальной машины

```
netcat 10.10.21.219 12345
```

Теперь отследим этот процесс в Wireshark. Подробнее об использовании Wireshark написано в практикуме № 1. При вводе какой-либо строки в терминале с включенной утилитой `netcat`, в Wireshark отразится пакет, который передаётся с локальной машины на удалённую на порт `12345`. При этом всю передаваемую информацию и текст можно увидеть на экране.

Перейдём ко второй части задания – вновь подключимся по `ssh` к удалённой машине соседа (процедура подключения подробно описана в пункте 4.1). Так же, как и выше, при вводе какой-либо строки в терминале при подключении через `ssh`, в Wireshark отразится пакет, однако в данном случае передаваемую информацию и текст увидеть не получится, так как она будет зашифрована.

Пример перехвата пакетов для утилиты `netcat` изображён на рисунке 5. Содержимое одного из пакетов изображено на рисунке 6. Как видно из

содержимого, передаваемое сообщение не шифруется. На рисунке 7 изображены пакеты утилиты `ssh`.

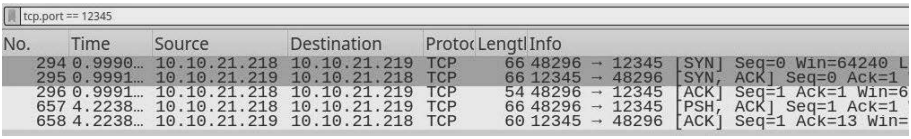


Рисунок 5. Пакеты, передающиеся при использовании `netcat`

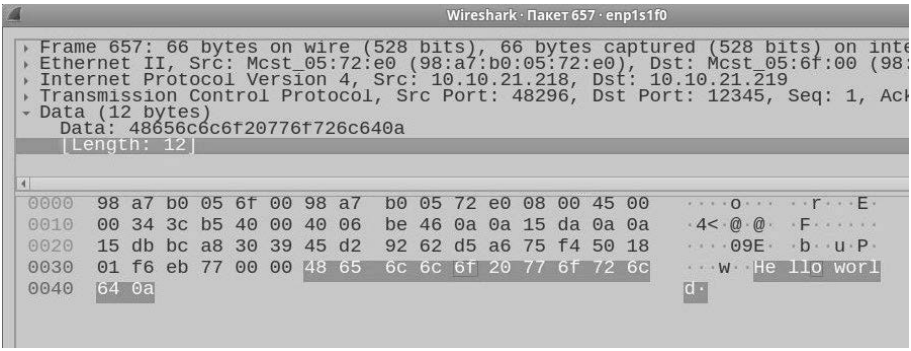


Рисунок 6. Содержимое пакета при использовании `netcat`

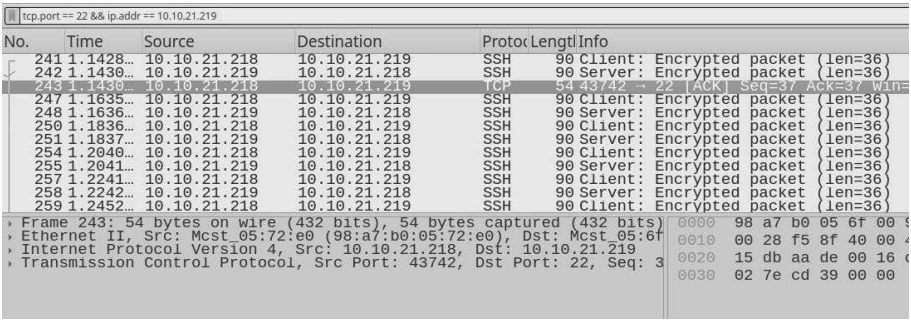


Рисунок 7. Пакеты, передающиеся при использовании `ssh`

4.4. Анализ трафика HTTP и HTTPS

В данном пункте, как и в пункте три, потребуется использовать приложение Wireshark. Для начала выберем два сайта. Первый – соединение, к которому происходит по протоколу HTTP, второй – соединение по HTTPS. Например, сайт `http://www.partizansk.org` и `https://habr.com/ru`.

Откроем Mozilla Firefox, для этого в терминале введём команду

```
firefox
```

В открывшемся окне в адресную строку введём первый адрес. В нашем примере это `http://www.partizansk.org`. При этом в wireshark будут отображены пакеты, которые были получены при подключении к сайту. В отчёт необходимо занести протокол, по которому состоялось соединение.

Теперь подключимся ко второму сайту - `https://habr.com/ru`. Для него так же определим протокол, по которому произошло соединение. Полученный протокол требуется занести в отчёт. Дополнительно в отчёте необходимо указать какое из этих двух соединений шифруется.

Практикум № 9
Трансляция адресов (NAT)

1. ЦЕЛЬ РАБОТЫ

Целью лабораторной работы является освоение навыков настройки статической и динамической трансляции адресов. Изучение механизма NAT.

2. ЗАДАНИЕ НА РАБОТУ

Компьютеры соединены по схеме, изображённой на рисунке 1.

Наименование интерфейсов указано для примера, на практике оно может отличаться. В нашем же случае `eth0 – enp1s1f0`, а `eth1 – enp1s1f1`.

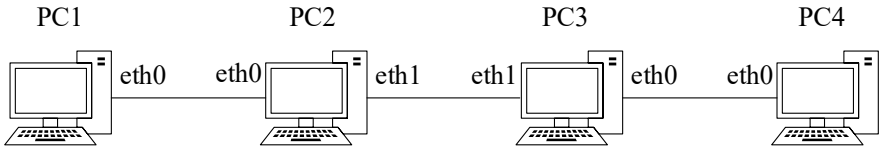


Рисунок 1. Схема сети

1. Настройка сетевых интерфейсов
   - Необходимо настроить конфигурационные файлы в /etc/net/ifaces.
2. Настройка трансляции адресов
   - При обращении с узла 1 к службе, расположенной на узле 3, обеспечьте возможность скрытия узла 1 средствами узла 2;
   - Обеспечьте возможность переноса службы с узла 2 на узел 3 так, чтобы при обращении к ней с узла 1 узел 2 считался «прозрачным».

3. ТЕРМИНЫ И ИСПОЛЬЗУЕМЫЕ КОМАНДЫ

Используемые команды

`ip` – утилита для работы с сетевыми интерфейсами.

`iptables` – утилита для управления работы межсетевым экранированием.

Некоторые опции команды `iptables`:

`-t таблица` – задаёт таблицу, к которой будет применена команда.

`-A цепочка` – добавляет одно или несколько правил в конец указанной цепочки. Виды цепочек – PREROUTING (для изменения всех входящих пакетов), INPUT (для пакетов, направленных в локальную сеть), FORWARD (для проходящих пакетов), OUTPUT (для пакетов, исходящих из локальной сети), POSTROUTING (для изменения всех исходящих пакетов).

`-j цель` – определяет цель правила; т.е., что делать, когда пакет попадает под условия правила. Целью может быть цепочка, определённая пользователем (отличная от цепочки правила), одна из встроенных целей, определяющая окончательное действие над пакетом, или расширение.

`-s ip` – адрес источника пакета.

`-d ip` – адрес назначения пакета.

`netcat` – утилита для передачи текстовых сообщений по протоколу tcp между двумя машинами.

`sysctl` – команда для считывания/записи параметров системы.

4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

4.0. Предварительная настройка

Все команды в лабораторной работе выполняются с правами суперпользователя. Подробнее о том, как получить права суперпользователя написано в практикуме № 1.

Для примера выберем, что компьютеры имеют порядковые номера, как изображено на рисунке 2.

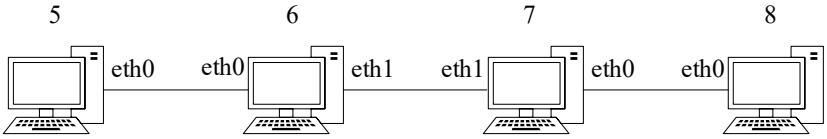


Рисунок 2. Пример схемы соединения с указанием номером рабочих мест

4.1. Настройка конфигурации

Пропишем ip адрес и маршрут для каждого компьютера. Адрес будем определять согласно номеру рабочего места компьютера по следующему правилу

```
10.10.XY.Z/24
```

где

X – номер ряда с рабочими местами;

Y – минимальный номер между двумя рабочими местами;

Z – номер рабочего места.

Предположим, что мы проводим настройку 1 ряда. Согласно примеру с рисунка 2, компьютерам необходимо присвоить следующие адреса:

5:
- 10.10.15.5/24 на интерфейсе enp1s1f0

6:
- 10.10.15.6/24 на интерфейсе enp1s1f0
- 10.10.16.6/24 на интерфейсе enp1s1f1

7:
- 10.10.16.7/24 на интерфейсе enp1s1f1
- 10.10.17.7/24 на интерфейсе enp1s1f0

8:
- 10.10.17.8/24 на интерфейсе enp1s1f0

Для шестого и седьмого компьютеров разрешим перемещение трафика с интерфейса на интерфейс

```
sysctl –w net.ipv4.ip_forward=1
```

4.2. Настройка трансляции адресов

Для настройки трансляции сетевых адресов существует утилита iptables. Воспользуемся ей, для подмены источника пакета при передаче пакетов в локальной сети.

На шестом и седьмом компьютерах выполним

```
iptables -t nat -A POSTROUTING –o enp1s1f1 -j
MASQUERADE
```

Для проверки правильности выполнения работы воспользуемся утилитой iperf, указав сервером машину с номером 5, а клиентом – машину с номером 7 или сервер – машина 8, а клиент – 6. Подробнее об использовании утилиты написано в практикуме № 1.

Практикум № 10
Веб-сервер Apache

1. ЦЕЛЬ РАБОТЫ

Изучение и настройка веб-сервера Apache на операционной системе ОС Альт.

2. ЗАДАНИЕ НА РАБОТУ

1. Настройте веб-сервер таким образом, чтобы по адресу sX.example.com (где X – номер рабочего места) открывалась веб-страница с текстом “Test”.
2. При помощи механизма htaccess настройте пароль для доступа к веб-странице, созданной в пункте 1.
3. Настройте https доступ к странице, созданной в пункте 1.

3. ТЕРМИНЫ И ИСПОЛЬЗУЕМЫЕ КОМАНДЫ

Термины

Веб-сервер – это сервер, принимающий HTTP-запросы от клиентов и выдающий им HTTP-ответы, обычно вместе с HTML-страницей, изображением, файлом, медиа-потоком или другими данными. Веб-сервером называют как программное обеспечение, выполняющее функции веб-сервера, так и непосредственно компьютер, на котором это программное обеспечение работает. Клиент, которым обычно является веб-браузер, передаёт веб-серверу запросы на получение ресурсов, обозначенных URL-адресами.

Используемые команды

`which утилита` – команда для обнаружения установленных утилит.
`apt-get install утилита` – команда для установки указанной утилиты.
`nano, pluma` – текстовые редакторы.
`firefox` – веб-браузер.
`cat > файл` – команда для перевода всех сообщений в указанный файл.
`a2ensite` – активация сайта в apache.
`a2enmod` – активация модуля в apache.
`a2enport` – активация порта в apache.

`systemctl restart httpd2` – перезапуск apache.
`htpasswd файл` – команда для создания нового пользователя и сохранения его пароля в указанный файл.
`openssl` – команда для работы с ssl сертификатами.

4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

4.1. Установка необходимых утилит

Перед началом выполнения лабораторной работы необходимо убедиться в том, что все необходимые утилиты были установлены, для этого получим права суперпользователя, использовав команду (подробнее об использовании команды см. практикум № 1)

```
su -
```

Далее, введём команду

```
which apache2
```

Если пакет не удалось обнаружить (рисунок 1), то установим его с помощью команды

```
apt-get install apache2
```

После ввода команды (рисунок 2) необходимо ввести `y`.

```
[user@alt218 ~]$ which apache2
which: no apache2 in (/home/user/bin:/usr/bin:/bin:/usr/local/bin:/usr/games)
[user@alt218 ~]$ _
```

Рисунок 1. Отсутствие пакета apache2 в системе

```
[root@alt218 ~]# apt-get install apache2
Чтение списков пакетов... Завершено
Построение дерева зависимостей... Завершено
Следующие дополнительные пакеты будут установлены:
  apache2-ab                    apache2-mod_cache_disk
  apache2-base                  apache2-mods
  apache2-cgi-bin               condstopstart-common
  apache2-cgi-bin-printenv      condstopstart-web
  apache2-cgi-bin-test-cgi      libapr1
  apache2-datadirs              libaprutil1
  apache2-htcacheclean          libdb5.3
  apache2-htcacheclean-control  perl-DBM
  apache2-html                  webserver-cgi-bin-control
  apache2-htpasswd              webserver-common
  apache2-httpd-worker          webserver-common-control
  apache2-icons
Следующие НОВЫЕ пакеты будут установлены:
  apache2                       apache2-icons
  apache2-ab                    apache2-mod_cache_disk
  apache2-base                  apache2-mods
  apache2-cgi-bin               condstopstart-common
  apache2-cgi-bin-printenv      condstopstart-web
  apache2-cgi-bin-test-cgi      libapr1
```

Рисунок 2. Установка пакета apache2

Результат установки изображён на рисунке 3.

```
17: apache2-htcacheclean-1:2.4.65-alt1  ################################# [ 71%]
auto
daemon
18: apache2-mod_cache_disk-1:2.4.65-alt1################################# [ 75%]
19: apache2-cgi-bin-test-cgi-1:2.4.65-al################################# [ 79%]
20: apache2-cgi-bin-printenv-1:2.4.65-al################################# [ 83%]
21: apache2-cgi-bin-1:2.4.65-alt1       ################################# [ 88%]
22: apache2-html-1:2.4.65-alt1          ################################# [ 92%]
23: apache2-icons-1:2.4.65-alt1         ################################# [ 96%]
24: apache2-1:2.4.65-alt1               ################################# [100%]
Завершено.
[root@alt218 ~]#
```

Рисунок 3. Результат установки пакета apache2

4.2. Создание веб-страницы

Стандартный конфигурационный файл для сайтов веб-сервера Apache находится в директории `/etc/httpd2/conf/sites-available`. Если актуальная директория – не домашняя (обозначается символом “~” в терминале), то необходимо предварительно выполнить команду `cd ~`.

```
cd /etc/httpd2/conf/sites-available
```

```
[root@alt218 ~]# cd /etc/httpd2/conf/sites-available/
[root@alt218 sites-available]#
```

Рисунок 4. Процесс перемещения в директорию sites-available

В этом каталоге находится файл `default.conf`. Скопируем его, в файл `site1.conf` (название выбрано для примера), использовав команду (рисунок 5)

```
cp default.conf site1.conf
```

```
[root@alt218 sites-available]# cp default.conf site1.conf
[root@alt218 sites-available]#
```

Рисунок 5. Копирование содержимого файла `default.conf` в файл `site1.conf`

Откроем ранее созданный файл в текстовом редакторе `pluma`, введя команду (рисунок 6)

```
pluma site1.conf
```

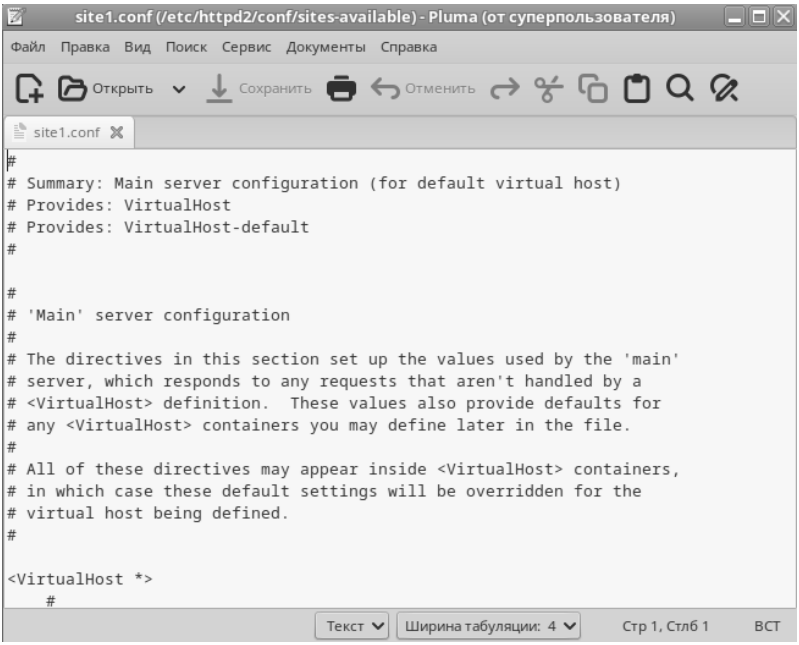


Рисунок 6. Фрагмент файла `site1.conf`

В открывшемся редакторе в разделе `<VirtualHost *>` необходимо заменить:

```
#ServerName www.example.com:80
→
ServerName www.s00.example.com

DocumentRoot “/var/www/html”
→
DocumentRoot “/var/www/site1”

<Directory “/var/www/html”>
→
<Directory “/var/www/site1”>
```

Если студент выбрал другое название для файла site1.conf, то название необходимо сменить, чтобы они совпадали (пусть ранее была выполнена команда `cp default.conf test007.conf`, тогда название директории должно иметь вид `/var/www/test007`). Результат изменений отображён на рисунке 7.

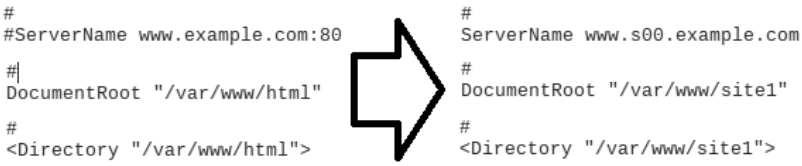


Рисунок 7. Строки, которые необходимо изменить

После всех манипуляций сохраним изменённый файл, кликнув на “Сохранить” (рисунок 8).

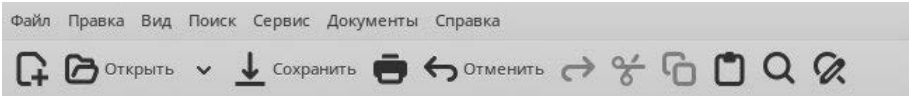


Рисунок 8. Внешний вид панели редактора Pluma

Теперь создадим директорию, которую указали в DocumentRoot. Для этого перейдём в каталог /var/www/, использовав команду

```
cd /var/www/
```

После чего создадим новую директорию site1 и перейдём в неё. Процесс выполнения отображён на рисунке 9.

```
[root@alt218 sites-available]# cd /var/www
[root@alt218 www]# mkdir site1
[root@alt218 www]# cd site1/
[root@alt218 site1]#
```

Рисунок 9. Процесс создания и перемещения в директорию site1

В каталоге создадим файл index.html, куда запишем Hello elbrus! Для создания можно использовать текстовый редактор pluma, nano, либо утилиту cat. Рассмотрим вариант с использованием утилиты cat.

Введя команду

```
cat > index.html
```

Весь последующий набранный текст будет сохранён в указанный файл. Для окончания ввода потребуется нажать CTRL+D. Пример выполнения изображён на рисунке 10.

```
[root@alt218 site1]# cat > index.html
Hello elbrus!
[root@alt218 site1]#
```

Рисунок 10. Запись строки в текстовый файл

Теперь необходимо сопоставить созданной веб-странице ip адрес, чтобы браузер знал, где искать запрашиваемый сайт. Для этого откроем файл hosts, находящийся в /etc/hosts и допишем туда следующий текст: 127.0.0.5 www.s00.example.com (название сайта должно соответствовать тому, что было указано в ServerName). Для открытия файла воспользуемся утилитой nano

```
nano /etc/hosts
```

Результат выполнения изображён на рисунке 11. Количество строк может отличаться от того, что показаны на рисунке. Для сохранения файла нажмём CTRL+X, далее, следуя инструкциям, введём y и нажмём enter.

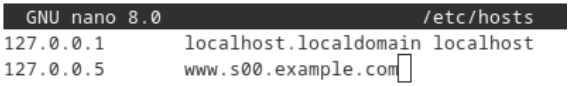


Рисунок 11. Содержимое файла `/etc/hosts`

Остался заключительный шаг – включение сайта и перезагрузка веб-сервера. Команда

```
a2ensite site1
```

Активирует сайт, но для её корректной работы необходим перезапуск веб-сервера с помощью команды (рисунок 12)

```
systemctl restart httpd2
```

```
[root@alt218 site1]# a2ensite site1
Site site1 installed;
        run service httpd2 condreload to fully enable.
[root@alt218 site1]# systemctl restart httpd2
```

Рисунок 12. Активация сайта `site1`

Для проверки правильности выполненной работы, откроем веб-браузер `firefox` и введём в адресную строку `www.s00.example.com`. Если, в итоге, будет открыта страница, где написано `Hello elbrus!`, то пункт задания выполнен верно. Пример страницы с надписью `Hello Elbrus!` приведён на рисунке 13.

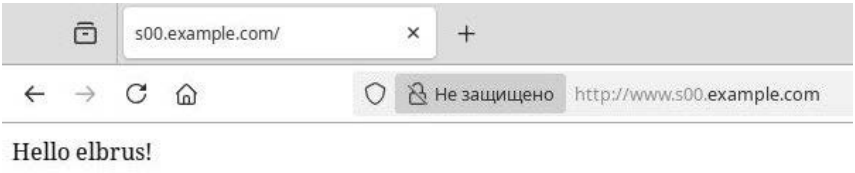


Рисунок 13. Фрагмента сайта, открытого в веб-браузере

4.3. Настройка доступа к веб-странице

Для начала требуется включить модули, которые отвечают за доступ по паролю. Для их активации модулей используется команда

`a2enmod модуль`

Список модулей представлен ниже:

- `auth_basic`;
- `authn_core`;
- `authn_file`;
- `authz_user`.

Листинг выполнения команды представлен на рисунке 14.

```
[root@alt218 ~]# a2enmod auth_basic
Module auth_basic installed;
        run service httpd2 condreload to fully enable.
[root@alt218 ~]# a2enmod authn_core
Module authn_core installed;
        run service httpd2 condreload to fully enable.
[root@alt218 ~]# a2enmod authn_file
Module authn_file installed;
        run service httpd2 condreload to fully enable.
[root@alt218 ~]# a2enmod authz_user
Module authz_user installed;
        run service httpd2 condreload to fully enable.
```

Рисунок 14. Активация модулей

Для активации запроса пароля потребуется создать файл в директории сайта: `/var/www/site1` с названием `.htaccess`. Для этого воспользуемся текстовым редактором `nano`. В файл необходимо записать:

```
AuthType Basic
AuthName “Private area”
AuthUserFile “/var/www/site1/htpasswd”
Require user elbrus
```

Выше:

`AuthType Basic` – для аутентификации использовать базовый модуль `auth_basic` (существует модули `auth_digest` и `auth_form`. Первый перенаправляет пользователя на отдельную страницу аутентификации, а второй на отдельную форму);

`AuthName “Private area”` – текст, который будет выведен как информация при запросе аутентификации;

`AuthUserFile "/var/www/site1/htpasswd"` – путь к файлу, в котором находятся пользователи и пароли от них. Только от имени тех, кто перечислен в файле можно зайти на веб-страницу.

`Require user elbrus` – какой пользователь имеет доступ к веб-странице. В нашем примере – `elbrus`.

Пример заполнения файла приведён на рисунке 15.

```
 GNU nano 8.0                          .htaccess
AuthType Basic
AuthName "Private Area"
AuthUserFile "/var/www/site0/htpasswd"
Require user elbrus
```

Рисунок 15. Содержимое файла `.htaccess`

Теперь создадим пользователя и пароль к нему, чтобы предоставить доступ на веб-странице. Для это используем команду `htpasswd`, которая создаст нового пользователя (ключ `-c` – создать) с именем `elbrus`. Имя пользователя здесь и в строке `Require user elbrus` должно совпадать. Листинг выполнения команды представлен на рисунке 16. Полная команда будет выглядеть так

```
htpasswd -c /var/www/site1/htpasswd test
```

```
[root@alt218 site1]# htpasswd -c /var/www/site1/htpasswd test
New password:
Re-type new password:
Adding password for user test
```

Рисунок 16. Создание пароля для пользователя `test`

Последний шаг – дописать `AllowOverride all` в файл `/etc/httpd2/conf/sites-available/site1.conf`, строку допишем после `Include conf/include/Directory_html_default.conf` (рисунок 17). С помощью этой строки веб-сервер переопределяет стандартные конфигурационные данные и подключает файл `.htaccess`. Для записи воспользуемся редактором `nano`

```
nano /etc/httpd2/conf/sites-available/site1.conf
```

```
GNU nano 8.0        /etc/httpd2/conf/sites-available/site1.conf
        <Directory "/var/www/site1">
                # Summary: Configure for html documents in DocumentRoot
                # Requires: Directory-html
                Include conf/include/Directory_html_default.conf
                AllowOverride all
        </Directory>
```

Рисунок 17. Фрагмент файла `site1.conf`

Для вступления в силу всех изменений и подключенных модулей, необходимо выполнить команду

```
systemctl restart httpd2
```

О правильности выполненного задания можно судить по изменённой форме входа на веб-страницу. Теперь вместо отображения строки `Test`, предлагается ввести имя пользователя и пароль (рисунок 18).

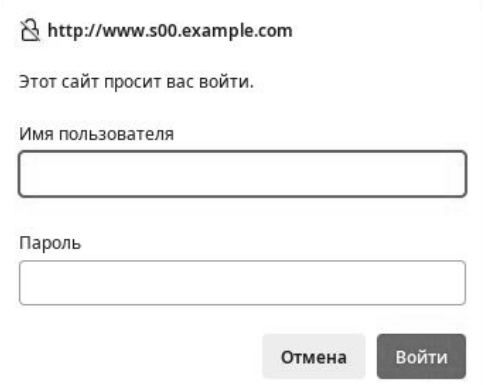


Рисунок 18. Меню аутентификации на сайте

После аутентификации веб-страница выдаст `Hello Elbrus!`, как и в первом пункте задания.

4.4. Настройка HTTPS доступа

Для работы с https необходимо установить модуль `apache_mod.ssl`, проверим, не установлен ли этот модуль, выполнив следующую команду

```
which apache2-mod_ssl
```

Если модуль не установлен (рисунок 19), то установим его, введя команду (рисунок 20)

```
apt-get install apache2-mod_ssl
```

```
[root@alt218 site0]# which apache2-mod_ssl
which: no apache2-mod_ssl in (/root/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local
/sbin:/usr/local/bin:/usr/games)
```

Рисунок 19. Проверка установленного модуля ssl

```
[root@alt218 site0]# apt-get install apache2-mod_ssl
Чтение списков пакетов... Завершено
Построение дерева зависимостей... Завершено
Следующие НОВЫЕ пакеты будут установлены:
  apache2-mod_ssl
0 будет обновлено, 1 новых установлено, 0 пакетов будет удалено и 74 не будет об
новлено.
Необходимо получить 234kB архивов.
После распаковки потребуется дополнительно 626kB дискового пространства.
Получено: 1 http://elbrus.ivk.ru p11_e2k/branch/e2kv5/classic apache2-mod_ssl 1:
2.4.65-alt1@1755396431 [234kB]
Получено 234kB за 0s (1155kB/s).
Совершаем изменения...
Подготовка...                          ################################# [100%]
Обновление / установка...
1: apache2-mod_ssl-1:2.4.65-alt1       ################################# [100%]
Завершено.
```

Рисунок 20. Установка модуля ssl для apache2

Для доступа по HTTPS перейдём в директорию, где находится файл `site1.conf` и создадим новый файл `site1_https.conf`, скопировав его из стандартного `default_https.conf`. Результат выполнения команды представлен на рисунке 21.

```
cd /etc/httpd2/conf/sites-available
cp default_https.conf site1_https.conf
```

Название `site1` должно совпадать с тем, что было создано для обычной веб-страницы в пункте 4.2

```
[root@alt218 site0]# cd /etc/httpd2/conf/sites-available/
[root@alt218 sites-available]# cp default_https.conf site1_https.conf
[root@alt218 sites-available]#
```

Рисунок 21. Копирование содержимого файла

Откроем файл через редактор `nano` (можно использовать любой другой текстовый редактор, например, `pluma`) и изменим следующие строки (рисунок 22):

```
DocumentRoot “/var/www/html” →
DocumentRoot “/var/www/site1”

ServerName www.example.com:443 →
ServerName www.s00.example.com:443

SSLCertificateFile
“/var/lib/ssl/certs/httpd2.cert” →
SSLCertificateFile “/var/www/site1/ssl.cert”

SSLCertificateKeyFile
“/var/lib/ssl/private/httpd2.key” →
SSLCertificateKeyFile “/var/www/site1/ssl.key”
```



Рисунок 22. Строки, которые необходимо изменить

Активируем модуль для работы с ssl и порт для https запросов (рисунок 23)

```
a2enmod ssl
a2enport https
```

```
[root@alt218 sites-available]# a2enmod ssl
Module ssl installed;
        run service httpd2 condreload to fully enable.
[root@alt218 sites-available]# a2enport https
Port config https installed;
        run service httpd2 condreload to fully enable.
[root@alt218 sites-available]#
```

Рисунок 23. Активация модуля ssl и порта https

Наконец, создадим самоподписанный ssl сертификат

```
openssl req -new -x509 -days 365 -nodes -out
/var/www/site1/ssl.cert -keyout /var/www/site1/ssl.key
```

Расшифруем команду выше:

`req` – создание запроса на подпись сертификата;
`-new` – создание нового сертификата;
`-x509` – создание самостоятельно подписанного сертификата;
`-days 365` – срок действия сертификата;
`-nodes` – не создавать кодовую фразу для защиты сертификата;
`-out <файл>` – имя файла создаваемого сертификата;
`-keyout <файл>` – имя файла создаваемого приватного ключа.

Процесс создания сертификата отображён на рисунке 24.

```
[root@alt218 sites-available]# openssl req -new -x509 -days 365 -nodes -out /var
/www/site1/ssl.cert -keyout /var/www/site1/ssl.key
.......+...+..+.++++++++++++++++++++++++++++++++++++++++*..+....+..+...+.......+.
.....+...+......+.....+..........+...+...+...+..+.......+...+..+..........+..+..
.......+..........+.....+......++++++++++++++++++++++++++++++++++++++++*.....+...
...........+...+...+..................+.+.........+..+.......+..+.+.............
....+.+..+.......+..+.+...................+.........+..+...+.+..+.......+..+.....
.....+.........+.....+.......+.........+.....+.+..+.+......+...................+.
.....+.....+....+.......................+............+..++++++
.+......+......+...+......+......+...........++++++++++++++++++++++++++++++++++++
++++*.+.++++++++++++++++++++++++++++++++++++++*.+...+..+.........+.+..+.......+
........+...+....+...+..............+........+...+......+.+..+.+...........+...
+.......+..............+..........+...........+....+..................+...+......
.+.....+...+....+..+...+...+............+............+............+...........+
.+.....+....+..+.......+...+.........+.....+......+.........+.+.........+.......
.+.+.....+....+......+..+....+.....+.+........+......+...+...+..................
+.+.....+.+.....+...+....+......+......+...+..+................+..+.......+...+.
.+.+..+......+.......+...+..+.........+...........+...+..+................+.+..+..
........+........+....+.........+.....+................+.+.....+....+.........+..
+...+..........+...........+.......+..................+...+.......+.....+.+.....+
.+...+...........+......+......+...+......+......+....+...+.........+.......+....
..+.....................+.....................+.+..+.+..............+......+.....
..+..+............+.+............+..+..........+.....+......+............+...+..
```

Рисунок 24. Процесс создания сертификата

Для вступления в силу всех изменений и подключенных модулей, необходимо выполнить команду

```
systemctl restart httpd2
```

О правильности выполнения можно судить, зайдя в `firefox` и введя адрес `https://www.s00.example.com` (рисунок 25). Обратите внимание, заходить нужно именно на адрес `https` вместо `http`.

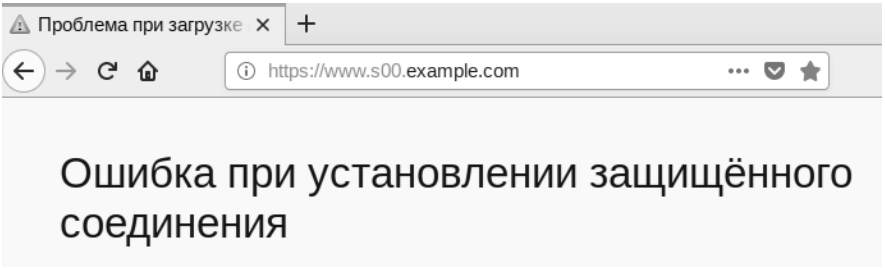


Рисунок 25. Результат установления защищённого соединения

Подключиться к веб-сайту с нашим самоподписанным сертификатом веб-браузер не разрешит. Но подключение по протоколу `https` состоялось, что говорит о правильности выполнения работы.

Практикум № 11
Использование пакетного менеджера RPM

1. ЦЕЛЬ РАБОТЫ

Научиться пользоваться пакетным менеджером RPM.

2. ЗАДАНИЕ НА РАБОТУ

1. Определите, где находится исполняемый файл, который запускает текстовый редактор `Pluma`.
2. Определите, какому пакету принадлежит исполняемый файл `pluma`.
3. Определите, какие файлы входят в этот пакет.
4. Определите версию, лицензию этого пакета, автора и дату сборки.
5. Определите, какие файлы и каким образом были изменены с момента установки пакета `setup`.
6. Вычислите общее количество пакетов в системе.
7. Определите, от каких файлов и пакетов зависит пакет `ipcalc`.

3. ТЕРМИНЫ И ИСПОЛЬЗУЕМЫЕ КОМАНДЫ

Используемые команды

`which утилита` – команда, обнаруживающая исполняющий файл для указанной утилиты.

`ll` – разновидность команды `ls`, с выводом полной информации о файлах.

`rpm` – команда для работы с пакетами. Список некоторых флагов:

`-i путь` – установить пакет;

`-u` – обновить пакет;

`-e` – удалить пакет;

`-ql` – выводит список файлов, входящих в пакет;

`-qi` – информация о пакете;

`-qa` – выводит список всех установленных пакетов в системе;

`-qf` – выводит, какому пакеты принадлежит файл;

`-V` – проверка целостности файла, выводит все изменения, которые произошли с файлом после установки пакета;

`-p` – выводит путь до файла, если он не установлен.

4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

4.1. Определение местонахождения исполняемого файла

Чтобы определить, где находится исполняемый файл, который запускает текстовый редактор `Pluma`. Нам потребуется команда

```
which pluma
```

Затем проверим, является ли файл исполняемым, для этого перейдем в данный каталог, используя команду

```
cd /usr/bin/
```

Далее, выполним команду, которая выведет все файлы из данной папки с указанием прав использования для различных групп пользователей и обернём её в команду-фильтр `grep`, чтобы вывести только то, что соответствует редактору `pluma`

```
ll | grep pluma
```

Если в правах файла находится символ `x`, то файл является исполняемым. Результат выполнения команд, описанных выше, приведён на рисунке 1.

```
[user@alt218 Рабочий стол]$ which pluma
/usr/bin/pluma
[user@alt218 Рабочий стол]$ cd /usr/bin/
[user@alt218 bin]$ ll | grep pluma
-rwxr-xr-x  1 root root       1580000 ноя  7 16:25 pluma
```

Рисунок 1. Определение прав файла `pluma`

Если же символа нет, то необходимо перейти в режим суперпользователя, и набрать команду

```
chmod u+x /usr/bin/pluma
```

Выполнив её, файл `pluma` станет исполняемым для после чего необходимо вновь использовать команду `ll | grep pluma` (рисунок 2)

```
[root@alt218 ~]# chmod u+x /usr/bin/pluma
[root@alt218 ~]# cd /usr/bin/
[root@alt218 bin]# ll | grep pluma
-rwxr-xr-x  1 root root       1580000 дек 23 22:42 pluma
```

Рисунок 2. Изменение прав файла `pluma`

4.2. Определение принадлежности пакету

Чтобы определить к какому пакету принадлежит исполняемый файл `pluma`, наберем следующую команду (рисунок 3)

```
rpm -qf /usr/bin/pluma
```

```
[user@alt218 bin]$ rpm -qf /usr/bin/pluma
mate-text-editor-1.28.0-alt1.e2kv5
```

Рисунок 3. Определение принадлежности пакету

4.3. Определение файлов, входящих в пакет

Чтобы определить какие файлы входят в этот пакет, наберем команду (рисунок 4)

```
rpm -ql название пакета
```

```
/usr/share/pluma
/usr/share/pluma/icons
/usr/share/pluma/icons/pluma-plugin.png
/usr/share/pluma/plugins
/usr/share/pluma/plugins/docinfo
/usr/share/pluma/plugins/docinfo/docinfo.ui
/usr/share/pluma/plugins/externaltools
/usr/share/pluma/plugins/externaltools/tools
/usr/share/pluma/plugins/externaltools/tools/build
/usr/share/pluma/plugins/externaltools/tools/open-terminal-here
```

Рисунок 4. Определение файлов, входящих в пакет

4.4. Определение сведений о пакете

Для определения версии, лицензии этого пакета, автора и даты сборки, воспользуемся командой (рисунок 5)

```
rpm -qi название пакета
```

```
[user@alt218 bin]$ rpm -qi mate-text-editor-1.28.0-alt1.e2kv5
Name        : mate-text-editor
Epoch       : 1
Version     : 1.28.0
Release     : alt1
Architecture: e2kv5
Install Date: Пт 07 ноя 2025 16:25:13
Group       : Graphical desktop/MATE
Size        : 28792916
License     : GPLv3+
Signature   : RSA/SHA1, Чт 29 фев 2024 10:47:24, Key ID cea7f56e5689c9f0
Source RPM  : mate-text-editor-1.28.0-alt1.src.rpm
Build Date  : Чт 29 фев 2024 10:47:07
Build Host  : mike-sisyphus_e2k.hasher.altlinux.org
Relocations : (not relocatable)
Packager    : Valery Inozemtsev <shrek@altlinux.ru>
Vendor      : ALT Linux Team
URL         : http://mate-desktop.org/
Summary     : Text editor for the MATE desktop
Description :
pluma is a small, but powerful text editor designed specifically for
the MATE desktop. It has most standard text editor functions and fully
supports international text in Unicode. Advanced features include syntax
highlighting and automatic indentation of source code, printing and editing
of multiple documents in one window.

pluma is extensible through a plugin system, which currently includes
support for spell checking, comparing files, viewing CVS ChangeLogs, and
adjusting indentation levels.
```

Рисунок 5. Определение сведений о пакете

4.5. Определение изменений в файлах

Чтобы определить какие файлы и каким образом были изменены с момента установки пакета `setup`, нам потребуется команда (рисунок 6)

```
rpm -V setup
```

```
[user@alt218 bin]$ rpm -V setup
S.?....T.  c /etc/securetty
```

Рисунок 6. Определение изменений в файлах

4.6. Определение количества пакетов в системе

Для определения общего количества пакетов, установленных в системе, используем команду `rpm -qa`, обернув её в команду для подсчёта строк `wc -l` (рисунок 7)

```
rpm -qa | wc -l
```

```
[user@alt218 bin]$ rpm -qa | wc -l
2195
```

Рисунок 7. Количество установленных пакетов

4.7. Определение зависимости пакета

Для проверки от каких файлов и пакетов зависит пакет `ipcalc`, в начале, потребуется узнать путь до исполняемого файла, использовав команду

```
which ipcalc
```

Для определения списка пакетов, от которых зависит наш пакет (без которых его работа невозможна), используем команду

```
rpm –qR ipcalc
```

Результат выполнения команды представлен на рисунке 8.

```
[user@alt218 bin]$ which ipcalc
/usr/bin/ipcalc
[user@alt218 bin]$ rpm -qR ipcalc
/usr/bin/perl
rpmlib(PayloadIsLzma)
```

Рисунок 8. Список пакетов, от которых зависит пакет `ipcalc`

Практикум № 12
Удалённый доступ по протоколу VNC

1. ЦЕЛЬ РАБОТЫ

Изучить сетевой протокол VNC. Определить основные этапы и особенности использования протокола. Настройка удаленного доступа по протоколу VNC.

2. ЗАДАНИЕ НА РАБОТУ

1. Настройте VNC сервер таким образом, чтобы ваш сосед мог подключиться к новому сеансу на вашей машине с правами только на просмотр. Зафиксируйте соответствующие команды.
2. Зайдите в тот же сеанс, что и ваш сосед, запустите некоторые приложения. Проконтролируйте, что ваш сосед, видит тот же сеанс, что и вы. Зафиксируйте команды.

3. ТЕРМИНЫ И ИСПОЛЬЗУЕМЫЕ КОМАНДЫ

Термины

Virtual Network Computing (VNC) — система удалённого доступа к рабочему столу компьютера, использующая протокол RFB (англ. Remote FrameBuffer, удалённый кадровый буфер). Управление осуществляется путём передачи нажатий клавиш на клавиатуре и движений мыши с одного компьютера на другой и ретрансляции содержимого экрана через компьютерную сеть.

Используемые команды

`vncserver` – команда, позволяющая создать vnc-сервер удаленного доступа. При выполнении команды потребуется ввести два пароля. Первый пароль нужен для полного доступа с возможностью управления. Второй пароль для доступа в качестве зрителя.

`vncviewer ip:display` – команда для подключения к указанному дисплею vnc-сервера с указанным ip.

4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

4.1. Создание vnc-сервера и подключение к нему

Для активации vnc-сервера зайдём в режим суперпользователя

```
su -
```

Далее, введём команду

```
which vncserver
```

Если пакет не удалось обнаружить, то установим его с помощью команды

```
apt-get install tigervnc-server
```

Далее выполним команду `vncserver`, после чего введём первый и второй пароли (подробнее о паролях в списке использованных команд)

```
vncserver
```

Результат выполнения команды представлен на рисунке 1.

```
[root@alt218 ~]# vncserver

You will require a password to access your desktops.

Password:
Verify:
Would you like to enter a view-only password (y/n)? y
Password:
Verify:

New 'alt218:1 (root)' desktop is alt218:1

Creating default startup script /root/.vnc/xstartup
Creating default config /root/.vnc/config
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/alt218:1.log
```

Рисунок 1. Активация vnc-сервера

Для подключения к текущему сеансу, определим ip компьютера, введя команду

```
ip a
```

Для примера, возьмём, что `ip - 10.10.21.218`.

На соседнем компьютере подключимся к созданному vnc-серверу

```
vncviewer 10.10.21.218:1
```

4.2. Запуск приложения на удалённом компьютере

Для начала создадим новый сеанс на компьютере с включённым vnc-сервером, используя команду (рисунок 2)

```
vncserver –list –DISPLAY:2
```

```
[root@alt218 ~]# vncserver -list -DISPLAY:2

TigerVNC server sessions:

X DISPLAY #     PROCESS ID
```

Рисунок 2. Список активных vnc-серверов

Подключимся на данной машине и на удалённой машине к вновь созданному сеансу, используя на первой машине пароль к полному доступу, а на второй к доступу в качестве зрителя

```
vncviewer 10.10.21.218:2
```

Активируем `pluma` и проследим, что на удалённой машине отобразился открытый текстовый редактор.

Практикум № 13
Статическая маршрутизация IPv4

1. ЦЕЛЬ РАБОТЫ

Данная работа предназначена для изучения статической маршрутизации трафика по протоколу IPv4 в сети Интернет, получения практического опыта в работе с маршрутизацией трафика IPv4 в ОС Альт; работа с основными конфигурационными файлами сетевой подсистемы, настройка и управление сетевыми интерфейсами, умение работать с утилитами для диагностики сетевых соединений, работающих по протоколу IPv4.

2. ЗАДАНИЕ НА РАБОТУ

В работе используется топология сети, представленная на рисунке 1. Сеть содержит три подсети, работающих по протоколу IPv4. Маршрутизаторы R2 и R3 являются транзитными маршрутизаторами и осуществляют пересылку пакетов между подсетями 1 и 3, а также выполняют роль шлюзов для маршрутизаторов R1 и R4. Маршрутизаторам R1, R2, R3 и R4 соответствуют рабочие станции под номерами PC1, PC2, PC3 и PC4 соответственно. Адресация в моделируемой сети осуществляется следующим образом:

- Подсеть 1: 10.10.$X(N_{min})$.0/24 или 10.10.X1.0/24
- Подсеть 2: 10.10.$X(N_{min}+1)$.0/24 или 10.10.X2.0/24
- Подсеть 3: 10.10.$X(N_{min}+2)$.0/24 или 10.10.X3.0/24

где $N_{min} = 1$ – наименьший номер из четырех рабочих станций;

X = номер ряда с рабочими станциями, для примера возьмём равное 1.

Наименование интерфейсов указано для примера, на практике оно может отличаться. В нашем же случае `eth0 - enp1s1f0`, а `eth1 - enp1s1f1`.

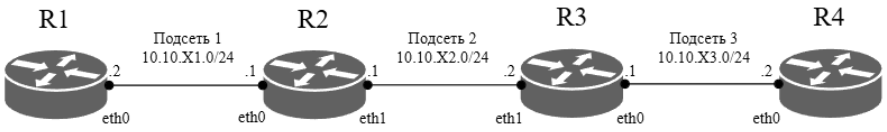


Рисунок 1. Схема сети лабораторной работы

Основные задачи лабораторной работы:

- Настроить все сетевые интерфейсы машин в соответствии со схемой лабораторной сети;

- Сделать маршрутизаторы R1 и R2 шлюзами по умолчанию для маршрутизаторов R1 и R4, прописав на них соответствующие маршруты;

- На транзитных маршрутизаторах R1 и R2 настроить пересылку пакетов между интерфейсами и прописать маршруты в удаленные подсети;

- Проверить работу сети с помощью утилит диагностики сетевых соединений: ping, traceroute.

3. ТЕРМИНЫ И ИСПОЛЬЗУЕМЫЕ КОМАНДЫ

Используемые команды

`cd` – (Change directory) команда перемещения между директориями;

`ls` – (list) команда просмотра содержимого директории;

`nano` – текстовый редактор для создания и редактирования текстовых файлов;

`ip` – утилита для настройки и управления сетевыми интерфейсами:

`ip address` – отображение информации о сетевых интерфейсах;

`ip route` – отображение информации о маршрутах;

`ip neighbor` – отображение ARP таблицы устройства.

`grep` – утилита для поиска информации;

`ping` – утилита для проверки сетевых соединений;

`traceroute` – утилита для трассировки маршрута в сети;

`mtr` – утилита для трассировки маршрута, показывает информацию в реальном времени.

4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

4.1. Настройка маршрутизатора R1

Для настройки маршрутизации в сети сначала необходимо настроить все сетевые интерфейсы на четырех маршрутизаторах.

Конфигурационные фалы всех сетевых интерфейсов находятся в директории /etc/net/ifaces. Переход в директорию осуществляется командой `cd`:

```
[user@alt219]$ cd /etc/net/ifaces/
```

Воспользуемся командой `ls` для того, чтобы просмотреть содержимое директории:

```
[user@alt219 ifaces]$ ls
```

Вывод команды `ls` покажет директории всех сетевых интерфейсов: enp1s1f0, enp1s1f1, enp1s1f2 и петлевого loopback-интерфейса lo.

Так как на R1 необходимо настроить только интерфейс enp1s1f0, то переходим в его директорию командой `cd` и выведем её содержимое командой `ls`:

```
[user@alt219 ifaces]$ cd enp1s1f0
[user@alt219 ifaces]$ ls
Options
```

В директории находится файл options - основной конфигурационный файл интерфейса enp1s1f0.

Для настройки интерфейса enp1s1f0 необходимо в конфигурационном файле /enp1s1f0/options изменить основные параметры интерфейса. Для редактирования конфигурационного файла нужны права суперпользователя root.

Вход в режим суперпользователя осуществляется командой `su -`, после чего требуется ввести пароль суперпользователя:

```
[user@alt219 enp1s1f0]$ su -
```

Отредактировать конфигурационный файл можно открыв его с помощью редактора nano:

```
[root@alt219]# nano /etc/net/ifaces/enp1s1f0/options
```

При стандартной настройке системы все сетевые интерфейсы управляются автоматически через Network Manager, необходимо включить конфигурирование интерфейсов вручную с помощью etcnet. Для этого в открывшемся окне редактора необходимо изменить основные параметры:

`BOOTPROTO=static` – использовать статически заданный IP-адрес;

`NM_CONTROLLED=no` – отключаем автоматическое управление интерфейсом через Network Manager;

`DISABLED=no` – включаем управление интерфейсом через etcnet;

`ONBOOT=yes` – включать интерфейс при загрузке системы;

`CONFIG_IPV4=yes` – режим работы с IPv4.

После всех настроек используем Ctrl+O чтобы сохранить изменения и Ctrl+X чтобы выйти из файла конфигурации.

Далее необходимо назначить статический IP-адрес на интерфейс enp1s1f0. Для этого в /etc/net/ifaces/ enp1s1f0/ нужно создать файл с именем ipv4address и в него вписать необходимый IP-адрес в формате <x.x.x.x/маска>

Используя утилиту `nano`, чтобы создать файл с именем ipv4address и сразу перейти к его редактированию:

```
[root@alt219 enp1s1f0]# nano ipv4address
```

`10.10.11.2/24` – прописываем IP-адрес в созданном файле ipv4address

Сохраняем Ctrl+O и выходим из редактирования Ctrl+X.

Так как R2 является шлюзом по умолчанию для маршрутизатора R1, то на R1 необходимо прописать маршрут, по умолчанию отправляющий все пакеты на маршрутизатор R2. Для этого в /etc/net/ifaces/enp1s1f0 создается файл с именем ipv4route и в него прописывается маршрут по умолчанию:

```
[root@alt219 enp1s1f0]# nano ipv4route
```

`default via 10.10.11.1` – маршрут по умолчанию на R2

Чтобы изменения применились, необходим перезапуск сетевой службы:

```
[root@alt219 enp1s1f0]#service network restart
```

Выход из суперпользователя осуществляется командой exit или Ctrl+D.

Проверка настройки IP-адреса командой `ip address` или сокращенно `ip a`:

```
[user@alt219]$ ip a
2: enp1s1f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group default qlen 1000
    link/ether 98:a7:b0:00:de:a0 brd ff:ff:ff:ff:ff:ff
    inet 10.10.11.2/24 brd 10.10.11.3 scope global enp1s1f0
       valid_lft forever preferred_lft forever
```

Первая запись `link/ether 00:0c:29:df:4e:8a brd ff:ff:ff:ff:ff:ff` – это физические адреса (MAC адреса) интерфейса enp1s1f0, где `ff:ff:ff:ff:ff:ff` - широковещательный MAC адрес, необходимый для работы протокола преобразования адресов ARP (Address Resolution Protocol).

Вторая запись `inet 10.10.11.2/24 brd 10.10.11.3 scope global enp1s1f0` это настроенный нами интерфейс enp1s1f0, где `inet`

10.10.11.2/24 – сам IP-адрес интерфейса, а brd 10.10.11.3 – широковещательный адрес для нашей подсети, рассчитанный автоматически в соответствии с указанной маской.

Проверка таблицы маршрутизации на R1. Используется команда ip route или сокращенно ip r. Так как в работе используются IP-адреса подсетей, начинающиеся с 10.10, то для уточнения поиска по таблице маршрутизации дополним запрос ip r командой grep 10.10:

```
[user@alt219]$ ip r | grep 10.10
default via 10.10.11.1 dev enp1s1f0
10.10.11.0/24 dev enp1s1f0 proto kernel scope link src 10.10.11.2
```

В таблице маршрутизации есть две записи. Первая запись – наш прописанный маршрут по умолчанию на маршрутизатор R2. Вторая запись – непосредственно подключенная напрямую к интерфейсу enp1s1f0 сеть 10.10.11.0/24.

4.2. Настройка маршрутизатора R2

Для настройки сетевых интерфейсов на маршрутизаторе R2 редактируются их конфигурационные файлы в директории /etc/net/ifaces/.

Для интерфейса enp1s1f0 в /etc/net/ifaces/enp1s1f0 редактируется файл options:

```
[root@alt218]# nano /etc/net/ifaces/enp1s1f0/options
BOOTPROTO=static
NM_CONTROLLED=no
DISABLED=no
ONBOOT=yes
CONFIG_IPV4=yes
```

Настройка IPv4-адреса на enp1s1f0:

```
[root@alt218]# nano /etc/net/ifaces/enp1s1f0/ipv4address
10.10.11.1/24
```

Для интерфейса enp1s1f1 в /etc/net/ifaces/enp1s1f1 редактируется файл options:

```
[root@alt218]# nano /etc/net/ifaces/enp1s1f1/options
BOOTPROTO=static
```

```
NM_CONTROLLED=no
DISABLED=no
ONBOOT=yes
CONFIG_IPV4=yes
```

Интерфейсу enp1s1f1 на R2 принадлежит адрес из второй подсети. Настройка IPv4-адреса на enp1s1f1:

```
[root@alt218]# nano /etc/net/ifaces/enp1s1f1/ipv4address
10.10.12.1/24
```

Так как R2 является транзитным маршрутизатором, у него должна быть возможность перенаправлять пакеты из одного интерфейса в другой. Для этого необходимо включить форвардинг пакетов на уровне ядра системы.

В файле /etc/net/sysctl.conf нужно отредактировать строку: `net.ipv4.ip_forward=0`, в которой 0 необходимо заменить на 1.

```
[root@alt218]# nano /etc/net/sysctl.conf
net.ipv4.ip_forward=1
```

После произведенных настроек интерфейсов и форвардинга пакетов, необходимо перезапустить сетевую службу и проверить настроенные IP-адреса:

```
[root@alt218]# service network restart
[root@alt218]# ip a
2: enp1s1f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group default qlen 1000
    link/ether 98:a7:b0:00:dd:60 brd ff:ff:ff:ff:ff:ff
    inet 10.10.11.1/24 brd 10.10.11.3 scope global enp1s1f0
       valid_lft forever preferred_lft forever
3: enp1s1f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group default qlen 1000
    link/ether 98:a7:b0:00:dd:61 brd ff:ff:ff:ff:ff:ff
    inet 10.10.12.1/24 brd 10.10.12.3 scope global enp1s1f1
       valid_lft forever preferred_lft forever
```

После настроек необходима проверка сетевой связности между подсетями 1 и 2. Для этого с маршрутизатора R1 проверяется доступность подсети 10.10.12.0/24, находящейся за интерфейсом enp1s1f1 маршрутизатора R2.

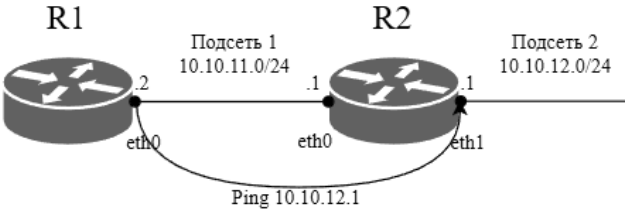


Рисунок 2. Проверка связности сети командой ping

Чтобы проверить работу сети IPv4, используется утилита ping, работающая по протоколу ICMP.

На R1 выполняется команда `ping` и указывается адрес интерфейса enp1s1f1 маршрутизатора R2:

```
    [root@alt219 ~]# ping 10.10.12.1 -c 3
    PING 10.10.12.1 (10.10.12.1) 56(84) bytes of data.
    64 bytes from 10.10.12.1: icmp_req=1 ttl=64
time=0.099 ms
    64 bytes from 10.10.12.1: icmp_req=2 ttl=64
time=0.095 ms
    64 bytes from 10.10.12.1: icmp_req=3 ttl=64
time=0.137 ms
    --- 10.10.12.1 ping statistics ---
    3 packets transmitted, 3 received, 0% packet loss,
time 1998ms
    rtt min/avg/max/mdev = 0.095/0.110/0.137/0.020 ms
```

Ключ –c (count) задает количество посылаемых ICMP запросов. Интерфейс enp1s1f1 маршрутизатора R2 доступен, ICMP пакеты успешно возвращаются.

После того как взаимодействие между R1 и R2 установлено, устройства добавят друг друга в свою таблицу соответствия сетевых и физических адресов – ARP таблицу, установив соответствие между MAC-адресами и IP-адресами интерфейсов маршрутизаторов.

Проверка ARP таблицы на маршрутизаторе R1 осуществляется командой `ip n (neighbor)`. Так как нам необходимы только адреса, начинающиеся с 10.10, то дополним запрос `ip n` командой `grep 10.10`:

```
[root@alt218 ~]# ip n | grep 10.10
10.10.11.1 dev enp1s1f0 lladdr 98:a7:b0:00:dd:60
REACHABLE
```

ARP таблица содержит записи, устанавливающие соответствие между MAC-адресами и IP-адресами соседнего узла, т.е. маршрутизатора R2.

В ARP таблице маршрутизатора R2 также будет находиться IP-адрес и соответствующий ему MAC-адрес маршрутизатора R1:

```
[root@alt218 ~]# ip n | grep 10.10
10.10.11.2 dev enp1s1f0 lladdr 98:a7:b0:00:de:a0
REACHABLE
```

- Отметка `REACHABLE` показывает, что запись о соответствии появилась недавно, и IP-адрес `10.10.11.2` на R1 доступен.
- Отметка `STALE` означает, что запись о доступности соседнего устройства устарела.
- Отметка `UNREACHABLE` означает недоступность соседнего устройства по этому IP-адресу.

Поскольку R2 является транзитным маршрутизатором между подсетями 1 и 3, у него должен быть маршрут в третью подсеть. Взаимосвязь с подсетью 3 происходит через интерфейс enp1s1f1, поэтому маршрут добавляется в конфигурационном файле этого интерфейса:

```
[root@alt218]# nano /etc/net/ifaces/enp1s1f1/ipv4route
10.10.13.0/24 via 10.10.12.2
```

Запись добавляет маршрут в подсеть 3 через интерфейс enp1s1f1 маршрутизатора R3.

На R2, после настройки маршрута, необходима перезагрузка сетевой службы и проверка таблицы маршрутизации. Так же дополняем запрос `ip r` командой `grep 10.10`:

```
[root@alt218]# service network restart
[root@alt218 ~]# ip r | grep 10.10
10.10.11.0/24 dev enp1s1f0 proto kernel scope link
src 10.10.11.1
```

```
    10.10.12.0/24 dev enp1s1f1 proto kernel scope link
src 10.10.12.1
    10.10.13.0/24 via 10.10.12.2 dev enp1s1f1
```

В таблице маршрутизации первые две записи означают подключенные напрямую сети к интерфейсам enp1s1f0 и enp1s1f1. Последняя запись – прописанный маршрут в подсеть 3 с интерфейса enp1s1f1 через транзитный маршрутизатор R3.

4.3. Настройка маршрутизатора R3

Для настройки сетевых интерфейсов на маршрутизаторе R3 редактируются их конфигурационные файлы в директории /etc/net/ifaces/.

Для интерфейса enp1s1f1 в /etc/net/ifaces/enp1s1f1 редактируется файл options:

```
    [root@alt217]# nano /etc/net/ifaces/enp1s1f1/options
    BOOTPROTO=static
    NM_CONTROLLED=no
    DISABLED=no
    ONBOOT=yes
    CONFIG_IPV4=yes
```

Настройка IPv4-адреса на enp1s1f1:

```
    [root@alt217]# nano
/etc/net/ifaces/enp1s1f1/ipv4address
    10.10.12.2/24
```

Для интерфейса enp1s1f0 в /etc/net/ifaces/enp1s1f0 редактируется файл options:

```
    [root@alt217]# nano /etc/net/ifaces/enp1s1f0/options
    BOOTPROTO=static
    NM_CONTROLLED=no
    DISABLED=no
    ONBOOT=yes
    CONFIG_IPV4=yes
```

Интерфейсу enp1s1f0 на R3 принадлежит адрес из третьей подсети. Настройка IPv4-адреса на enp1s1f0:

```
[root@alt217]# nano
/etc/net/ifaces/enp1s1f0/ipv4address
10.10.13.1/24
```

Так как R3 является транзитным маршрутизатором, у него должна быть возможность перенаправлять пакеты из одного интерфейса в другой. Для этого необходимо включить форвардинг пакетов на уровне ядра системы.

В файле /etc/net/sysctl.conf нужно отредактировать строку: `net.ipv4.ip_forward=0`, в которой 0 необходимо заменить на 1.

```
[root@alt217]# nano /etc/net/sysctl.conf
net.ipv4.ip_forward=1
```

Аналогично транзитному маршрутизатору R2, у маршрутизатора R3 должен быть маршрут в первую подсеть. Взаимосвязь с подсетью 1 происходит через интерфейс enp1s1f1, поэтому маршрут добавляется в конфигурационном файле этого интерфейса:

```
[root@alt217]# nano /etc/net/ifaces/enp1s1f1/ipv4route
10.10.11.0/24 via 10.10.12.1
```

Запись добавляет маршрут в подсеть 1 через маршрутизатор R2.

После произведенных настроек интерфейсов и форвардинга пакетов, необходим перезапуск сетевой службы и проверка настроенных IP-адресов и маршрутов:

```
[root@alt217]# service network restart
[root@alt217]# ip address
2: enp1s1f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu
1500 qdisc pfifo_fast state UNKNOWN group default qlen 1000
    link/ether 98:a7:b0:00:de:20 brd ff:ff:ff:ff:ff:ff
    inet 10.10.13.1/24 brd 10.10.13.3 scope global
enp1s1f0
       valid_lft forever preferred_lft forever
3: enp1s1f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu
1500 qdisc pfifo_fast state UNKNOWN group default qlen 1000
    link/ether 98:a7:b0:00:de:21 brd ff:ff:ff:ff:ff:ff
    inet 10.10.12.2/24 brd 10.10.12.3 scope global
enp1s1f1
```

```
valid_lft forever preferred_lft forever
```

Таблица маршрутизации на R3. В таблице маршрутизации информация о подключенных напрямую подсетях, а также о том, что первая подсеть 10.10.11.0/24 доступна через маршрутизатор R2:

```
[root@alt217]# ip r | grep 10.10
10.10.11.0/24 via 10.10.12.1 dev enp1s1f1
10.10.12.0/24 dev enp1s1f1 proto kernel scope link src
10.10.12.2
10.10.13.0/24 dev enp1s1f0 proto kernel scope link src
10.10.13.1
```

После настроек необходима проверка сетевой связности между подсетями 1 и 3 командой `ping`. Для этого с маршрутизатора R1 проверяется доступность подсети 10.10.13.0/24, находящейся за интерфейсом enp1s1f0 маршрутизатора R3:

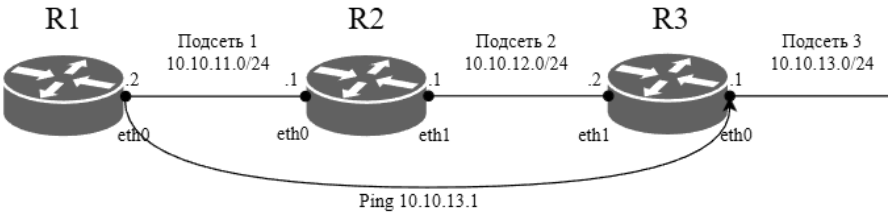


Рисунок 3. Проверка связности сети командой ping

```
[root@alt219 ~]# ping 10.10.13.1 -c 3
PING 10.10.13.1 (10.10.13.1) 56(84) bytes of data.
64 bytes from 10.10.13.1: icmp_req=1 ttl=63 time=0.261
ms
64 bytes from 10.10.13.1: icmp_req=2 ttl=63 time=0.130
ms
64 bytes from 10.10.13.1: icmp_req=3 ttl=63 time=0.133
ms
--- 10.10.13.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time
1998ms
rtt min/avg/max/mdev = 0.130/0.174/0.261/0.062 ms
```

ICMP пакеты успешно возвращаются, интерфейс enp1s1f0 маршрутизатора R3 доступен.

4.4. Настройка маршрутизатора R4

Для настройки сетевых интерфейсов на маршрутизаторе R4 редактируются их конфигурационные файлы в директории /etc/net/ifaces/.

Для интерфейса enp1s1f0 в /etc/net/ifaces/enp1s1f0 редактируется файл options:

```
[root@alt216]# nano /etc/net/ifaces/enp1s1f0/options
BOOTPROTO=static
NM_CONTROLLED=no
DISABLED=no
ONBOOT=yes
CONFIG_IPV4=yes
```

Настройка IPv4-адреса на enp1s1f0:

```
[root@alt216]# nano
/etc/net/ifaces/enp1s1f0/ipv4address
10.10.13.2/24
```

Так как R3 является шлюзом по умолчанию для маршрутизатора R4, то на R4 необходимо прописать маршрут, по умолчанию отправляющий все пакеты на маршрутизатор R3. Для этого в /etc/net/ifaces/ enp1s1f0 создается файл с именем ipv4route и в него прописывается маршрут по умолчанию:

```
[root@alt216 enp1s1f0]# nano ipv4route
default via 10.10.13.1 –  маршрут по умолчанию на R3
```

После произведенных настроек необходим перезапуск сетевой службы и проверка настроенных IP-адресов и маршрутов:

```
[root@alt216]# service network restart
[root@alt216]# ip address
2:  enp1s1f0:  <BROADCAST,MULTICAST,UP,LOWER_UP>  mtu
1500 qdisc pfifo_fast state UNKNOWN group default qlen 1000
       link/ether 98:a7:b0:00:df:20 brd ff:ff:ff:ff:ff:ff
       inet  10.10.13.2/24  brd  10.10.13.3  scope  global
enp1s1f0
          valid_lft forever preferred_lft forever
```

Проверка маршрута по умолчанию на R4 с помощью команды `ip route`:

```
[root@alt216 ~]# ip r | grep 10.10
default via 10.10.13.1 dev enp1s1f1
10.10.13.0/24 dev enp1s1f1 proto kernel scope link src 10.10.13.2
```

В таблице маршрутизации в дополнении к непосредственно подключенной напрямую сети, появился маршрут по умолчанию на R3, прописанный первой строкой.

4.5. Проверка работы построенной сети

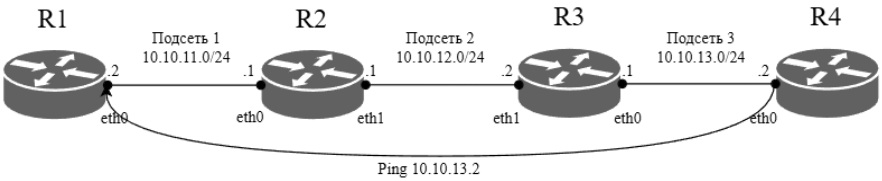


Рисунок 4. Проверка работы сети командой ping

Проверка работы настроенной сети осуществляется командой `ping` с маршрутизатора R1 на маршрутизатор R4:

```
[root@alt219 ~]# ping 10.10.13.2 -c 5
PING 10.10.13.2 (10.10.13.2) 56(84) bytes of data.
64 bytes from 10.10.13.2: icmp_req=1 ttl=62 time=0.349 ms
64 bytes from 10.10.13.2: icmp_req=2 ttl=62 time=0.219 ms
64 bytes from 10.10.13.2: icmp_req=3 ttl=62 time=0.183 ms
64 bytes from 10.10.13.2: icmp_req=4 ttl=62 time=0.185 ms
64 bytes from 10.10.13.2: icmp_req=5 ttl=62 time=0.194 ms
--- 10.10.13.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.183/0.226/0.349/0.062 ms
```

Отправленные ICMP пакеты успешно возвращаются, маршрутизатор R4 доступен, настроенная сеть функционирует правильно.

Проверить каким путем проходят пакеты между подсетями можно с помощью утилиты traceroute:

```
[root@alt219 ~]# traceroute 10.10.13.2
traceroute to 10.10.13.2       , 30 hops max, 60 byte
packets
 1  gateway (10.10.11.1)          0.157 ms  0.115 ms
0.086 ms
 2  10.10.12.2 (10.10.12.2)       0.207 ms  0.133 ms
0.111 ms
 3  10.10.13.2 (10.10.13.2)       0.184 ms  0.178 ms
0.126 ms
```

Маршрутизатор R1, не зная маршрута до R4, отправляет свои пакеты на шлюз по умолчанию – маршрутизатор R2. Транзитные маршрутизаторы R2 и R3 знают о всех подсетях, и используя свои транзитные маршруты, перенаправляют пакеты между подсетями. Получив пакеты от R1, R2 отправляет их на R3, а R3 на R4. Так как R4 также не знает о подсети 10.10.11.0/24, в которой находится R1, он отправляет пакеты на свой шлюз по умолчанию – маршрутизатор R3, а тот уже через R2 на R1.

4.6. Утилита mtr для диагностики сетевых соединений

Путь прохождения пакетов можно также отследить с помощью утилиты mtr. Утилита mtr является аналогом traceroute и в реальном времени показывает потери и задержки пакетов на определенных участках сети:

```
[root@alt219]# mtr 10.10.13.2
My traceroute [v0.95] 2025-11-24T02:11:11+0300
Host                    Packets                    Pings
             loss%  Snt   Last   Avg  Best  Wrst   StDev
1. 10.10.11.1  0.0%    14   0.4    1.1   0.4   1.4    0.3
2. 10.10.12.2  0.0%    13   2.2    2.3   2.2   2.4    0.1
3. 10.10.13.2  0.0%    13   2.2    2.3   2.2   2.4    0.1
```

Loss% – процент потерь пакетов;

Snt – количество отправленных пакетов;

Last – время задержки последнего полученного пакета, мс;
Avg – среднее время задержки, мс;
Best – наименьшее зафиксированное время задержки, мс;
Wrst – наибольшее зафиксированное время задержки, мс;
StDev – среднеквадратичное отклонение времени задержки, мс.

Практикум № 14
Протокол динамической маршрутизации RIPv2

1. ЦЕЛЬ РАБОТЫ

Изучение особенностей работы протокола маршрутной информации RIPv2 (Routing Information Protocol) версии 2 с помощью отечественных программно-аппаратных средств. Изучение пакета quagga, реализующего протоколы маршрутизации, анализ его функционала в рамках настройки сети, работающей по протоколу динамической маршрутизации RIPv2.

2. ЗАДАНИЕ НА РАБОТУ

В работе используется топология сети, представленная на рисунке 1. Маршрутизаторам R1, R2, R3 и R4 соответствуют рабочие станции под номерами PC1, PC2, PC3 и PC4, соответственно. Адресация в моделируемой сети осуществляется следующим образом:

- Подсеть 1: $10.10.X(N_{min}).0/24$ или 10.10.X1.0/24
- Подсеть 2: $10.10.X(N_{min}+1).0/24$ или 10.10.X2.0/24
- Подсеть 3: $10.10.X(N_{min}+2).0/24$ или 10.10.X3.0/24
- Подсеть 4: $10.10.X(N_{min}+3).0/24$ или 10.10.X4.0/24

где $N_{min} = 1$ – наименьший номер из четырех рабочих станций;

X = номер ряда с рабочими станциями, для примера возьмём равное 1.

Наименование интерфейсов указано для примера, на практике оно может отличаться. В нашем же случае `eth0 – enp1s1f0`, а `eth1 – enp1s1f1`.

Последние цифры IP-адресов интерфейсов соответствуют номеру машины.

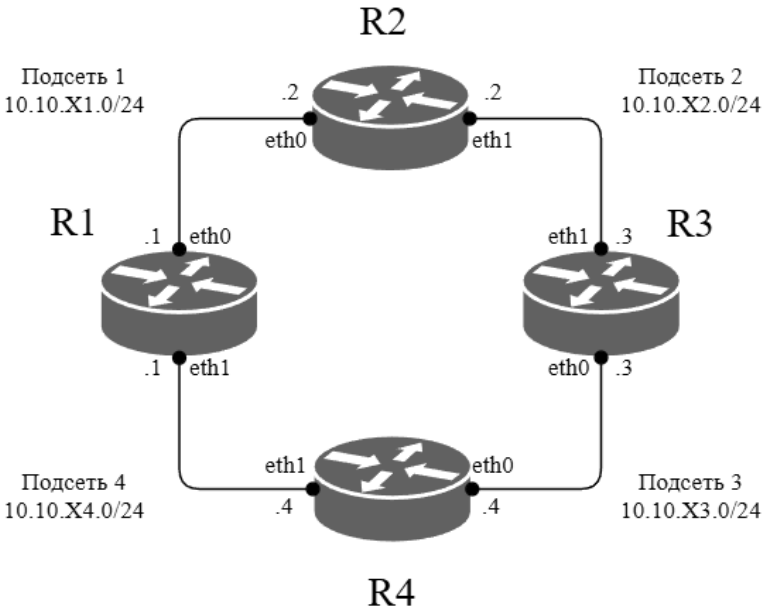


Рисунок 1. Схема лабораторной сети

Основные задачи работы:

- Подготовить машины к работе в сети и установить необходимые пакеты;
- Изменить имя устройств и настроить сетевые интерфейсы, задействованные в работе протокола RIPv2 с помощью демона zebra;
- С помощью демона ripd настроить работу протокола RIPv2 на всех устройствах, настроить анонсирование необходимых подсетей и проверить работу протокола с помощью анализатора трафика Wireshark;
- Проверить отказоустойчивость построенной сети RIPv2, смоделировав отказ канала связи между R1 и R2 путем отключения интерфейса enp1s1f0 на R1;
- Проверить изменение метрик маршрутов протокола RIPv2 при перестроении сетевой топологии из-за отключения канала связи между R1 и R2.

3. ТЕРМИНЫ И ИСПОЛЬЗУЕМЫЕ КОМАНДЫ

Используемые команды

`apt-get install` <имя пакета> – команда пакетного менеджера Apt, используемого в ОС Альт. Инициирует установку указанных пакетов;

`chown` – команда смены владельца (change owner), когда также необходимо сменить права вложенные файлы и папки, используется ключ -R (рекурсивно);

`nano` – текстовый редактор, также может создавать текстовые файлы;

`systemctl <start/stop>` – команда управления системными сервисами

`netstat` – команда отображения сетевых соединений системы. Основные ключи: -t (TCP-соединения), -u (UDP-соединения), -l (слушающие сокеты), -p (номер процесса в системе), -n (не переводить IP-адреса в доменные имена);

`grep` – утилита для поиска информации;

`ip` – утилита для настройки и управления сетевыми интерфейсами:

`ip address` – отображение информации о сетевых интерфейсах;

`ip route` – отображение информации о маршрутах;

`telnet` – утилита для подключения к сетевым узлам по протоколу telnet;

`ping` – утилита для проверки сетевых соединений;

`traceroute` – утилита для трассировки маршрута в сети.

Основные команды сетевых демонов quagga

`Show running config` – показать текущую конфигурацию устройства;

`Hostname` <имя> – смена имени устройства;

`Interface` <имя> – переход к конфигурированию интерфейса;

`Ip address` <IP-адрес/маска> – Настройка IP-адреса интерфейса;

`Router rip` – запуск протокола RIP на устройстве и переход к его конфигурированию;

`Version 2` – включение второй версии протокола RIP;

`Timers basic` <Update> <Timeout> <Flush> – настройка таймеров протокола RIPv2;

`Network` <сеть/маска> – настройка анонсирования заданной сети;

`Show ip route rip` – показать таблицу маршрутизации RIPv2;

`Write memory` – сохранение конфигурации устройства.

4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

4.1. Установка пакетов и подготовка машин к работе

Для установки пакета quagga, как и любых дополнительных пакетов, необходимы права суперпользователя root, поэтому переходим в режим работы от root командой su – и вводим пароль суперпользователя:

```
[user@alt219 ~]$ su -
Password:
```

Используя пакетный менеджер apt, командой `apt-get install` устанавливаем необходимые пакеты. Кроме пакета quagga, для выполнения лабораторной работы понадобятся пакеты telnet и wireshark. Используем ключ –y, чтобы автоматически подтверждать установку пакетов:

```
[root@alt219 ~]# apt-get install -y quagga telnet wireshark
```

Конфигурационные файлы пакета quagga находятся в директории /etc/quagga/, а логи – в /var/log/quagga/.

Чтобы пакет мог корректно работать, необходимо сменить права на файлы конфигурации и логов, которые он использует. Командой `chown` делаем пользователя quagga и его одноименную группу владельцами файлов:

```
[root@alt219 ~]# chown -R quagga:quagga /etc/quagga/
[root@alt219 ~]# chown -R quagga:quagga /var/log/quagga/
```

После того, как пакеты установлены, необходимо на всех машинах включить форвардинг пакетов на уровне ядра системы. Это необходимо для того, чтобы маршрутизаторы могли пересылать пакеты между своими интерфейсами.

Включение форвадинга производится редактированием конфигурационного файла системы sysctl.conf, находящегося в директории /etc/net/. В конфигурационном файле в строке net.ipv4.ip_forward=0 необходимо изменить 0 на 1:

```
[root@alt219 ~]# nano /etc/net/sysctl.conf
net.ipv4.ip_forward=1
```

Сохраняем изменения Ctrl+O и выходим с помощью Ctrl+X. После настройки нужно перезапустить сетевые сервисы системы командой

```
[root@alt219 ~]# service network restart
```

4.2. Подключение к демонам и работа с конфигурационными файлами пакета quagga

Для настройки динамической маршрутизации RIPv2 нужны управляющий демон zebra и демон протокола RIPv2 – ripd.

Запускаем необходимые демоны:

```
[root@alt219 ~]# systemctl start zebra
[root@alt219 ~]# systemctl start ripd
```

Настраивать сетевые демоны можно как редактируя их конфигурационные файлы, находящиеся в директории /etc/quagga/, так и подключаясь к их консоли по telnet.

После запуска демонов на нашей машине появятся сокеты (IP-адрес + порт), прослушивающие входящие соединения, с помощью которых и осуществляется подключение к консоли демонов.

Проверим появление необходимых сокетов командой `netstat`, так как демоны quagga используют порты, начинающиеся от 2601, то для точного поиска дополняем запрос командой `grep 260`:

```
[root@alt219 ~]# netstat -tlnp | grep 260
Proto       Local  Address    Foreign  Address    State
PID/Program name
tcp      127.0.0.1:2601      0.0.0.0:*    LISTEN     -
150338/zebra
tcp      127.0.0.1:2600      0.0.0.0:*    LISTEN     -
150338/zebra
tcp      127.0.0.1:2602      0.0.0.0:*    LISTEN     -
150387/ripd
```

На локальном хосте (127.0.0.1) демон ripd слушает порт 2602, а zebra – порт 2601.

Используем telnet для подключения к консоли zebra или ripd, указывается IP-адрес и порт необходимого демона:

`[root@alt219 ~]# telnet 127.0.0.1 2601` – подключение к zebra

`[root@alt219 ~]# telnet 127.0.0.1 2602` – подключение к ripd

Подключимся к zebra. Вводим пароль zebra (по умолчанию) и попадаем в его консоль:

```
[root@alt219 ~]# telnet 127.0.0.1 2601
Connected to 127.0.0.1.
User Access Verification
Password:
Router>
```

Для того чтобы начать настройку, сначала нужно войти в привилегированный режим командой `enable`. Привилегированный режим также защищен паролем (по умолчанию zebra):

```
Router> enable
Password:
Router#
```

Чтобы посмотреть текущую конфигурацию маршрутизатора, используется команда `show running-config`, или сокращенно `show run`:

```
Router# show running-config
```

Вывод команды покажет конфигурацию, записанную в файле /etc/quagga/zebra.conf, рассмотрим ее подробнее:

Имя хоста:

```
hostname Router
```

Зашифрованный пароль для подключения к нашему маршрутизатору:

```
password 8 LrjDz/a2KALVQ
```

Зашифрованный пароль для входа в привилегированный режим enable:

```
enable password 8 LrjDz/a2KALVQ
```

Команда, включающая шифрование паролей при просмотре конфигурации:

```
service password-encryption
```

Путь к файлу с логами:

```
log file /var/log/quagga/zebra.log
```

Список контроля доступа (ACL) с именем localhost, разрешающий доступ к консоли демона zebra только с локального хоста, и запрещающий все остальные соединения:

```
access-list localhost permit 127.0.0.1/32
access-list localhost deny any
```

Команды, указывающие применять список доступа localhost для входящих соединений по виртуальным терминальным линиям vty. Подключаясь по telnet, мы занимаем одну виртуальную терминальную линию vty:

```
line vty
```

```
access-class localhost
```

Внесение изменений в конфигурацию устройства производится в режиме глобального конфигурирования (config). Вход в режим осуществляется командой `configure terminal` или сокращённо `conf t`:

```
router# configure terminal
router(config)#
```

4.3. Настройка маршрутизатора R1

Для работы протокола RIPv2 на маршрутизаторе R1 с помощью демона zebra настраиваются все сетевые интерфейсы, задействованные в работе протокола RIPv2. Подключаемся к zebra и входим в режим глобального конфигурирования:

```
[root@alt219 ~]# telnet 127.0.0.1 2601
Trying 127.0.0.1...
Connected to 127.0.0.1.
User Access Verification
Password:
Router>
Router> enable
Router#
Router# configure terminal
Router(config)#
```

Изменим имя устройства:

```
Router(config)# hostname R1
```

В процессе работы протокола RIPv2 будут задействованы два интерфейса маршрутизатора: enp1s1f0 и enp1s1f1.

Настройка интерфейса enp1s1f0:

`R1(config)#interface enp1s1f0` – переход к конфигурированию интерфейса;

`R1(config-if)#ip address 10.10.11.1/24` – присваиваем адрес и маску подсети;

`R1(config-if)#exit` – выход из режима настройки интерфейса enp1s1f0.

```
R1(config)#
```

Настройка интерфейса enp1s1f1:

```
R1(config)#interface enp1s1f1ъ
R1(config-if)#ip address 10.10.14.1/24
R1(config-if)#exit
R1(config)#
```

Чтобы сохранить все произведенные настройки, нужно сохранить текущую конфигурацию zebra командой `write memory` или сокращенно `wr`:

```
R1(config)# write memory
Configuration saved to /etc/quagga/zebra.conf
R1(config)# exit
```

Вся конфигурация была перезаписана в /etc/quagga/zebra.conf.

Настройка демона zebra завершена, отключиться от демона можно с помощью комбинации Ctrl+D.

Далее на маршрутизаторе R1 необходимо настроить работу протокола RIPv2 с помощью демона ripd. Подключаемся к ripd по telnet, вводим пароль zebra (по умолчанию) и попадаем в его консоль:

```
[root@alt219 ~]# telnet 127.0.0.1 2602
Trying 127.0.0.1...
Connected to 127.0.0.1.
User Access Verification
Password:
router-RIP>
router-RIP> enable
router-RIP#
```

Командой `show running-config` можно посмотреть текущую конфигурацию, записанную в /etc/quagga/ripd.conf. Для внесения изменений в конфигурацию, переходим в режим глобального конфигурирования, используя команду `configure terminal`:

```
router-RIP# configure terminal
router-RIP(config)#
```

Изменим имя устройства:

```
router-RIP(config)# hostname R1-RIP
```

Теперь необходимо включить процесс RIP на маршрутизаторе:

`R1-RIP(config)# router rip` – включаем RIP на устройстве и переходим к его конфигурированию;

По умолчанию используется устаревшая первая версия протокола RIP, для включения версии 2 используем команду:

```
R1-RIP(config)# version 2
```

Далее необходимо прописать все подключенные сети, которые необходимо анонсировать маршрутизатору R1 по протоколу:

```
R1-RIP(config-router)# network 10.10.11.0/24
R1-RIP(config-router)# network 10.10.14.0/24
```

Чтобы увеличить скорость конвергенции сети при изменениях топологии, уменьшим время стандартных таймеров. Таймера Update с 30 до 10 секунд, таймера Timeout с 180 до 60 секунд и таймера Flush с 120 до 30 секунд:

```
R1-RIP(config-router)# timers basic 10 60 30
R1-RIP(config-router)# exit
```

Настройка ripd завершена. Сохраняем конфигурацию и отключаемся от ripd с помощью Ctrl+D:

```
R1-RIP(config)# write memory
Configuration saved to /etc/quagga/ripd.conf
R1-RIP(config)# exit
```

4.4. Настройка маршрутизатора R2

Теперь производится настройка интерфейсов маршрутизатора R2. На R2 запускаем демон zebra, подключаемся к нему по telnet и входим в режим глобального конфигурирования:

```
[root@alt218 ~]# systemctl start zebra
[root@alt218 ~]# telnet 127.0.0.1 2601
Trying 127.0.0.1...
Connected to 127.0.0.1.
User Access Verification
Password:
Router>
Router enable
Router#
Router# configure terminal
```

```
Router(config)#
```

Изменим имя устройства:

```
Router(config)# hostname R2
```

Настройка интерфейса enp1s1f0:

```
R2(config)#interface enp1s1f0
R2(config-if)# ip address 10.10.11.2/24
R2(config-if)# exit
```

Настройка интерфейса enp1s1f1:

```
R2(config)#interface enp1s1f1
R2(config-if)#ip address 10.10.12.2/24
R2(config-if)#exit
R2(config)#
```

Настройка демона zebra завершена, необходимо сохранить конфигурацию. Отключиться от демона можно с помощью комбинации Ctrl+D.

```
R2(config)# write memory
Configuration saved to /etc/quagga/zebra.conf
R2(config)# exit
```

Далее на маршрутизаторе R2 необходимо настроить работу протокола RIPv2 с помощью демона ripd. Запускаем демон, подключаемся к ripd по telnet и входим в режим глобального конфигурирования:

```
[root@alt218 ~]# systemctl start ripd
[root@alt218 ~]# telnet 127.0.0.1 2602
Trying 127.0.0.1...
Connected to 127.0.0.1.
User Access Verification
Password:
router-RIP>
router-RIP> enable
router-RIP#
router-RIP# configure terminal
router-RIP(config)#
```

Изменим имя устройства:

```
router-RIP(config)# hostname R2-RIP
```

Теперь необходимо включить процесс RIP, указать вторую версию протокола, и прописать сети, которые будет анонсировать наш маршрутизатор:

```
R2-RIP(config)# router rip
R2-RIP(config-router)# version 2
```

Пропишем все нужные для анонсирования сети, подключенные к маршрутизатору:

```
R2-RIP(config-router)# network 10.10.11.0/24
R2-RIP(config-router)# network 10.10.13.0/24
```

Чтобы увеличить скорость конвергенции сети при изменениях топологии, уменьшим время стандартных таймеров: таймера Update с 30 до 10 секунд, таймера Timeout с 180 до 60 секунд и таймера Flush с 120 до 30 секунд:

```
R2-RIP(config-router)# timers basic 10 60 30
```

Проверим текущую конфигурацию демона ripd командой `show run`, если все настроено верно, то конфигурация должна содержать следующие строки:

```
R2-RIP(config)# show run
!
router rip
 version 2
 network 10.10.11.0/24
 network 10.10.13.0/24
!
```

Настройка ripd завершена. Сохраняем конфигурацию и отключаемся от ripd с помощью Ctrl+D:

```
R2-RIP(config)# write memory
Configuration saved to /etc/quagga/ripd.conf
R2-RIP(config)# exit
```

4.5. Настройка маршрутизатора R3

Настройка интерфейсов маршрутизатора R3 производится с помощью демона zebra. На R3 запускаем демон zebra, подключаемся к нему по telnet и входим в режим глобального конфигурирования:

```
[root@alt217 ~]# systemctl start zebra
[root@alt217 ~]# telnet 127.0.0.1 2601
```

```
Trying 127.0.0.1...
Connected to 127.0.0.1.
User Access Verification
Password:
Router>
Router> enable
Router#
Router# configure terminal
Router(config)#
```

Изменим имя устройства:

```
Router(config)# hostname R3
```

Настройка интерфейса enp1s1f0:

```
R3(config)#interface enp1s1f0
R3(config-if)#ip address 10.10.13.3/24
R3(config-if)#exit
```

Настройка интерфейса enp1s1f1:

```
R3(config)#interface enp1s1f1
R3(config-if)#ip address 10.10.12.3/24
R3(config-if)#exit
R3(config)#
```

Настройка демона zebra завершена, необходимо сохранить конфигурацию. Отключиться от демона можно с помощью комбинации Ctrl+D:

```
R3(config)# write memory
Configuration saved to /etc/quagga/zebra.conf
R3(config)# exit
```

Далее на маршрутизаторе R3 необходимо настроить работу протокола RIPv2 с помощью демона ripd. Запускаем демон, подключаемся к ripd по telnet и входим в режим глобального конфигурирования:

```
[root@alt217 ~]# systemctl start ripd
[root@alt217 ~]# telnet 127.0.0.1 2602
User Access Verification
Password:
router-RIP>
router-RIP> enable
```

```
router-RIP#
router-RIP# configure terminal
router-RIP(config)#
```

Изменим имя устройства:

```
router-RIP(config)# hostname R3-RIP
```

Теперь необходимо включить процесс RIP, указать вторую версию протокола, и прописать сети, которые будет анонсировать наш маршрутизатор:

```
R3-RIP(config)# router rip
R3-RIP(config-router)# version 2
```

Пропишем все нужные для анонсирования сети, подключенные к маршрутизатору:

```
R3-RIP(config-router)# network 10.10.12.0/24
R3-RIP(config-router)# network 10.10.13.0/24
```

Чтобы увеличить скорость конвергенции сети при изменениях топологии, уменьшим время стандартных таймеров: таймера Update с 30 до 10 секунд, таймера Timeout с 180 до 60 секунд и таймера Flush с 120 до 30 секунд:

```
R3-RIP(config-router)# timers basic 10 60 30
```

Настройка демона ripd завершена, необходимо сохранить конфигурацию. Отключиться от демона можно с помощью комбинации Ctrl+D.

```
R3-RIP(config)# write memory
Configuration saved to /etc/quagga/ripd.conf
R3-RIP(config)# exit
```

4.6. Настройка маршрутизатора R4

Настройка интерфейсов маршрутизатора R4 производится с помощью демона zebra. На R4 запускаем демон zebra, подключаемся к нему по telnet и входим в режим глобального конфигурирования:

```
[root@alt216 ~]# systemctl start zebra
[root@alt216 ~]# telnet 127.0.0.1 2601
Trying 127.0.0.1...
Connected to 127.0.0.1.
User Access Verification
Password:
```

```
Router>
Router> enable
Router#
Router# configure terminal
Router(config)#
```

Изменим имя устройства:

```
Router(config)# hostname R4
```

Настройка интерфейса enp1s1f0:

```
R4(config)#interface enp1s1f0
R4(config-if)#ip address 10.10.13.4/24
R4(config-if)#exit
```

Настройка интерфейса enp1s1f1:

```
R4(config)#interface enp1s1f1
R4(config-if)#ip address 10.10.14.4/24
R4(config-if)#exit
R4(config)#
```

Настройка демона zebra завершена, необходимо сохранить конфигурацию. Отключиться от демона можно с помощью комбинации Ctrl+D:

```
R4(config)# write memory
Configuration saved to /etc/quagga/zebra.conf
R4(config)# exit
```

Далее на маршрутизаторе R4 необходимо настроить работу протокола RIPv2 с помощью демона ripd. Запускаем демон, подключаемся к ripd по telnet и входим в режим глобального конфигурирования:

```
[root@alt216 ~]# systemctl start ripd
[root@alt216 ~]# telnet 127.0.0.1 2602
Trying 127.0.0.1...
Connected to 127.0.0.1.
User Access Verification
Password:
router-RIP>
router-RIP> enable
router-RIP#
router-RIP# configure terminal
```

```
router-RIP(config)#
```

Изменим имя устройства:

```
router-RIP(config)# hostname R4-RIP
```

Теперь необходимо включить процесс RIP, указать вторую версию протокола, и прописать сети, которые будет анонсировать наш маршрутизатор:

```
R4-RIP(config)# router rip
R4-RIP(config-router)# version 2
```

Пропишем все нужные для анонсирования сети, подключенные к маршрутизатору:

```
R4-RIP(config-router)# network 10.10.13.0/24
R4-RIP(config-router)# network 10.10.14.0/24
```

Чтобы увеличить скорость конвергенции сети при изменениях топологии, уменьшим время стандартных таймеров: таймера Update с 30 до 10 секунд, таймера Timeout с 180 до 60 секунд и таймера Flush с 120 до 30 секунд:

```
R4-RIP(config-router)# timers basic 10 60 30
```

Настройка демона ripd завершена, необходимо сохранить конфигурацию. Отключиться от демона можно с помощью комбинации Ctrl+D.

```
R4-RIP(config)# write memory
Configuration saved to /etc/quagga/ripd.conf
R4-RIP(config)# exit
```

4.7. Проверка работы протокола RIPv2

Как только протокол RIPv2 настроен, маршрутизаторы начинают рассылку специальных Update-сообщений, содержащих маршрутную информацию. По умолчанию каждые 30 секунд рассылаются пакеты Response на групповой IP-адрес 224.0.0.9, предназначенный для всех маршрутизаторов, поддерживающих протокол RIPv2. Сообщения RIPv2 response инкапсулируются в протокол UDP и используют порт 520.

Проверим работу протокола RIPv2 на R1 с помощью анализатора трафика, для этого на интерфейсе enp1s1f1 запустим Wireshark и зададим фильтр по протоколу RIP:

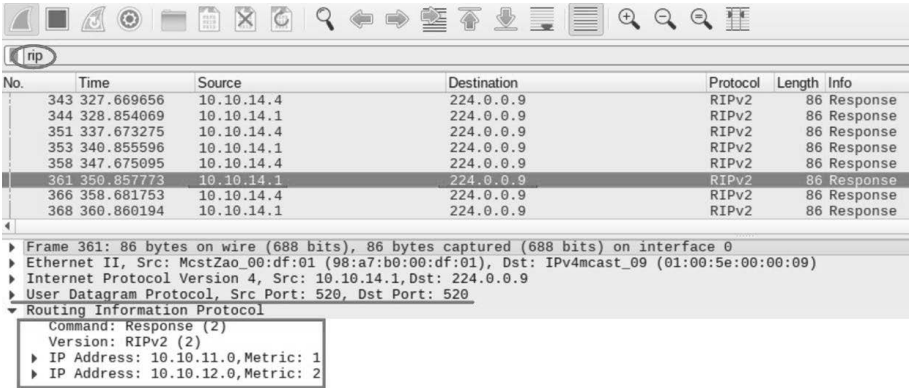


Рисунок 2. Анализ работы протокола RIPv2 на R1

Анализатор трафика показывает, как маршрутизатор R1 с интерфейса enp1s1f1, который имеет IP-адрес 10.10.14.1, отправляет пакет RIPv2 response на групповой IP-адрес 224.0.0.9. на порт 520. Пакеты RIPv2 инкапсулированы протоколом UDP, и содержат основою информацию: тип пакета (Command), версию протокола (Version 2, для работы в сетях IPv4) и маршрутную информацию о состоянии известных сетей.

В анализируемом пакете R1 отправляет в сторону R4 маршрутную информацию о известных ему сетях: о сети 10.10.11.0/24 и о сети 10.10.12.0/24, а также указывает соответствующую метрику. При этом сеть 10.10.11.0/24 подключена напрямую к интерфейсу enp1s1f0 и имеет метрику 1, а о сети 10.10.12.0/24 маршрутизатор узнал из анонсов RIP от маршрутизатора R2 и анонсирует ее с метрикой 2, так как до нее от R1 существует два транзитных участка.

4.8. Проверка сети, функционирующей по протоколу RIPv2

Проверим работу протокола RIPv2 проверкой доступности сети 10.10.12.0/24 с маршрутизатора R1. Так как данная сеть не подключена напрямую к маршрутизатору R1, то с помощью протокола RIPv2 он должен получить к ней маршрут через маршрутизатор R2:

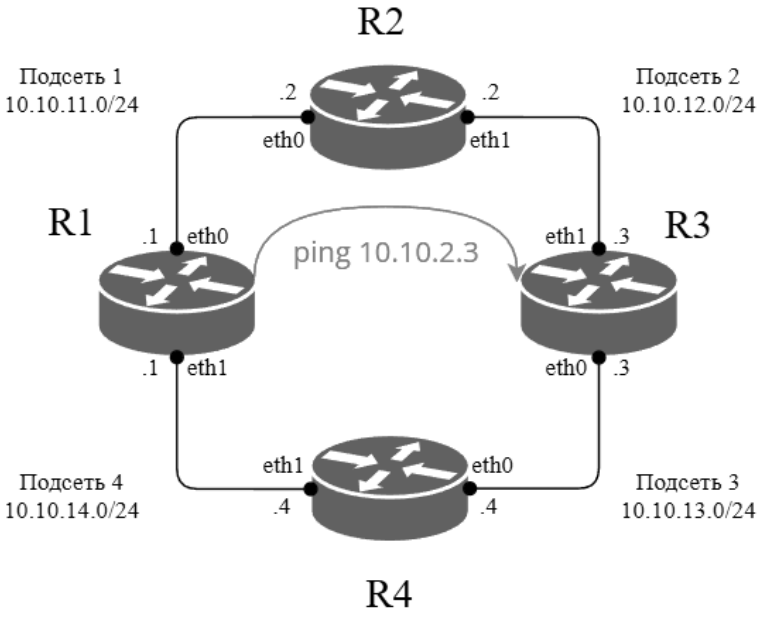


Рисунок 3. Схема моделируемой сети

Чтобы проверить работу сети, используется утилита ping, работающая по протоколу ICMP. На R1 выполняется команда `ping` и указывается адрес интерфейса enp1s1f1 маршрутизатора R3:

```
    [root@alt219 ~]# ping 10.10.12.3 -c 3
    PING 10.10.12.3 (10.10.12.3) 56(84) bytes of data.
    64 bytes from 10.10.12.3: icmp_req=1 ttl=63 time=0.208
ms
    64 bytes from 10.10.12.3: icmp_req=2 ttl=63 time=0.130
ms
    64 bytes from 10.10.12.3: icmp_req=3 ttl=63 time=0.151
ms
    --- 10.10.12.3 ping statistics ---
    3 packets transmitted, 3 received, 0% packet loss, time
1999ms
    rtt min/avg/max/mdev = 0.130/0.163/0.208/0.032 ms
```

Ключ –c (count) задает количество посылаемых ICMP запросов. Интерфейс enp1s1f1 маршрутизатора R3 доступен, ICMP пакеты успешно возвращаются, значит R1 получил необходимый маршрут и сеть работает корректно.

Проверим таблицу маршрутизации RIPv2 на R1. Подключаемся по telnet к zebra, и в привилегированном режиме (enable) используем команду `show ip route rip`:

```
[root@alt219 ~]# telnet 127.0.0.1 2601
Router>
Router> enable
R1# show ip ro rip
Codes: K - kernel route, C - connected, S - static, R
- RIP, O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel, >
- selected route, * - FIB route

R>* 10.10.12.0/24 [120/2] via 10.10.11.2, enp1s1f0,
00:08:48
R>* 10.10.13.0/24 [120/2] via 10.10.14.4, enp1s1f1,
00:05:31
```

В таблице маршрутизации указаны маршруты, полученные R1 из анонсов RIP от своих соседей R2 и R4. Сеть 10.10.12.0/24 доступна через 10.10.11.2 (маршрутизатор R2), а сеть 10.10.13.0/24 доступна через 10.10.14.4 (маршрутизатор R4).

Убедимся в записи таблицы маршрутизации для подсети 10.10.12.0/24, которая доступна через 10.10.11.2, т.е. R2. Выполним с R1 трассировку маршрута утилитой traceroute до адреса 10.10.12.3, находящегося в подсети 10.10.12.0/24:

```
[root@alt219 ~]# traceroute 10.10.12.3
traceroute to 10.10.12.3 (10.10.12.3), 30 hops max, 60
byte packets
 1  10.10.11.2 (10.10.11.2)  0.102 ms  0.058 ms  0.049
ms
 2  10.10.12.3 (10.10.12.3)  0.149 ms  0.112 ms  0.112
ms
```

Как показывает трассировка маршрута, пакеты сначала попадают на адрес 10.10.11.2 маршрутизатора R2, а потом достигают адреса назначения 10.10.12.3 на маршрутизаторе R3.

Проверим как протокол RIPv2 перестраивает маршруты до подсетей при изменении сетевой топологии, путем отключения интерфейса enp1s1f0 на маршрутизаторе R1. После отключения порта, протокол RIPv2 перестроит маршрут к подсети 10.10.12.0/24 через маршрутизатор R4.

Как только происходит отключение интерфейса, на маршрутизаторе R2 происходит вытеснение маршрута. Вытеснение маршрута подразумевает анонсирование отказавшего маршрута со специальным значением метрики – бесконечностью. Маршрутизаторы считают маршруты с бесконечной метрикой отказавшими. Протокол RIP определяет бесконечность как значение 16.

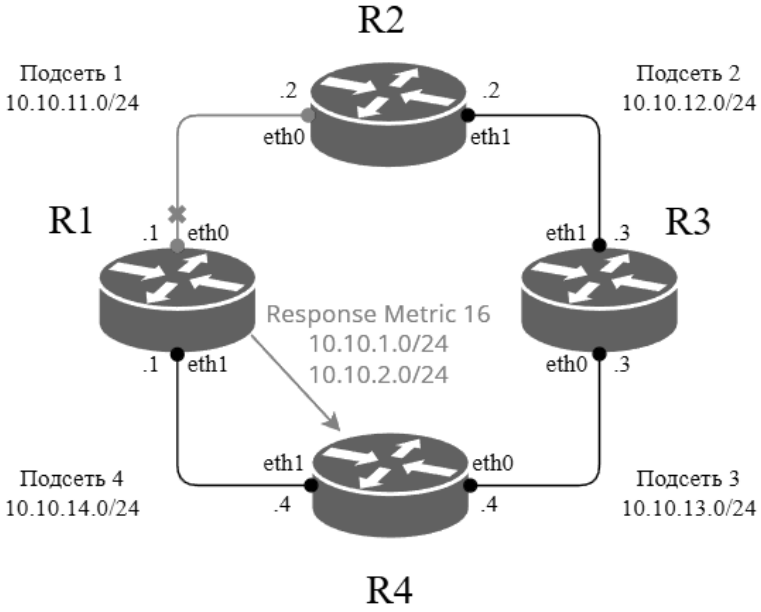


Рисунок 4. Схема сети при отключении интерфейса enp1s1f0

Запускаем Wireshark на интерфейсе enp1s1f1 маршрутизатора R1 и зададим фильтр по протоколу RIP. После отключаем интерфейс enp1s1f0 маршрутизатора R2 и с помощью Wireshark смотрим анонсируемую информацию с интерфейса enp1s1f1 маршрутизатора R1:

```
[root@alt218 ~]# ip link set dev enp1s1f0 down
```

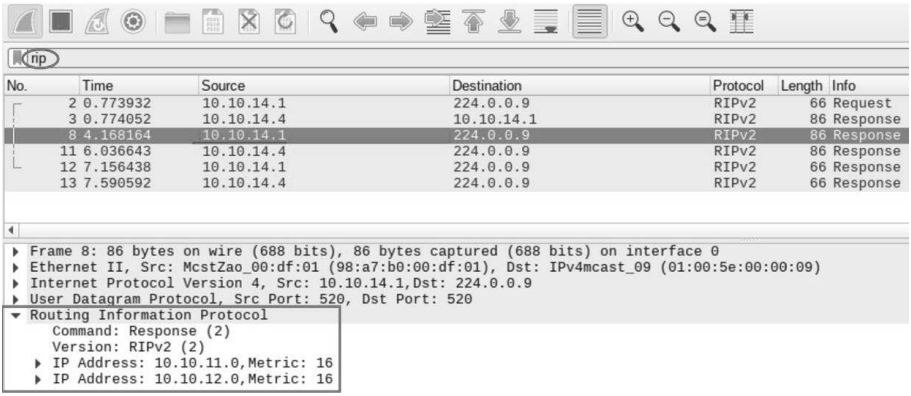


Рисунок 5. Анализ работы протокола RIPv2 при отключении интерфейса

После отключения интерфейса enp1s1f0, маршрутизатор R1 с интерфейса enp1s1f1, который имеет адрес 10.10.14.1, отправляет пакет с обновлением маршрутной информации на групповой адрес 224.0.0.9, в котором указаны сети 10.10.11.0/24 и 10.10.12.0/24 с метрикой 16, что означает их недоступность. Маршрутизатор R1 анонсирует эти сети, так как до отключения они были доступны через интерфейс enp1s1f0. После этого начнется перестроение маршрутов до этих подсетей через маршрутизаторы R4 и R3.

Из-за специфики работы протокола RIPv2, маршрут до подсетей 10.10.11.0/24 и 10.10.12.0/24 не будет перестроен сразу, а только после истечения таймера таймаута, так как отправив обновление маршрутов для недоступных сетей с метрикой 16, маршрутизатор R1 будет хранить эти маршруты пока не истечет таймер таймаута (по умолчанию 180 секунд). Поэтому протокол RIPv2 имеет большое время сходимости, что является одной из причин, по которой он практически не используется в современных сетях. В нашем случае таймер снижен до 60 секунд, по истечении которых маршрут будет перестроен.

После перестроения маршрутов, используем утилиту ping с маршрутизатора R1, проверим доступность адреса 10.10.11.2 из подсети 10.10.11.0/24:

```
    [root@alt219 ~]# ping 10.10.11.2 -c 5
    PING 10.10.11.2 (10.10.11.2) 56(84) bytes of data.
    64 bytes from 10.10.11.2: icmp_req=1 ttl=62 time=0.193
ms
    64 bytes from 10.10.11.2: icmp_req=2 ttl=62 time=0.174
ms
    64 bytes from 10.10.11.2: icmp_req=3 ttl=62 time=0.183
ms
```

```
    64 bytes from 10.10.11.2: icmp_req=4 ttl=62 time=0.180
ms
    64 bytes from 10.10.11.2: icmp_req=5 ttl=62 time=0.170
ms
    -- 10.10.11.2 ping statistics ---
    5 packets transmitted, 5 received, 0% packet loss, time
3996ms
    rtt min/avg/max/mdev = 0.170/0.180/0.193/0.007 ms
```

Интерфейс enp1s1f0 маршрутизатора R2 доступен, ICMP пакеты успешно возвращаются, протокол RIPv2 перестроил маршрут к сети 10.10.11.0/24 через маршрутизаторы R4 и R3.

После перестроения маршрута, используя утилиту traceroute с маршрутизатора R1 проверим путь трафика до адреса 10.10.11.2:

```
    [root@alt219 ~]# traceroute 10.10.11.2
    traceroute to 10.10.11.2 (10.10.11.2), 30 hops max, 60
byte packets
     1  10.10.4.4 (10.10.14.4)  0.133 ms  0.052 ms  0.053
ms
     2  10.10.3.3 (10.10.13.3)  0.170 ms  0.119 ms  0.109
ms
     3  10.10.1.2 (10.10.11.2)  0.204 ms  0.168 ms  0.141
ms
```

Трассировка показывает, что пакеты сначала попадают на узел 10.10.14.4 (маршрутизатор R4), а потом через маршрутизатор R3 (10.10.13.3) попадают в нужную подсеть на заданный IP-адрес 10.10.11.2.

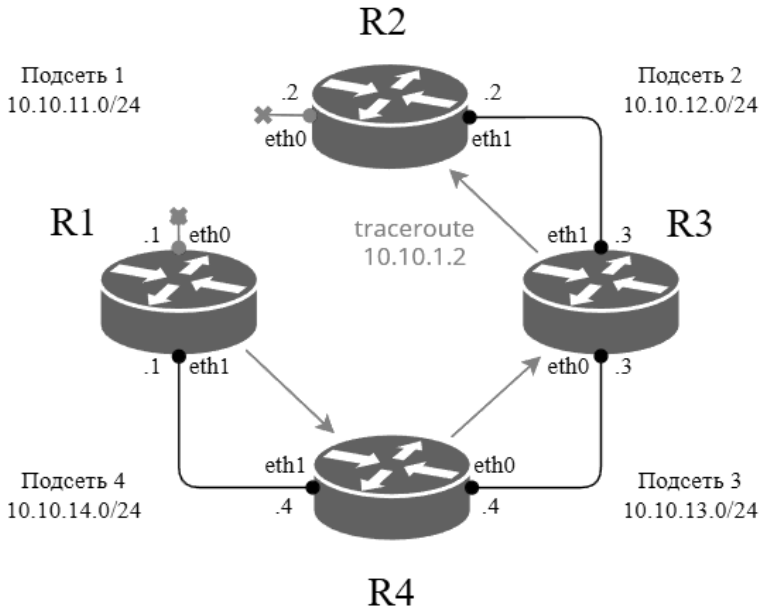


Рисунок 6. Схема прохождения трафика в сети

Когда маршруты перестроены, проверим таблицу маршрутизации RIPv2 на R1. Для этого подключимся к zebra и в привилегированном режиме (enable) используем команду `show ip route rip`:

```
R1# show ip ro rip
Codes: K - kernel route, C - connected, S - static, R
- RIP,
       O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel,
       > - selected route, * - FIB route

R>* 10.10.11.0/24 [120/4] via 10.10.14.4, enp1s1f1,
00:02:34
R>* 10.10.12.0/24 [120/3] via 10.10.14.4, enp1s1f1,
00:02:36
R>* 10.10.13.0/24 [120/2] via 10.10.14.4, enp1s1f1,
00:09:36
```

Теперь в таблице маршрутизации указано, что все существующие сети доступны через 10.10.14.4 (маршрутизатор R4) с интерфейса enp1s1f1.

Управление передачей трафика в сети происходит с помощью метрик маршрутов, в таблице маршрутизации метрика маршрута указана вторым числом в квадратных скобках, первое число — это административная дистанция протокола маршрутизации.

Административная дистанция используется маршрутизатором для того, чтобы определить какому протоколу маршрутизации доверять больше, если у обоих протоколов есть маршрут до подсети назначения. Например, протокол OSPF имеет административную дистанцию 110, а устаревший протокол RIPv2 – административную дистанцию 120. Таким образом, если в сети используются два и более протокола маршрутизации, у которых есть свои маршруты к сети назначения, то маршрутизатор выберет маршрут по протоколу с наименьшей административной дистанцией. У статических маршрутов административная дистанция равна единице, так как они прописываются вручную, а значит степень доверия к ним выше.

Если же в сети используется один протокол маршрутизации, и у него есть несколько разных маршрутов до подсети назначения, то маршрутизатор выбирает маршрут уже на основании метрик. Чем меньше метрика, тем больше маршрутизатор доверяет этому маршруту.

У разных протоколов маршрутизации метрика рассчитывается по-разному. Например, у протокола OSPF метрика рассчитывается на основании суммарной цены маршрута, которая в свою очередь зависит от ширины полосы пропускания интерфейсов и каналов связи на маршруте до подсети назначения. А у протокола RIPv2 метрика рассчитывается исходя из числа транзитных участков между маршрутизатором и подсетью назначения.

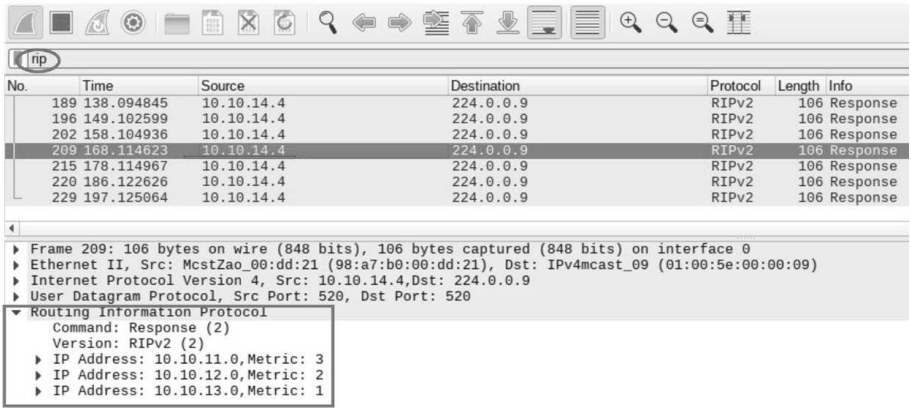


Рисунок 7. Анализ трафика на R1

При этом подсеть 10.10.11.0/24 R4 анонсирует с метрикой 3, так как между интерфейсом enp1s1f0 маршрутизатора R4 и подсетью 10.10.11.0/24 есть три транзитных участка: первый между R4 и R3, второй между R3 и R2, а третий начинается за интерфейсом enp1s1f0 маршрутизатора R2. Сеть 10.10.12.0/24 анонсирована с метрикой 2, т.к. имеет два транзитных участка, а сеть 10.10.13.0/24 он анонсирует с метрикой 1, так как она подключена напрямую к маршрутизатору R4.

Практикум № 15
Протокол динамической маршрутизации OSPF

1. ЦЕЛЬ РАБОТЫ

Изучение особенностей работы и настройки протокола поиска первого кратчайшего пути OSPF (Open Shortest Path First) с помощью отечественных программно-аппаратных средств. Изучение пакета quagga, реализующего протоколы маршрутизации, анализ его функционала в рамках настройки сети, работающей по протоколу динамической маршрутизации OSPF.

2. ЗАДАНИЕ НА РАБОТУ

В работе используется топология сети, представленная на рисунке 1. Маршрутизаторам R1, R2, R3 и R4 соответствуют рабочие станции под номерами PC1, PC2, PC3 и PC4, соответственно. Адресация в моделируемой сети осуществляется следующим образом:

- Подсеть 1: $10.10.X(N_{min}).0/24$ или 10.10.X1.0/24
- Подсеть 2: $10.10.X(N_{min}+1).0/24$ или 10.10.X2.0/24
- Подсеть 3: $10.10.X(N_{min}+2).0/24$ или 10.10.X3.0/24
- Подсеть 4: $10.10.X(N_{min}+3).0/24$ или 10.10.X4.0/24

где $N_{min} = 1$ – наименьший номер из четырех рабочих станций;

X = номер ряда с рабочими станциями, для примера возьмём равное 1.

Наименование интерфейсов указано для примера, на практике оно может отличаться. В нашем же случае `eth0 – enp1s1f0`, а `eth1 – enp1s1f1`.

Последние цифры IP-адресов интерфейсов соответствуют номеру машины.

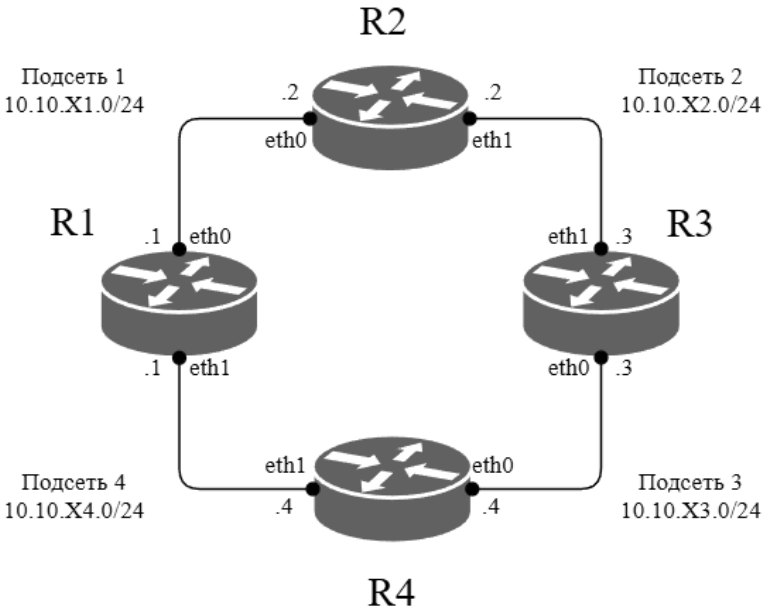


Рисунок 1. Схема лабораторной сети

Основные задачи работы:

- Подготовить машины к работе в сети и установить необходимые пакеты;
- Изменить имя устройств и настроить сетевые интерфейсы, задействованные в работе протокола OSPF с помощью демона zebra;
- С помощью демона ospfd настроить работу протокола OSPF на всех устройствах, настроить анонсирование необходимых подсетей и проверить работу протокола с помощью анализатора трафика Wireshark;
- Проверить отказоустойчивость построенной сети OSPF, смоделировав отказ канала связи между R1 и R2 путем отключения интерфейса enp1s1f0 на R1;
- Проверить работу протокола OSPF при изменении полосы пропускания каналов связи между маршрутизаторами R1, R2 и R4.

3. ТЕРМИНЫ И ИСПОЛЬЗУЕМЫЕ КОМАНДЫ

Используемые команды

`apt-get install` <имя пакета> – команда пакетного менеджера Apt, используемого в ОС Альт. Инициирует установку указанных пакетов;

`chown` – команда смены владельца (change owner), когда также необходимо сменить права вложенные файлы и папки, используется ключ -R (рекурсивно);

`nano` – текстовый редактор, также может создавать текстовые файлы;

`systemctl <start/stop>` – команда управления системными сервисами

`netstat` – команда отображения сетевых соединений системы. Основные ключи: -t (TCP-соединения), -u (UDP-соединения), -l (слушающие сокеты), -p (номер процесса в системе), -n (не переводить IP-адреса в доменные имена);

`grep` – утилита для поиска информации;

`ip` – утилита для настройки и управления сетевыми интерфейсами:

`ip address` – отображение информации о сетевых интерфейсах;

`ip route` – отображение информации о маршрутах;

`telnet` – утилита для подключения к сетевым узлам по протоколу telnet;

`ping` – утилита для проверки сетевых соединений;

`traceroute` – утилита для трассировки маршрута в сети.

Основные команды сетевых демонов quagga

`Show running config` – показать текущую конфигурацию устройства;

`Hostname` <имя> – смена имени устройства;

`Interface` <имя> – переход к конфигурированию интерфейса;

`Ip address` <IP-адрес/маска> – Настройка IP-адреса интерфейса;

`router ospf` – запуск протокола OSPF на устройстве и переход к его конфигурированию;

`Router-id` <x.x.x.x> – настройка идентификатора маршрутизатора OSPF;

`Network` <сеть/маска> `area` <№ зоны OSPF> – настройка анонсирования заданной сети и присвоение ей конкретной зоны;

`Show ip route ospf` – показать таблицу маршрутизации OSPF;

`Show ip ospf neighbor` – информация о соседях маршрутизатора OSPF;

`Write memory` – сохранение конфигурации устройства.

4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

4.1. Установка пакетов и подготовка машин к работе

Для установки пакета quagga, как и любых дополнительных пакетов, необходимы права суперпользователя root, поэтому переходим в режим работы от root командой su – и вводим пароль суперпользователя:

```
[user@alt219 ~]$ su -
Password:
```

Устанавливаем необходимые пакеты. Кроме пакета quagga, для выполнения лабораторной работы понадобятся пакеты telnet и wireshark. Используем ключ –y, чтобы автоматически подтверждать установку пакетов:

```
[root@alt219 ~]# apt-get install –y quagga telnet wireshark
```

Конфигурационные файлы пакета quagga находятся в директории /etc/quagga/, а логи – в /var/log/quagga/.

Чтобы пакет мог корректно работать, необходимо сменить права на файлы конфигурации и логов, которые он использует. Командой `chown` делаем пользователя quagga и его одноименную группу владельцами файлов:

```
[root@alt219 ~]# chown –R quagga:quagga /etc/quagga/
[root@alt219 ~]# chown -R quagga:quagga /var/log/quagga/
```

После того, как пакеты установлены, необходимо на всех машинах включить форвардинг пакетов на уровне ядра системы. Это необходимо для того, чтобы маршрутизаторы могли пересылать пакеты между своими интерфейсами.

Включение форвадинга производится редактированием конфигурационного файла системы sysctl.conf, находящегося в директории /etc/net/. В конфигурационном файле в строке net.ipv4.ip_forward=0 необходимо изменить 0 на 1:

```
[root@alt219 ~]# nano /etc/net/sysctl.conf
net.ipv4.ip_forward=1
```

Сохраняем изменения Ctrl+O и выходим с помощью Ctrl+X. После настройки нужно перезапустить сетевые сервисы системы командой `service network restart`:

```
[root@alt219 ~]# service network restart
```

4.2. Подключение к демонам и работа с конфигурационными файлами пакета quagga

Для настройки динамической маршрутизации OSPF нужны управляющий демон zebra и демон протокола OSPF – ospfd.

Запускаем необходимые демоны:

```
[root@alt219 ~]# systemctl start zebra
[root@alt219 ~]# systemctl start ospfd
```

Настраивать сетевые демоны можно как редактируя их конфигурационные файлы, находящиеся в директории /etc/quagga/, так и подключаясь к их консоли по telnet.

После запуска демонов на нашей машине появятся сокеты (IP-адрес + порт), прослушивающие входящие соединения, с помощью которых и осуществляется подключение к консоли демонов.

Проверим появление необходимых сокетов командой `netstat`, так как демоны quagga используют порты, начинающиеся от 2601, то для точного поиска дополняем запрос командой `grep 260:`

```
[root@alt219 ~]# netstat -tlnp | grep 260
Proto   Local Address  Foreign Address  State  PID/Program name
tcp      127.0.0.1:2601     0.0.0.0:*    LISTEN      - 735/zebra
tcp      127.0.0.1:2604     0.0.0.0:*    LISTEN      - 737/ospfd
```

На локальном хосте (127.0.0.1) демон ospfd слушает порт 2604, а zebra – порт 2601.

Используем telnet для подключения к консоли zebra или ospfd, указывается IP-адрес и порт необходимого демона:

`[root@alt219 ~]# telnet 127.0.0.1 2601` – подключение к zebra

`[root@alt219 ~]# telnet 127.0.0.1 2604` – подключение к ospfd

Подключимся к zebra. Вводим пароль zebra (по умолчанию) и попадаем в его консоль:

```
[root@alt219 ~]# telnet 127.0.0.1 2601
Trying 127.0.0.1...
Connected to 127.0.0.1.
User Access Verification
Password:
Router>
```

Для того чтобы начать настройку, сначала нужно войти в привилегированный режим командой `enable`. Привилегированный режим также защищен паролем (по умолчанию zebra):

```
Router> enable
Password:
Router#
```

Чтобы посмотреть текущую конфигурацию маршрутизатора, используется команда `show running-config`, или сокращенно `show run`:

```
Router# show running config
```

Вывод команды покажет конфигурацию, записанную в файле /etc/quagga/zebra.conf, рассмотрим ее подробнее:

Имя хоста:

```
hostname Router
```

Зашифрованный пароль для подключения к нашему маршрутизатору:

```
password 8 LrjDz/a2KALVQ
```

Зашифрованный пароль для входа в привилегированный режим enable:

```
enable password 8 LrjDz/a2KALVQ
```

Команда, включающая шифрование паролей при просмотре конфигурации:

```
service password-encryption
```

Путь к файлу с логами:

```
log file /var/log/quagga/zebra.log
```

Список контроля доступа (ACL) с именем localhost, разрешающий доступ к консоли демона zebra только с локального хоста, и запрещающий все остальные соединения:

```
access-list localhost permit 127.0.0.1/32
access-list localhost deny any
```

Команды, указывающие применять список доступа localhost для входящих соединений по виртуальным терминальным линиям vty. Подключаясь по telnet, мы занимаем одну виртуальную терминальную линию vty:

```
line vty
```

```
access-class localhost
```

Внесение изменений в конфигурацию устройства производится в режиме глобального конфигурирования (config). Вход в режим осуществляется командой `configure terminal` или сокращённо `conf t`:

```
router# configure terminal
router(config)#
```

4.3. Настройка маршрутизатора R1

Для работы протокола OSPF на маршрутизаторе R1 с помощью демона zebra настраиваются все сетевые интерфейсы, задействованные в работе протокола OSPF. Подключаемся к zebra и входим в режим глобального конфигурирования:

```
[root@alt219 ~]# telnet 127.0.0.1 2601
Trying 127.0.0.1...
Connected to 127.0.0.1.
User Access Verification
Password:
Router>
Router> enable
Router#
Router# configure terminal
Router(config)#
```

Изменим имя устройства:

```
Router(config)# hostname R1
```

В процессе работы протокола OSPF будут задействованы два интерфейса маршрутизатора: enp1s1f0 и enp1s1f1.

Настройка интерфейса enp1s1f0:

`R1(config)# interface enp1s1f0` – переход к конфигурированию интерфейса;

`R1(config-if)# ip address 10.10.11.1/24` – присваиваем адрес и маску подсети;

`R1(config-if)# bandwidth 100000` – задается скорость работы интерфейса в кбит/с (100 Мбит/с). Команда необходима для работы механизмов QoS (Quality of Service) и не меняет реальную физическую скорость интерфейса, а лишь заставляет маршрутизатор думать, что скорость является такой.

`R1(config-if)# exit` – выход из режима настройки интерфейса enp1s1f0.

```
R1(config)#
```

Настройка интерфейса enp1s1f1:

```
R1(config)# interface enp1s1f1
R1(config-if)# ip address 10.10.14.1/24
R1(config-if)#             bandwidth             100000
R1(config-if)# exit
R1(config)#
```

Чтобы сохранить все произведенные настройки, нужно сохранить текущую конфигурацию zebra командой `write memory` или сокращенно `wr`:

```
R1(config)# write memory
Configuration saved to /etc/quagga/zebra.conf
R1(config)# exit
```

Вся конфигурация была перезаписана в /etc/quagga/zebra.conf.

Настройка демона zebra завершена, отключиться от демона можно с помощью комбинации Ctrl+D.

Далее на маршрутизаторе R1 необходимо настроить работу протокола OSPF с помощью демона ospfd. Подключаемся к ospfd по telnet:

```
[root@alt219 ~]# telnet 127.0.0.1 2604
Trying 127.0.0.1...
Connected to 127.0.0.1.
User Access Verification
Password:
router-OSPF>
router-OSPF> enable
router-OSPF#
```

Командой `show running-config` можно посмотреть текущую конфигурацию, записанную в /etc/quagga/ospfd.conf. Для внесения изменений в конфигурацию, переходим в режим глобального конфигурирования:

```
router-OSPF# configure terminal
router-OSPF(config)#
```

Изменим имя устройства:

```
router-OSPF(config)# hostname R1-OSPF
```

Теперь необходимо включить процесс OSPF и прописать сети, которые будет анонсировать наш маршрутизатор:

- `R1-OSPF(config)# router ospf` – включаем OSPF на устройстве и переходим к его конфигурированию;
- Для правильной работы у маршрутизатора OSPF должен быть задан идентификатор маршрутизатора Router ID (RID).
- Изначально, при запуске процесса OSPF, маршрутизаторы в качестве Router ID используют самый большой IP-адрес физического интерфейса. Оставлять настройку по умолчанию не рекомендуется, так как в случае отключения физического интерфейса нарушится и процесс OSPF, идентификатор которого был привязан к IP-адресу физического интерфейса, это может привести к нарушению работы протокола OSPF на маршрутизаторе.
- Router ID назначается следующим образом в порядке приоритета:
- с помощью команды `router-id`. Если команда задана, то маршрутизатор использует ее значение;
- если на каком-либо loopback-интерфейсе маршрутизатора сконфигурирован IP-адрес и этот интерфейс в состоянии up, то маршрутизатор выбирает loopback-интерфейс с самым большим IP-адресом;
- маршрутизатор сам выбирает самый большой IP-адрес из имеющихся физических интерфейсов.

Таким образом, лучше назначать RID вручную командой `router-id` или использовать логические loopback-интерфейсы, т.к. они не полагаются на аппаратные средства, и переходят в рабочее состояние сразу при запуске устройства.

Зададим Router ID с помощью команды `router-id`:

```
R1-OSPF(config-router)# router-id 1.1.1.1
```

Далее необходимо прописать все сети, которые будет анонсировать маршрутизатор по OSPF. При настройке сетей в OSPF необходимо указывать к какой зоне `area` они принадлежат.

В лабораторной работе будет использоваться одна нулевая зона, называемая магистральной (backbone area). Если в сети существует несколько зон, все они должны быть подсоединены к магистральной зоне.

Пропишем все сети, подключенные к R1, которые необходимо анонсировать, и определим их в нулевую зону:

```
R1-OSPF(config-router)# network 10.10.11.0/24 area 0
R1-OSPF(config-router)# network 10.10.14.0/24 area 0
R1-OSPF(config-router)# exit
```

Настройка демона ospfd завершена. Сохраняем конфигурацию и отключаемся от ospfd с помощью Ctrl+D.

```
R1-OSPF(config)# write memory
Configuration saved to /etc/quagga/ospfd.conf
R1-OSPF(config)# exit
```

4.4. Проверка работы протокола OSPF

Как только маршрутизатор настроен, он начинает поиск соседей, рассылая пакеты «Hello» на групповой IP-адрес 224.0.0.5, предназначенный для всех маршрутизаторов, поддерживающих протокол OSPF. Маршрутизаторы OSPF получают пакеты Hello, посылаемые на этот IP-адрес и узнают из них о других соседях в сети.

Проверим работу протокола OSPF на R1 с помощью анализатора трафика, для этого на интерфейсе enp1s1f1 запустим Wireshark и зададим фильтр по протоколу OSPF.

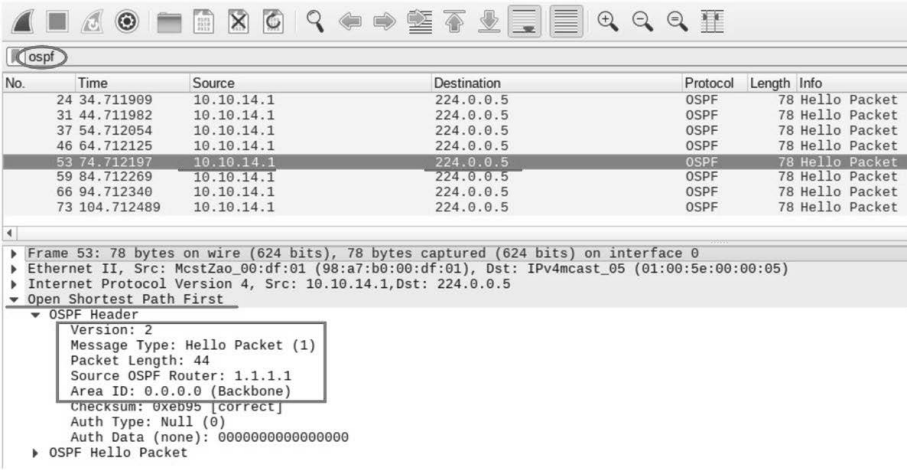


Рисунок 2. Анализ работы протокола OSPF на R1

Анализатор трафика показывает, как маршрутизатор R1 с интерфейса enp1s1f1, который имеет IP-адрес 10.10.14.1, отправляет пакеты OSPF Hello на групповой IP-адрес 224.0.0.5. Заголовок пакета OSPF Hello содержит основные

параметры маршрутизатора, а именно: версию протокола OSPF (Version 2, версия OSPF для протокола IPv4), тип пакета (Hello Packet), Router ID маршрутизатора (Source OSPF Router) и идентификатор зоны (Area ID).

Hello пакеты используются для обнаружения соседей, а также выступают в роли keepalive-пакетов для проверки доступности, и отправляются по умолчанию каждые 10 секунд. Пакет несет в себе параметры, о которых маршрутизаторы OSPF должны договориться перед тем как становиться соседями.

4.5. Настройка маршрутизатора R2

Теперь производится настройка интерфейсов маршрутизатора R2. На R2 запускаем демон zebra, подключаемся к нему по telnet и входим в режим глобального конфигурирования:

```
[root@alt218 ~]# systemctl start zebra
[root@alt218 ~]# telnet 127.0.0.1 2601
Trying 127.0.0.1...
Connected to 127.0.0.1.
User Access Verification
Password:
Router>
Router enable
Router#
Router# configure terminal
Router(config)#
```

Изменим имя устройства:

```
Router(config)# hostname R2
```

Настройка интерфейса enp1s1f0:

```
R2(config)#interface enp1s1f0
R2(config-if)# ip address 10.10.11.2/24
R2(config-if)# bandwidth 100000
R2(config-if)# exit
```

Настроим интерфейс enp1s1f1:

```
R2(config)# interface enp1s1f1
R2(config-if)# ip address 10.10.12.2/24
R2(config-if)# bandwidth 100000
```

```
R2(config-if)# exit
R2(config)#
```

Настройка демона zebra завершена, необходимо сохранить конфигурацию. Отключиться от демона можно с помощью комбинации Ctrl+D.

```
R2(config)# write memory
Configuration saved to /etc/quagga/zebra.conf
R2(config)# exit
```

Далее на маршрутизаторе R2 необходимо настроить работу протокола OSPF с помощью демона ospfd. Запускаем демон, подключаемся к ospfd по telnet и входим в режим глобального конфигурирования:

```
[root@alt218 ~]# systemctl start ospfd
[root@alt218 ~]# telnet 127.0.0.1 2604
Trying 127.0.0.1...
Connected to 127.0.0.1.
User Access Verification
Password:
router-OSPF>
router-OSPF> enable
router-OSPF#
router-OSPF# configure terminal
router-OSPF(config)#
```

Изменим имя устройства:

```
router-OSPF(config)# hostname R2-OSPF
```

Теперь необходимо включить процесс OSPF, задать идентификатор Router ID и прописать сети, которые будет анонсировать наш маршрутизатор:

```
R2-OSPF(config)# router ospf
R2-OSPF(config-router)# router-id 2.2.2.2
```

Пропишем все нужные для анонсирования сети, подключенные к маршрутизатору и определим их в нулевую зону:

```
R2-OSPF(config-router)# network 10.10.11.0/24 area 0
R2-OSPF(config-router)# network 10.10.12.0/24 area 0
```

Проверим текущую конфигурацию демона ospfd командой `show run`, если все настроено верно, то основная конфигурация должна содержать следующие строки:

```
R2-OSPF(config)# show run
!
router ospf
 ospf router-id 2.2.2.2
 network 10.10.11.0/24 area 0.0.0.0
 network 10.10.12.0/24 area 0.0.0.0
!
```

Настройка демона ospfd завершена, необходимо сохранить конфигурацию. Отключиться от демона можно с помощью комбинации Ctrl+D.

```
R2-OSPF(config)# write memory
Configuration saved to /etc/quagga/ospfd.conf
R2-OSPF(config)# exit
```

4.6. Настройка маршрутизатора R3

Настройка интерфейсов маршрутизатора R3 производится с помощью демона zebra. На R3 запускаем демон zebra, подключаемся к нему по telnet и входим в режим глобального конфигурирования:

```
[root@alt217 ~]# systemctl start zebra
[root@alt217 ~]# telnet 127.0.0.1 2601
Trying 127.0.0.1...
Connected to 127.0.0.1.
User Access Verification
Password:
Router>
Router> enable
Router#
Router# configure terminal
Router(config)#
```

Изменим имя устройства:

```
Router(config)# hostname R3
```

Настройка интерфейса enp1s1f0:

```
R3(config)#interface enp1s1f0
R3(config-if)# ip address 10.10.13.3/24
R3(config-if)# bandwidth 100000
R3(config-if)# exit
```
Настроим интерфейс enp1s1f1:
```
R3(config)# interface enp1s1f1
R3(config-if)# ip address 10.10.12.3/24
R3(config-if)# bandwidth 100000
R3(config-if)# exit
R3(config)#
```

Настройка демона zebra завершена, необходимо сохранить конфигурацию. Отключиться от демона можно с помощью комбинации Ctrl+D:
```
R3(config)# write memory
Configuration saved to /etc/quagga/zebra.conf
R3(config)# exit
```

Далее на маршрутизаторе R3 необходимо настроить работу протокола OSPF с помощью демона ospfd. Запускаем демон, подключаемся к ospfd по telnet и входим в режим глобального конфигурирования:
```
[root@alt217 ~]# systemctl start ospfd
[root@alt217 ~]# telnet 127.0.0.1 2604
Trying 127.0.0.1...
Connected to 127.0.0.1.
User Access Verification
Password:
router-OSPF>
router-OSPF> enable
router-OSPF#
router-OSPF# configure terminal
router-OSPF(config)#
```

Изменим имя устройства:
```
router-OSPF(config)# hostname R3-OSPF
```

Теперь необходимо включить процесс OSPF, задать идентификатор Router ID и прописать сети, которые будет анонсировать наш маршрутизатор:
```
R3-OSPF(config)# router ospf
R3-OSPF(config-router)# router-id 3.3.3.3
```

Пропишем все нужные для анонсирования сети, подключенные к маршрутизатору и определим их в нулевую зону:

```
R3-OSPF(config-router)# network 10.10.12.0/24 area 0
R3-OSPF(config-router)# network 10.10.13.0/24 area 0
R3-OSPF(config-router)# exit
```

Настройка демона ospfd завершена, необходимо сохранить конфигурацию. Отключиться от демона можно с помощью комбинации Ctrl+D.

```
R3-OSPF(config)# write memory
Configuration saved to /etc/quagga/ospfd.conf
R3-OSPF(config)# exit
```

4.7. Настройка маршрутизатора R4

Настройка интерфейсов маршрутизатора R4 производится с помощью демона zebra. На R4 запускаем демон zebra, подключаемся к нему по telnet и входим в режим глобального конфигурирования:

```
[root@alt216 ~]# systemctl start zebra
[root@alt216 ~]# telnet 127.0.0.1 2601
Connected to 127.0.0.1.
User Access Verification
Password:
Router>
Router> enable
Router#
Router# configure terminal
Router(config)#
```

Изменим имя устройства:

```
Router(config)# hostname R4
```

Настройка интерфейса enp1s1f0:

```
R4(config)# interface enp1s1f0
R4(config-if)# ip address 10.10.13.4/24
R4(config-if)# bandwidth 100000
R4(config-if)# exit
```

Настроим интерфейс enp1s1f1:

```
R4(config)# interface enp1s1f1
R4(config-if)# ip address 10.10.14.4/24
R4(config-if)# bandwidth 100000
R4(config-if)# exit
R4(config)#
```

Настройка демона zebra завершена, необходимо сохранить конфигурацию. Отключиться от демона можно с помощью комбинации Ctrl+D:

```
R4(config)# write memory
Configuration saved to /etc/quagga/zebra.conf
R4(config)# exit
```

Далее на маршрутизаторе R4 необходимо настроить работу протокола OSPF с помощью демона ospfd. Запускаем демон, подключаемся к ospfd по telnet и входим в режим глобального конфигурирования:

```
[root@alt216 ~]# systemctl start ospfd
[root@alt216 ~]# telnet 127.0.0.1 2604
Connected to 127.0.0.1.
User Access Verification
Password:
router-OSPF>
router-OSPF> enable
router-OSPF#
router-OSPF# configure terminal
router-OSPF(config)#
```

Изменим имя устройства:

```
router-OSPF(config)# hostname R4-OSPF
```

Теперь необходимо включить процесс OSPF, задать идентификатор Router ID и прописать сети, которые будет анонсировать наш маршрутизатор:

```
R4-OSPF(config)# router ospf
R4-OSPF(config-router)# router-id 4.4.4.4
```

Пропишем все нужные для анонсирования сети, подключенные к маршрутизатору и определим их в нулевую зону:

```
R4-OSPF(config-router)# network 10.10.14.0/24 area 0
R4-OSPF(config-router)# network 10.10.13.0/24 area 0
R4-OSPF(config-router)# exit
```

Настройка демона ospfd завершена, необходимо сохранить конфигурацию. Отключиться от демона можно с помощью комбинации Ctrl+D.

```
R4-OSPF(config)# write memory
Configuration saved to /etc/quagga/ospfd.conf
R4-OSPF(config)# exit
```

4.8. Анализ работы сети, работающей по протоколу OSPF

Когда все маршрутизаторы сконфигурированы, между ними должны установиться соседские отношения с помощью пакетов OSPF Hello. Проверим наличие соседских отношений на маршрутизаторе R1. Для этого подключаемся к ospfd и из привилегированного режима (enable) вводим команду `show ip ospf neighbor`:

```
[root@alt216 ~]# telnet 127.0.0.1 2604
R1-OSPF>
R1-OSPF> enable
R1-OSPF# show ip ospf neighbor
Neighbor ID  Pri   State      Dead Time    Address     Interface
2.2.2.2           1    Full/Backup      36.459s      10.10.11.2
enp1s1f0:10.10.11.1
4.4.4.4           1    Full/Backup      35.128s      10.10.14.4
enp1s1f1:10.10.14.1
```

Вывод команды показывает, что R1 установил соседские отношения с R2 и R4. Таблица содержит следующие обозначения:

Neighbor ID: Идентификатор (Router ID) маршрутизатора, с которым установлено соседство;

Pri: Поле приоритета, указывает приоритет соседнего маршрутизатора. По умолчанию приоритеты установлены на 1;

Dead Time: Оставшееся время, в течении которого маршрутизатор будет ждать пакет OSPF Hello от соседа, прежде чем объявить его недоступным (по умолчанию 40 секунд, т.е. 4 интервала Hello);

State: Поле State показывает функциональное состояние соседнего маршрутизатора. Состояние Full означает полную синхронизацию с соседом;

Address: Поле Address указывает IP-адрес интерфейса соседа, по которому с ним осуществляется соединение;

Interface: Поле Интерфейс указывает локальный интерфейс, на котором было установлено OSPF соседство.

Проверим работу протокола OSPF проверкой доступности сети 10.10.12.0/24 с маршрутизатора R1. Так как данная сеть не подключена напрямую к маршрутизатору R1, то с помощью протокола OSPF он должен получить к ней маршрут через маршрутизатор R2:

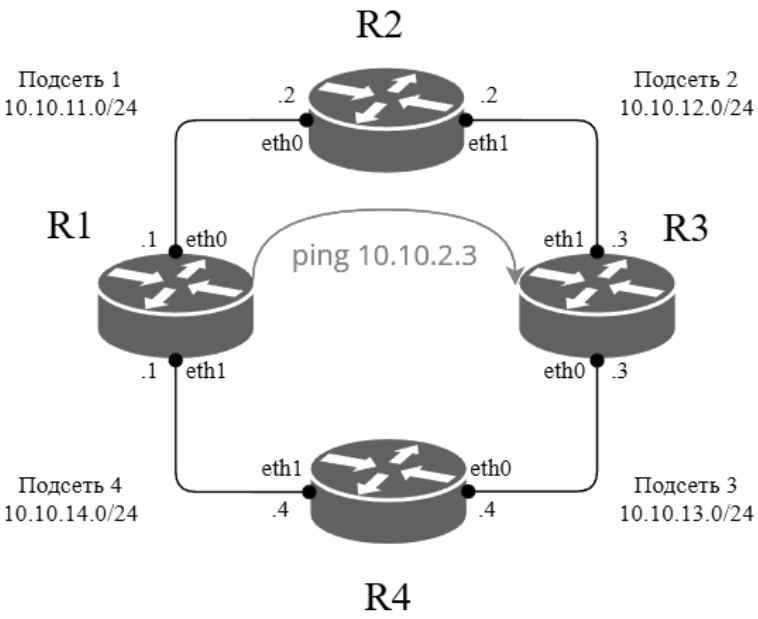


Рисунок 3. Проверка связности настроенной сети

На R1 выполним команду `ping` и укажем адрес интерфейса enp1s1f1 на R3:

```
[root@alt219 ~]# ping 10.10.12.3 -c 3
PING 10.10.12.3 (10.10.12.3) 56(84) bytes of data.
64 bytes from 10.10.12.3: icmp_req=1 ttl=63 time=0.176
ms
64 bytes from 10.10.12.3: icmp_req=2 ttl=63 time=0.138
ms
64 bytes from 10.10.12.3: icmp_req=3 ttl=63 time=0.121
ms
--- 10.10.12.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time
1998ms
```

```
rtt min/avg/max/mdev = 0.121/0.145/0.176/0.023 ms
```

Ключ –c (count) задает количество посылаемых ICMP запросов. Интерфейс enp1s1f1 маршрутизатора R3 доступен, ICMP пакеты успешно возвращаются, значит R1 получил необходимый маршрут и сеть работает верно.

Проверим таблицу маршрутизации OSPF на R1. Подключаемся по telnet к zebra, и в привилегированном режиме (enable) используем команду `show ip route ospf`:

```
[root@alt219 ~]# telnet 127.0.0.1 2601
R1>
R1> enable
R1# show ip ro ospf
Codes: K - kernel route, C - connected, S - static, R
- RIP,O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel,> -
selected route, * - FIB route

O      10.10.11.0/24  [110/1]  is  directly  connected,
enp1s1f0, 00:08:16
O>* 10.10.12.0/24 [110/2] via 10.10.11.2, enp1s1f0,
00:07:26
O>* 10.10.13.0/24 [110/2] via 10.10.14.4, enp1s1f1,
00:06:36
O      10.10.14.0/24  [110/1]  is  directly  connected,
enp1s1f1, 00:08:16
```

В таблице маршрутизации указаны маршруты, полученные R1 по OSPF от своих соседей R2 и R4. Сеть 10.10.12.0/24 доступна через 10.10.11.2 (маршрутизатор R2), а сеть 10.10.13.0/24 доступна через 10.10.14.4 (маршрутизатор R4).

Убедимся в записи таблицы маршрутизации для подсети 10.10.12.0/24, которая доступна через 10.10.11.2, т.е. R2. Выполним с R1 трассировку маршрута утилитой traceroute до адреса 10.10.12.3, находящегося в подсети 10.10.12.0/24:

```
[root@alt219 ~]# traceroute 10.10.12.3
traceroute to 10.10.12.3 (10.10.12.3), 30 hops max, 60
byte packets
 1  10.10.1.2 (10.10.11.2)  0.128 ms  0.082 ms  0.109
ms
```

```
    2  10.10.2.3 (10.10.12.3)  0.124 ms  0.097 ms  0.100
ms
```

Как показывает трассировка маршрута, пакеты сначала попадают на адрес 10.10.11.2 маршрутизатора R2, а потом достигают адреса назначения 10.10.12.3 на маршрутизаторе R3.

Проверим как протокол OSPF перестраивает маршруты до подсетей при изменении сетевой топологии, путем отключения интерфейса enp1s1f0 на маршрутизаторе R1. После отключения порта, протокол OSPF перестроит маршрут к подсети 10.10.12.0/24 через маршрутизатор R4.

Как только происходит отключение интерфейса, алгоритм OSPF создает анонсы LSA (Link State Advertisement), в которых содержатся изменения в базе данных LSDB (Link State Data Base) маршрутизатора, а именно – отключенный интерфейс enp1s1f0, и рассылает их своим соседям, а они своим соседям, пока у всех маршрутизаторов не появится одинаковая копия базы LSDB, содержащая информацию об измененной топологии сети.

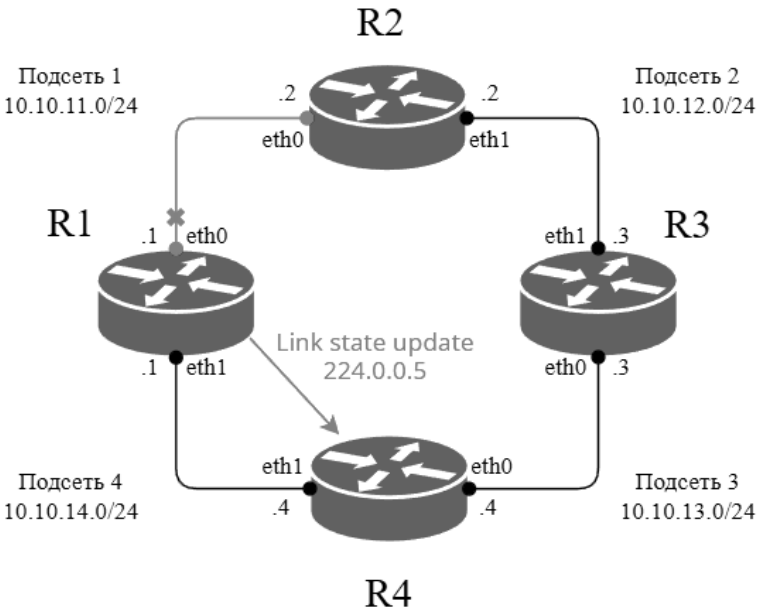


Рисунок 4. Схема сети при отключении интерфейса enp1s1f0

Запускаем Wireshark на интерфейсе enp1s1f1 маршрутизатора R1 и зададим фильтр по протоколу ospf. После отключаем интерфейс enp1s1f0 маршрутизатора R1:

```
[root@alt219 ~]# ip link set dev enp1s1f0 down
```

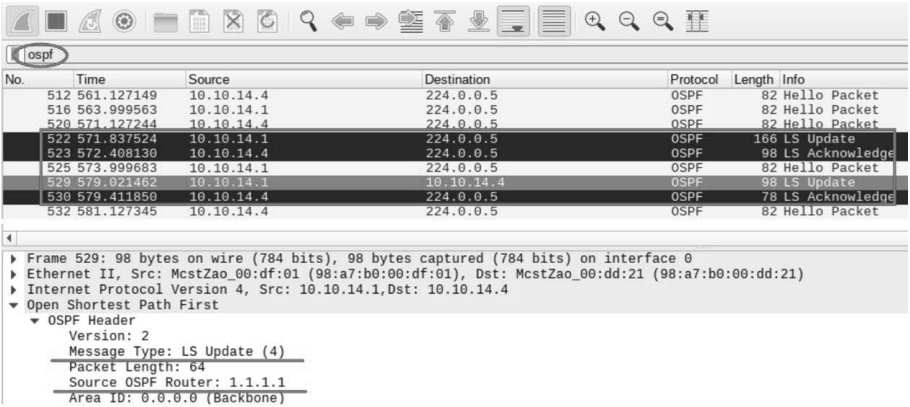


Рисунок 5. Анализ работы протокола OSPF

В частности, сообщения LSA, передающие информацию об изменении, называются обновлениями состояния канала LSU (Link State Update). Как показывает анализатор пакетов, при отключении интерфейса маршрутизатор R1 сразу начал отправку своим соседям на групповой адрес 224.0.0.5 Update сообщений LSU, содержащих изменения базы данных LSDB. Маршрутизаторы R1 и R4 обмениваются пакетами LS Update и подтверждениями LS Acknowledgment, после чего произойдет перестроение маршрутов в сети. Теперь базы LSDB всех маршрутизаторов содержат информацию об отключённом интерфейсе enp1s1f0 маршрутизатора R1 и запускают алгоритм Дейкстры для повторного расчета любых маршрутов, на которые повлиял отказ интерфейса.

После того, как интерфейс отключен, проверим доступна ли сеть 10.10.11.0/24 с маршрутизатора R1 при помощи команды `ping`:

```
[root@alt219 ~]# ping 10.10.11.2 -c 5
PING 10.10.11.2 (10.10.11.2) 56(84) bytes of data.
64 bytes from 10.10.11.2: icmp_req=1 ttl=62 time=0.241 ms
64 bytes from 10.10.11.2: icmp_req=2 ttl=62 time=0.187 ms
64 bytes from 10.10.11.2: icmp_req=3 ttl=62 time=0.185 ms
64 bytes from 10.10.11.2: icmp_req=4 ttl=62 time=0.184 ms
64 bytes from 10.10.11.2: icmp_req=5 ttl=62 time=0.202 ms
```

```
--- 10.10.11.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4000ms
rtt min/avg/max/mdev = 0.184/0.199/0.241/0.028 ms
```

Интерфейс enp1s1f0 маршрутизатора R2 доступен, ICMP пакеты успешно возвращаются и значит, что протокол OSPF перестроил маршрут к подсети 10.10.11.0/24 через маршрутизатор R4 и R3.

Чтобы проверить, каким путем прошли пакеты до подсети 10.10.11.0/24, с R1 выполним трассировку маршрута до адреса 10.10.11.2 на R2, который находится в этой подсети. На R1 используем команду `traceroute`:

```
[root@alt219 ~]# traceroute 10.10.11.2
traceroute to 10.10.11.2 (10.10.11.2), 30 hops max, 60 byte packets
 1  10.10.4.4 (10.10.14.4)  0.132 ms  0.056 ms  0.054 ms
 2  10.10.3.3 (10.10.13.3)  0.161 ms  0.113 ms  0.102 ms
 3  10.10.1.2 (10.10.11.2)  0.176 ms  0.152 ms  0.143 ms
```

Трассировка показывает, что пакеты сначала попадают на узел 10.10.14.4 (маршрутизатор R4), а потом через маршрутизатор R3 (10.10.13.3) попадают в нужную подсеть на заданный IP-адрес 10.10.11.2.

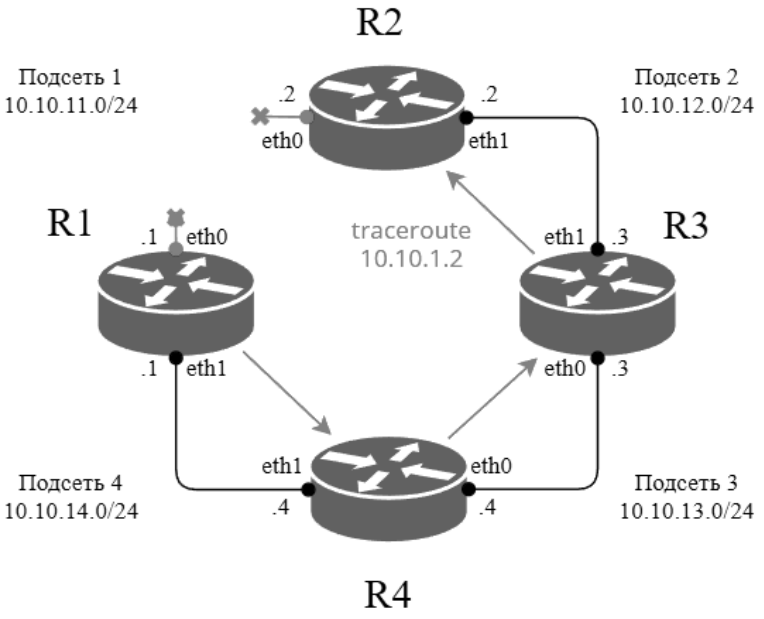


Рисунок 6. Схема прохождения трафика в сети

Таблица маршрутизации на R1 также покажет произошедшие изменения в сетевой топологии. Для этого подключимся к zebra и в привилегированном режиме (enable) используем команду `show ip route ospf`:

```
R1# show ip ro ospf
Codes: K - kernel route, C - connected, S - static, R - RIP,O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel,> - selected route, * - FIB route
O>* 10.10.11.0/24 [110/4] via 10.10.14.4, enp1s1f1, 00:02:37
O>* 10.10.12.0/24 [110/3] via 10.10.14.4, enp1s1f1, 00:03:09
O>* 10.10.13.0/24 [110/2] via 10.10.14.4, enp1s1f1, 00:13:27
O 10.10.14.0/24 [110/1] is directly connected, enp1s1f1, 00:15:07
```

Теперь в таблице маршрутизации указано, что все существующие сети доступны через 10.10.14.4 (маршрутизатор R4) с интерфейса enp1s1f1.

Протокол OSPF также может выбирать наилучший маршрут до подсетей исходя их полосы пропускания канала на пути прохождения трафика. Если снизить скорость линии между маршрутизаторами, то протокол OSPF также сначала начнет рассылку своим соседям Update сообщений, содержащих изменения в топологии сети, а после перестроит маршрут до сети назначения по более скоростным каналам.

На маршрутизаторе R1 вернем в активное состояние интерфейс enp1s1f0:

```
[root@alt219 ~]# ip link set dev enp1s1f0 up
```

Проверим таблицу маршрутизации OSPF на R1. Подключаемся по telnet к zebra, и в привилегированном режиме (enable) используем команду `show ip route ospf`. Подсеть 10.10.11.0/24 появилась в статусе подключенной напрямую (directly connected), а подсеть 10.10.12.0/24 снова доступна через R2:

```
R1# show ip ro ospf
Codes: K - kernel route, C - connected, S - static, R -
RIP, O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel, > -
selected route, * - FIB route

O   10.10.11.0/24 [110/1] is directly connected, enp1s1f0,
00:00:15
O>* 10.10.12.0/24 [110/2] via 10.10.11.2, enp1s1f0,
00:00:15
O>* 10.10.13.0/24 [110/2] via 10.10.14.4, enp1s1f1,
00:00:15
O   10.10.14.0/24 [110/1] is directly connected, enp1s1f1,
00:15:13
```

Изменим пропускную способность канала связи между R1 и R2. На интерфейсе enp1s1f0 маршрутизатора R1 программно снизим скорость интерфейса со 100 Мбит/с до 5 Мбит/с, в zebra в режиме глобального конфигурирования используем команду `bandwidth`, полоса указывается в кбит/с:

```
[root@alt219 ~]# telnet 127.0.0.1 2601
R1>
R1> enable
R1# configure terminal
R1(config)# interface enp1s1f0
R1(config-if)# bandwidth 5000
R1(config-if)# exit
```

Физически скорость интерфейса не изменилась, но маршрутизатор теперь считает скорость интерфейса равной 5 Мбит/с. После этого протокол OSPF снова перестроит маршрут ко всем подсетям через R4. Несмотря на три транзитных участка по новому маршруту через R4 и R3, линия между R1 и R2 все равно будет иметь меньший приоритет из-за низкой скорости передачи, хотя и имеет всего один транзитный участок.

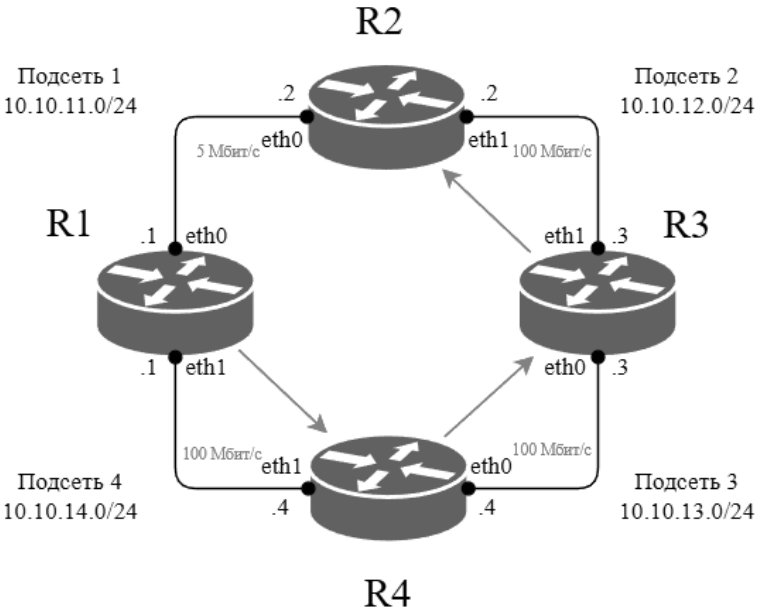


Рисунок 7. Изменение пропускной способности канала между R1 и R2

Таблица маршрутизации на R1 показывает изменения, произошедшие в сети. Теперь маршрут до всех подсетей идет снова идет через 10.10.14.4, т.е. через маршрутизатор R4:

```
    R1(config-if)# do sh ip ro ospf
    Codes: K - kernel route, C - connected, S - static, R -
RIP,
           O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel, N
- NHRP, > - selected route, * - FIB route

    O>*  10.10.11.0/24  [110/4]  via  10.10.14.4,  enp1s1f1,
00:00:07
```

```
    O>*  10.10.12.0/24  [110/3]  via  10.10.14.4,  enp1s1f1,
00:00:07
    O>*  10.10.13.0/24  [110/2]  via  10.10.14.4,  enp1s1f1,
00:00:07
    O   10.10.14.0/24 [110/1] is directly connected, enp1s1f1,
00:17:07
```

Управление передачей трафика в сети происходит с помощью метрик маршрутов, в таблице маршрутизации метрика маршрута указана вторым числом в квадратных скобках, первое число — это административная дистанция протокола маршрутизации.

Административная дистанция используется маршрутизатором для того, чтобы определить какому протоколу маршрутизации доверять больше, если у обоих протоколов есть маршрут до подсети назначения. Например, протокол OSPF имеет административную дистанцию 110, а устаревший протокол RIPv2 – административную дистанцию 120. Таким образом, если в сети используются два и более протокола маршрутизации, у которых есть свои маршруты к сети назначения, то маршрутизатор выберет маршрут по протоколу с наименьшей административной дистанцией. У статических маршрутов административная дистанция равна единице, так как они прописываются вручную, а значит степень доверия к ним выше.

Если же в сети используется один протокол маршрутизации, и у него есть несколько разных маршрутов до подсети назначения, то маршрутизатор выбирает маршрут уже на основании метрик. Чем меньше метрика, тем больше маршрутизатор доверяет этому маршруту.

У разных протоколов маршрутизации метрика рассчитывается по-разному. Например, у протокола RIPv2 метрика рассчитывается исходя из числа транзитных участков между маршрутизатором и подсетью назначения. А у протокола OSPF метрика рассчитывается на основании суммарной цены маршрута, которая в свою очередь зависит от ширины полосы пропускания интерфейсов и каналов связи на маршруте до подсети назначения.

Чтобы проверить, как меняется метрика маршрута в сети, на маршрутизаторе R1 программно изменим цену интерфейса enp1s1f1 путем уменьшения его скорости со 100 Мбит/с до 10 Мбит/с и проанализируем таблицу маршрутизации:

```
R1(config)#
R1(config)# interface enp1s1f1
R1(config-if)# bandwidth 10000
R1(config-if)# exit
```

После изменений трафик все равно будет проходить через маршрутизатор R3, так как суммарная цена маршрута через R3 ниже из-за более высокой пропускной способности каналов, проходящих через него.

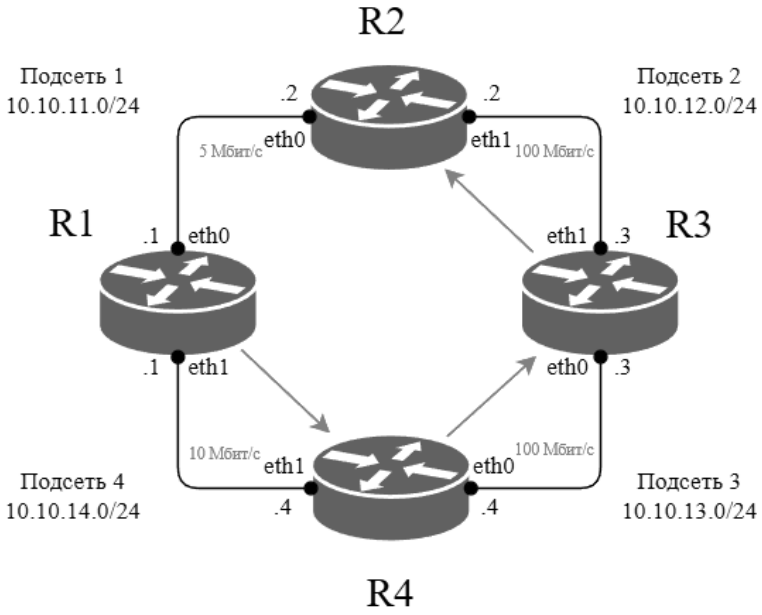


Рисунок 8. Изменение пропускной способности канала между R1 и R4

Уменьшая скорость интерфейса enp1s1f1 маршрутизатора R1, мы увеличиваем цену маршрута на участке сети между R1 и R4, что приводит к увеличению метрик на 9 единиц:

```
R1(config-if)# do show ip ro ospf
Codes: K - kernel route, C - connected, S - static, R
- RIP, O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel,>
- selected route, * - FIB route

O   10.10.11.0/24 [110/13] via 10.10.14.4, enp1s1f1,
00:02:37
O>* 10.10.12.0/24 [110/12] via 10.10.14.4, enp1s1f1,
00:02:37
```

```
   O>* 10.10.13.0/24 [110/11] via 10.10.14.4, enp1s1f1,
00:02:37
   O  10.10.14.0/24  [110/10]  is  directly  connected,
enp1s1f1, 00:02:37
```

После снижения скорости интерфейса enp1s1f1 метрики всех маршрутов увеличились, но трафик по-прежнему идет через R4:

```
   [root@alt219 ~]# traceroute 10.10.11.2
   Traceroute to 10.10.11.2 (10.10.11.2), 30 hops max, 60
byte packets
    1  10.10.14.4 (10.10.14.4)  0.132 ms  0.056 ms  0.054
ms
    2  10.10.13.3 (10.10.13.3)  0.161 ms  0.113 ms  0.102
ms
    3  10.10.11.2 (10.10.11.2)  0.176 ms  0.152 ms  0.143
ms
```

Практикум № 16
Протокол динамической маршрутизации BGP

1. ЦЕЛЬ РАБОТЫ

Изучение особенностей работы и настройки протокола граничного шлюза BGP (Border Gateway Protocol) с помощью отечественных программно-аппаратных средств. Изучение пакета quagga, реализующего протоколы маршрутизации, анализ его функционала в рамках настройки сети, работающей по протоколу динамической маршрутизации BGP.

2. ЗАДАНИЕ НА РАБОТУ

В работе используется топология сети, представленная на рисунке 1. Маршрутизаторам R1, R2, R3 и R4 соответствуют рабочие станции под номерами PC1, PC2, PC3 и PC4 соответственно. Адресация в моделируемой сети осуществляется следующим образом:

Подсеть 1: $10.10.X(N_{min}).0/24$ или 10.10.X1.0/24

Подсеть 2: $10.10.X(N_{min}+1).0/24$ или 10.10.X2.0/24

Подсеть 3: $10.10.X(N_{min}+2).0/24$ или 10.10.X3.0/24

Подсеть 4: $10.10.X(N_{min}+3).0/24$ или 10.10.X4.0/24

Номера автономных систем AS распределяются следующим образом:

AS R1: $6500(N_{min})$ или 65001 AS R2: $6500(N_{min}+1)$ или 65002

AS R3: $6500(N_{min}+2)$ или 65003 AS R4: $6500(N_{min}+3)$ или 65004

Сети в автономных системах реализуются с помощью loopback интерфейсов маршрутизаторов и имеют следующую адресацию:

Сеть в AS R1 : 192.168.(N_{min}).1 или 192.168.1.1

Сеть в AS R2 : 192.168.($N_{min}+1$).1 или 192.168.2.1

Сеть в AS R4 : 192.168.($N_{min}+3$).1 или 192.168.4.1

Сети в AS R3 : 192.168.10.1/30 и 192.168.20.1/30

где $N_{min} = 1$ – наименьший номер из четырех рабочих станций;

X = номер ряда с рабочими станциями, для примера возьмём равное 1.

Наименование интерфейсов указано для примера, на практике оно может отличаться. В нашем же случае `eth0 - enp1s1f0`, а `eth1 - enp1s1f1`.

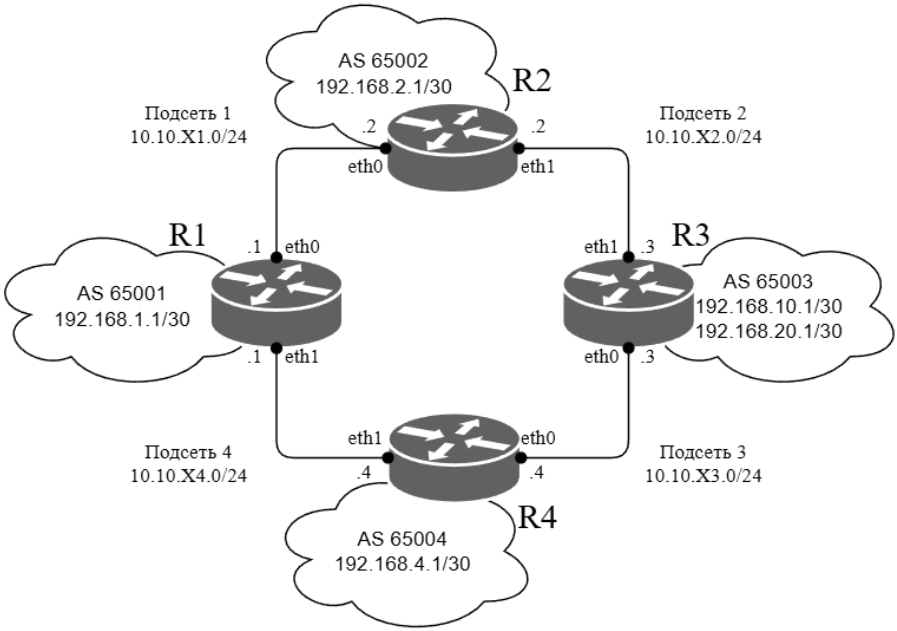


Рисунок 1. Схема моделируемой сети

Основные задачи работы:

- Подготовить машины к работе в сети и установить необходимые пакеты;
- Изменить имя устройств и настроить сетевые интерфейсы, задействованные в работе протокола BGP с помощью демона zebra;
- С помощью демона bgpd настроить работу протокола BGP на всех устройствах, настроить BGP-соседство и анонсирование необходимых префиксов, а также проверить работу протокола с помощью анализатора трафика Wireshark;
- На маршрутизаторе R1 с помощью списков префиксов запретить принимать BGP анонсы сети 192.168.10.0/30 от маршрутизатора R2;
- На маршрутизаторе R1 с помощью карт маршрутов настроить распределение трафика таким образом, чтобы трафик до сети 192.168.10.0/30 передавался через AS 65002, а до 192.168.20.0/30 через AS 65004.

3. ТЕРМИНЫ И ИСПОЛЬЗУЕМЫЕ КОМАНДЫ

Используемые команды

`apt-get install` <имя пакета> – команда пакетного менеджера Apt, используемого в ОС Альт. Инициирует установку указанных пакетов;

`chown` – команда смены владельца (change owner), когда также необходимо сменить права вложенные файлы и папки, используется ключ -R (рекурсивно);

`nano` – текстовый редактор, также может создавать текстовые файлы;

`systemctl <start/stop>` – команда управления системными сервисами

`netstat` – команда отображения сетевых соединений системы. Основные ключи: -t (TCP-соединения), -u (UDP-соединения), -l (слушающие сокеты), -p (номер процесса в системе), -n (не переводить IP-адреса в доменные имена);

`grep` – утилита для поиска информации;

`telnet` – утилита для подключения к сетевым узлам по протоколу telnet;

`ping` – утилита для проверки сетевых соединений;

`traceroute` – утилита для трассировки маршрута в сети.

Основные команды сетевых демонов quagga

`Show running config` – показать текущую конфигурацию устройства;

`Hostname` <имя> – смена имени устройства;

`Interface` <имя> – переход к конфигурированию интерфейса;

`Ip address` <IP-адрес/маска> – Настройка IP-адреса интерфейса;

`Router bgp` <№ AS> – запуск протокола BGP на устройстве и переход к его конфигурированию;

`Neighbor` <IP-адрес> `remote-as` <№ AS> – настройка соседа BGP;

`Network` <IP-адрес> `mask` <m.m.m.m> - настройка анонса сети;

`Show ip bgp neighbors` – информация о соседе BGP;

`Show ip bgp` – показать таблицу маршрутизации BGP;

`Show ip bgp summary` – информация о соседях маршрутизатора BGP;

`Write memory` – сохранение конфигурации устройства;

`Ip prefix-list` <имя списка> `[seq` <value>`]` <deny | permit> <network/length> `[ge` <value>`]` `[le` <value>`]` – настройка списка префиксов;

`Route-map` <имя карты> `<deny | permit>` <seq> – настройка карты маршрутов.

4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

4.1. Установка пакетов и подготовка машин к работе

Для установки пакета quagga, как и любых дополнительных пакетов, необходимы права суперпользователя root, поэтому переходим в режим работы от root командой su – и вводим пароль суперпользователя:

```
[user@alt219 ~]$ su -
Password:
```

Используя пакетный менеджер apt, командой `apt-get install` устанавливаем необходимые пакеты. Кроме пакета quagga, для выполнения лабораторной работы понадобятся пакеты telnet и wireshark. Используем ключ –y, чтобы автоматически подтверждать установку пакетов:

```
[root@alt219 ~]# apt-get install -y quagga telnet wireshark
```

Конфигурационные файлы пакета quagga находятся в директории /etc/quagga/, а логи – в /var/log/quagga/.

Чтобы пакет мог корректно работать, необходимо сменить права на файлы конфигурации и логов, которые он использует. Командой `chown` делаем пользователя quagga и его одноименную группу владельцами файлов:

```
[root@alt219 ~]# chown -R quagga:quagga /etc/quagga/
[root@alt219 ~]# chown -R quagga:quagga /var/log/quagga/
```

После того, как пакеты установлены, необходимо на всех машинах включить форвардинг пакетов на уровне ядра системы. Это необходимо, чтобы маршрутизаторы могли пересылать пакеты между своими интерфейсами.

Включение форвардинга производится редактированием конфигурационного файла системы sysctl.conf, находящегося в директории /etc/net/. В конфигурационном файле в строке net.ipv4.ip_forward=0 необходимо изменить 0 на 1:

```
[root@alt219 ~]# nano /etc/net/sysctl.conf
net.ipv4.ip_forward=1
```

Сохраняем изменения Ctrl+O и выходим с помощью Ctrl+X. После настройки нужно перезапустить сетевые сервисы системы командой `service network restart`:

```
[root@alt219 ~]# service network restart
```

4.2. Подключение к демонам и работа с конфигурационными файлами пакета quagga

Для настройки динамической маршрутизации OSPF нужны управляющий демон zebra и демон протокола BGP – bgpd.

Запускаем необходимые демоны:

```
[root@alt219 ~]# systemctl start zebra
[root@alt219 ~]# systemctl start bgpd
```

Настраивать сетевые демоны можно как редактируя их конфигурационные файлы, находящиеся в директории /etc/quagga/, так и подключаясь к их консоли по telnet.

После запуска демонов на нашей машине появятся сокеты (IP-адрес + порт), прослушивающие входящие соединения, с помощью которых и осуществляется подключение к консоли демонов.

Проверим появление необходимых сокетов командой `netstat:`

```
[root@alt219 ~]# netstat -tlnp
Proto    Local Address  Foreign Address  State  PID/Program name
tcp      127.0.0.1:2601     0.0.0.0:*    LISTEN     -  735/zebra
tcp      127.0.0.1:2605     0.0.0.0:*    LISTEN     - 1217/bgpd
```

На локальном хосте (127.0.0.1) демон bgpd слушает порт 2605, а zebra – порт 2601.

Используем telnet для подключения к консоли zebra или bgpd, указывается IP-адрес и порт необходимого демона:

`[root@alt219 ~]# telnet 127.0.0.1 2601` – подключение к zebra

`[root@alt219 ~]# telnet 127.0.0.1 2605` – подключение к bgpd

Подключимся к zebra. Вводим пароль zebra (по умолчанию) и попадаем в его консоль:

```
[root@alt219 ~]# telnet 127.0.0.1 2601
```

```
Trying 127.0.0.1...
Connected to 127.0.0.1.
User Access Verification
Password:
Router>
```

Для того чтобы начать настройку, сначала нужно войти в привилегированный режим командой `enable`. Привилегированный режим также защищен паролем (по умолчанию zebra):

```
Router> enable
Password:
Router#
```

Чтобы посмотреть текущую конфигурацию маршрутизатора, используется команда `show running-config`, или сокращенно `show run`:

```
Router# show running config
```

Вывод команды покажет конфигурацию, записанную в файле /etc/quagga/zebra.conf, рассмотрим ее подробнее:

Имя хоста:

```
hostname Router
```

Зашифрованный пароль для подключения к нашему маршрутизатору:

```
password 8 LrjDz/a2KALVQ
```

Зашифрованный пароль для входа в привилегированный режим enable:

```
enable password 8 LrjDz/a2KALVQ
```

Команда, включающая шифрование паролей при просмотре конфигурации:

```
service password-encryption
```

Путь к файлу с логами:

```
log file /var/log/quagga/zebra.log
```

Список контроля доступа (ACL) с именем localhost, разрешающий доступ к консоли демона zebra только с локального хоста, и запрещающий все остальные соединения:

```
access-list localhost permit 127.0.0.1/32
```

```
access-list localhost deny any
```

Команды, указывающие применять список доступа localhost для входящих соединений по виртуальным терминальным линиям vty. Подключаясь по telnet, мы занимаем одну виртуальную терминальную линию vty:

```
line vty
access-class localhost
```

Внесение изменений в конфигурацию устройства производится в режиме глобального конфигурирования (config). Вход в режим осуществляется командой `configure terminal` или сокращённо `conf t`:

```
router# configure terminal
router(config)#
```

4.3. Настройка маршрутизатора R1

Для работы протокола BGP на маршрутизаторе R1 с помощью демона zebra настраиваются все сетевые интерфейсы, задействованные в работе протокола BGP. Подключаемся к zebra и входим в режим глобального конфигурирования:

```
[root@alt219 ~]# telnet 127.0.0.1 2601
Trying 127.0.0.1...
Connected to 127.0.0.1.
User Access Verification
Password:
Router>
Router> enable
Router#
Router# configure terminal
Router(config)#
```

Изменим имя устройства:

```
Router(config)# hostname R1
```

В процессе работы протокола BGP будет задействован интерфейс маршрутизатора enp1s1f0.

Настройка интерфейса enp1s1f0:

`R1(config)#interface enp1s1f0` – переход к конфигурированию интерфейса;

`R1(config-if)#ip address 10.10.11.1/24` – присваиваем адрес и маску подсети;

`R1(config-if)#exit` – выход из режима настройки интерфейса enp1s1f0.

```
R1(config)#
```

Настройка интерфейса enp1s1f1:

```
R1(config)#interface enp1s1f1
R1(config-if)# ip address 192.168.4.1/24
R1(config-if)# exit
```

Настройка логического loopback-интерфейса lo:

```
R1(config)#interface lo
R1(config-if)# ip address 192.168.1.1/30
R1(config-if)# exit
```

Выход из режима глобального конфигурирования осуществляется с помощью команды `exit` или комбинации Ctrl+C.

Чтобы сохранить все произведенные настройки, нужно сохранить текущую конфигурацию zebra командой `write memory` или сокращенно `wr`:

```
R1(config)# write memory
Configuration saved to /etc/quagga/zebra.conf
R1(config)# exit
```

Вся конфигурация была перезаписана в /etc/quagga/zebra.conf.

Настройка демона zebra завершена, отключиться от демона можно с помощью комбинации Ctrl+D.

Далее на маршрутизаторе R1 необходимо настроить работу протокола BGP с помощью демона bgpd. Подключаемся к bgpd по telnet:

```
[root@alt219 ~]# telnet 127.0.0.1 2605
Trying 127.0.0.1...
Connected to 127.0.0.1.
User Access Verification
Password:
router-BGP>
router-BGP> enable
router-BGP#
```

Командой `show running-config` можно посмотреть текущую конфигурацию, записанную в /etc/quagga/bgpd.conf. Для внесения изменений в конфигурацию, переходим в режим глобального конфигурирования:

```
router-BGP# configure terminal
router-BGP(config)#
```

Изменим имя устройства:

```
router-BGP(config)# hostname R1-BGP
```

Теперь необходимо включить процесс BGP на маршрутизаторе и назначить номер автономной системы в нашей сети, а также прописать сети, которые будет анонсировать данный маршрутизатор.

Включаем BGP и задаем номер автономной системы AS 65001, переходим к его конфигурированию:

```
R1-BGP(config)# router bgp 65001
R1-BGP(config-router)#
```

Для работы протоколу BGP сначала необходимо установить соседские отношения с подключёнными непосредственно к нему маршрутизаторами. Соседи прописываются вручную, указывается IP-адрес и номер AS соседа. В нашей сети соседом маршрутизатора R1 является маршрутизатор R2 и маршрутизатор R4, соответственно, указываются их IP-адреса и номер их автономных систем AS:

```
R1-BGP(config-router)# neighbor 10.10.11.2 remote-as 65002
R1-BGP(config-router)# neighbor 10.10.14.4 remote-as 65004
```

Далее в формате `network <x.x.x.x> mask <m.m.m.m>` прописываются сети, которые необходимо анонсировать по протоколу BGP, в нашем случае за сеть в AS отвечает loopback-адрес маршрутизатора:

```
R1-BGP(config-router)# network 192.168.1.0 mask 255.255.255.252
R1-BGP(config-router)# exit
```

Настройка демона bgpd завершена, необходимо сохранить конфигурацию. Отключиться от демона можно комбинации Ctrl+D.

```
R1-BGP(config)# write memory
```

```
Configuration saved to /etc/quagga/bgpd.conf
R1-BGP(config)# exit
```

4.4. Настройка маршрутизатора R2

Теперь настраивается маршрутизатор R2. Для работы протокола BGP на маршрутизаторе R2 с помощью демона zebra настраиваются все сетевые интерфейсы, задействованные в работе протокола BGP. Подключаемся к zebra и входим в режим глобального конфигурирования:

```
[root@alt218 ~]# systemctl start zebra
[root@alt218 ~]# telnet 127.0.0.1 2601
Trying 127.0.0.1...
Connected to 127.0.0.1.
User Access Verification
Password:
Router>
Router enable
Router#
Router# configure terminal
Router(config)#
```

Изменим имя устройства:

```
Router(config)# hostname R2
```

Настройка интерфейса enp1s1f0:

```
R2(config)#interface enp1s1f0
R2(config-if)# ip address 10.10.11.2/24
R2(config-if)# exit
```

Настроим интерфейс enp1s1f1:

```
R2(config)# interface enp1s1f1
R2(config-if)# ip address 10.10.12.2/24
R2(config-if)# exit
```

Настройка логического loopback-интерфейса lo:

```
R2(config)#interface lo
R2(config-if)# ip address 192.168.2.1/30
R2(config-if)# exit
```

Настройка демона zebra завершена, необходимо сохранить конфигурацию. Отключиться от демона можно комбинации Ctrl+D.

```
R2(config)# write memory
Configuration saved to /etc/quagga/zebra.conf
R2(config)# exit
```

Далее на маршрутизаторе R2 необходимо настроить работу протокола BGP с помощью демона bgpd. Запускаем демон, подключаемся к bgpd по telnet и входим в режим глобального конфигурирования:

```
[root@alt218 ~]# systemctl start bgpd
[root@alt218 ~]# telnet 127.0.0.1 2605
Trying 127.0.0.1...
Connected to 127.0.0.1.
User Access Verification
Password:
router-BGP>
router-BGP> enable
router-BGP#
```

Для внесения изменений в конфигурацию, переходим в режим глобального конфигурирования:

```
router-BGP# configure terminal
router-BGP(config)#
```

Изменим имя устройства:

```
router-BGP(config)# hostname R2-BGP
```

Теперь необходимо включить процесс BGP на маршрутизаторе и назначить номер автономной системы в нашей сети, а также прописать сети, которые будет анонсировать данный маршрутизатор.

Включаем BGP и задаем номер автономной системы AS 65002, переходим к его конфигурированию;

```
R2-BGP(config)# router bgp 65002
R2-BGP(config-router)#
```

Для работы протоколу BGP сначала необходимо установить соседские отношения с подключёнными непосредственно к нему маршрутизаторами.

В нашей сети соседями маршрутизатора R2 являются маршрутизаторы R1 и R3, соответственно, указываются их IP-адреса и номера AS:

```
R2-BGP(config-router)# neighbor 10.10.11.1 remote-as
65001
R2-BGP(config-router)# neighbor 10.10.12.3 remote-as
65003
```

Далее в формате `network <x.x.x.x> mask <m.m.m.m>` прописываются сети, которые необходимо анонсировать по протоколу BGP, в нашем случае за сеть в AS отвечает loopback адрес маршрутизатора:

```
R2-BGP(config-router)# network 192.168.2.0 mask
255.255.255.252
```

Так как R2 находится в транзитной автономной системе между R1 и R3, необходимо прописать транзитные сети между ними:

```
R2-BGP(config-router)#network 10.10.11.0 mask
255.255.255.0
R2-BGP(config-router)#network 10.10.12.0 mask
255.255.255.0
```

Настройка демона bgpd завершена, необходимо сохранить конфигурацию. Отключиться от демона можно комбинации Ctrl+D.

```
R2-BGP(config)# write memory
Configuration saved to /etc/quagga/bgpd.conf
R2-BGP(config)# exit
```

4.5. Анализ работы протокола BGP

После настроек маршрутизаторы R1 и R2 начнут процесс установления соседских отношений, инициируя TCP-сессию по 179 порту. Проверим работу протокола BGP на R1 с помощью анализатора трафика, для этого на интерфейсе enp1s1f0 запустим Wireshark и зададим фильтр по протоколу TCP:

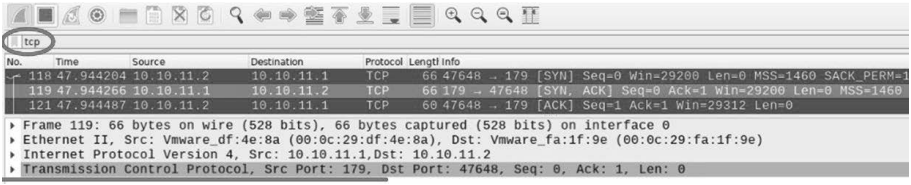


Рисунок 3. Процесс установления TCP-сессии

R1 отправляет SYN на 179 порт R2, R2 подтверждает установление сессии, отвечая R1 SYN+ACK, R1 отвечает подтверждением ACK. Таким образом происходит тройное рукопожатие, после чего TCP-сессия установлена.

После того, как TCP-сессия установлена, маршрутизаторы BGP начинают обмен сообщениями OPEN. OPEN – первый тип сообщений BGP, они отсылаются только в самом начале BGP-сессии для согласования параметров. В нём передаются версия протокола, номер AS, Hold Timer и Router ID. Чтобы BGP-сессия установилась, должны соблюдаться следующие условия:

Версии протокола должна быть одинаковой;

Номера AS в сообщении OPEN должны совпадать с настройками на удалённой стороне;

Router ID должны различаться.

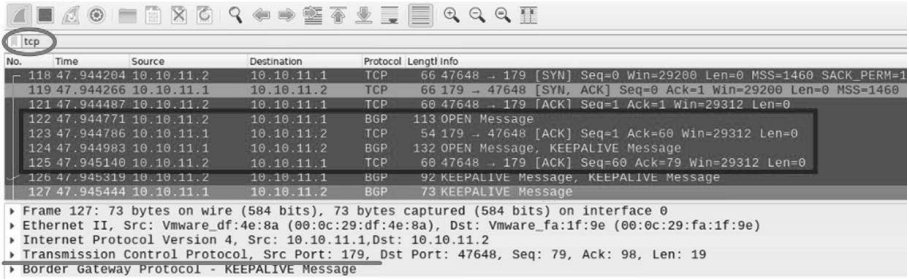


Рисунок 4. Процесс обмена сообщениями OPEN

Получив OPEN от R1, R2 отправляет свой OPEN, а также KEEPALIVE, говорящий о том, что OPEN от R1 получен – это сигнал для R1 переходить к следующему состоянию – ESTABLISHED.

После всех шагов маршрутизаторы переходят в стабильное состояние ESTABLISHED, означающее, что запущена правильная версия BGP и все настройки идентичны.

Для каждого соседа можно посмотреть Uptime – как долго он находится в состоянии ESTABLISHED, для этого подключимся к bgpd и в режиме enable и используя команду `show ip bgp neighbors`:

```
[root@alt219 ~]# telnet 127.0.0.1 2605
R1-BGP>
R1-BGP> enable
R1-BGP#
R1-BGP# show ip bgp neighbors
```

BGP neighbor is 10.10.11.2, remote AS 65002, local AS 65001, external link

BGP version 4, **remote router ID 192.168.2.1**

BGP state = Established, up for 00:18:53

Last read 00:00:53, hold time is 180, keepalive interval is 60 seconds

Команда выводит IP-адрес соседнего маршрутизатора, номер его AS и номер AS на локальном маршрутизаторе. Также показывается версия протокола BGP (4 версия для IPv4), router ID соседа, Uptime и таймеры рассылки сообщений keepalive и таймер удержания сессии Hold, по истечении которого сосед будет считаться недоступным.

После того как все состояния успешно пройдены, BGP-маршрутизаторы регулярно будут рассылать сообщения KEEPALIVE. Эти сообщения означают, что маршрутизатор находится в рабочем состоянии. Это происходит с истечением таймера Keepalive – по умолчанию 60 секунд.

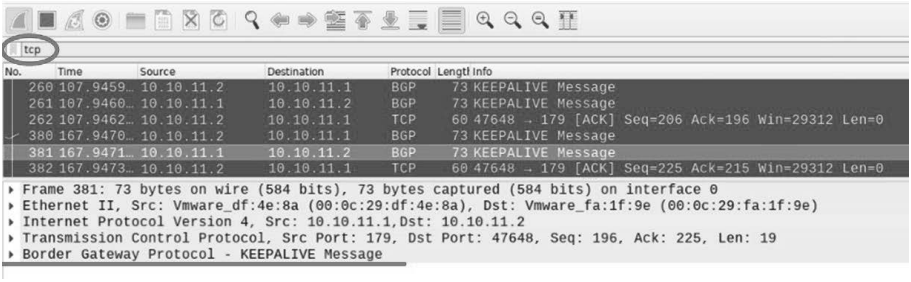


Рисунок 5. Процесс отправки сообщений KEEPALIVE

Как только происходят какие-либо изменения в сети, начинается обмен маршрутной информацией. Для это используются сообщения UPDATE. Каждое сообщение UPDATE одновременно может нести информацию об одном новом маршруте или о удалении группы старых.

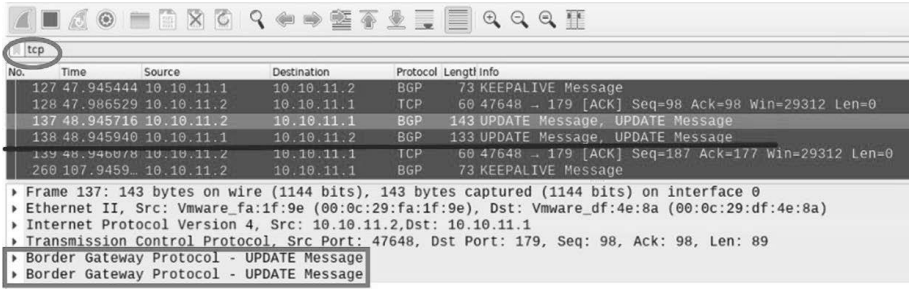


Рисунок 6. Обмен маршрутной информацией в пакетах UPDATE

4.6. Настройка маршрутизатора R3

Теперь настраивается маршрутизатор R3. Для работы протокола BGP на маршрутизаторе R3 с помощью демона zebra настраиваются все сетевые интерфейсы, задействованные в работе протокола BGP. Подключаемся к zebra и входим в режим глобального конфигурирования:

```
[root@alt217 ~]# systemctl start zebra
[root@alt217 ~]# telnet 127.0.0.1 2601
Trying 127.0.0.1...
Connected to 127.0.0.1.
User Access Verification
Password:
Router>
Router enable
Router#
Router# configure terminal
Router(config)#
```

Изменим имя устройства:

```
Router(config)# hostname R3
```

Настройка интерфейса enp1s1f0:

```
R3(config)#interface enp1s1f0
R3(config-if)# ip address 10.10.13.3/24
R3(config-if)# exit
```

Настройка интерфейса enp1s1f1:

```
R3(config)#interface enp1s1f1
```

```
R3(config-if)# ip address 10.10.12.3/24
R3(config-if)# exit
```

Настройка логического loopback-интерфейса lo:

```
R3(config)#interface lo
R3(config-if)# ip address 192.168.10.1/30
R3(config-if)# ip address 192.168.20.1/30
R3(config-if)# exit
```

Настройка демона zebra завершена, необходимо сохранить конфигурацию. Отключиться от демона можно комбинации Ctrl+D.

```
R3(config)# write memory
Configuration saved to /etc/quagga/zebra.conf
R3(config)# exit
```

Далее на маршрутизаторе R3 необходимо настроить работу протокола BGP с помощью демона bgpd. Запускаем демон, подключаемся к bgpd по telnet и входим в режим глобального конфигурирования:

```
[root@alt217 ~]# systemctl start bgpd
[root@alt217 ~]# telnet 127.0.0.1 2605
Trying 127.0.0.1...
Connected to 127.0.0.1.
User Access Verification
Password:
router-BGP>
router-BGP> enable
```

Для внесения изменений в конфигурацию, переходим в режим глобального конфигурирования:

```
router-BGP# configure terminal
router-BGP(config)#
```

Изменим имя устройства:

```
router-BGP(config)# hostname R3-BGP
```

Теперь необходимо включить процесс BGP на маршрутизаторе и назначить номер автономной системы в нашей сети, а также прописать сети, которые будет анонсировать данный маршрутизатор.

Включаем BGP и задаем номер автономной системы AS 65003, переходим к его конфигурированию;

```
R3-BGP(config)# router bgp 65003
R3-BGP(config-router)#
```

Для работы протоколу BGP сначала необходимо установить соседские отношения с подключёнными непосредственно к нему маршрутизаторами.

В нашей сети соседями маршрутизатора R3 является маршрутизатор R2 и R4, соответственно, указывается их IP-адреса и номера AS:

```
R3-BGP(config-router)# neighbor 10.10.12.2 remote-as 65002
R3-BGP(config-router)# neighbor 10.10.13.4 remote-as 65004
```

Далее в формате `network <x.x.x.x> mask <m.m.m.m>` прописываются сети, которые необходимо анонсировать по протоколу BGP, в нашем случае за сети в AS отвечает loopback адрес маршрутизатора:

```
R3-BGP(config-router)# network 192.168.10.0 mask 255.255.255.252
R3-BGP(config-router)# network 192.168.20.0 mask 255.255.255.252
R3-BGP(config-router)#exit
R3-BGP(config)#
```

Настройка демона bgpd завершена, необходимо сохранить конфигурацию. Отключиться от демона можно комбинации Ctrl+D.

```
R3-BGP(config)# write memory
Configuration saved to /etc/quagga/bgpd.conf
R3-BGP(config)# exit
```

4.7. Настройка маршрутизатора R4

Теперь настраивается маршрутизатор R4. Для работы протокола BGP на маршрутизаторе R4 с помощью демона zebra настраиваются все сетевые интерфейсы, задействованные в работе протокола BGP. Подключаемся к zebra и входим в режим глобального конфигурирования:

```
[root@alt216 ~]# systemctl start zebra
[root@alt216 ~]# telnet 127.0.0.1 2601
Trying 127.0.0.1...
```

```
Connected to 127.0.0.1.
User Access Verification
Password:
Router>
Router enable
Router#
Router# configure terminal
Router(config)#
```

Изменим имя устройства:

```
Router(config)# hostname R4
```

Настройка интерфейса enp1s1f0:

```
R4(config)#interface enp1s1f0
R4(config-if)# ip address 10.10.13.4/24
R4(config-if)# exit
```

Настроим интерфейс enp1s1f1:

```
R4(config)# interface enp1s1f1
R4(config-if)# ip address 10.10.14.4/24
R4(config-if)# exit
```

Настройка логического loopback-интерфейса lo:

```
R4(config)#interface lo
R4(config-if)# ip address 192.168.4.1/30
R4(config-if)# exit
```

Настройка демона zebra завершена, необходимо сохранить конфигурацию. Отключиться от демона можно комбинации Ctrl+D.

```
R4(config)# write memory
Configuration saved to /etc/quagga/zebra.conf
R4(config)# exit
```

Далее на маршрутизаторе R4 необходимо настроить работу протокола BGP с помощью демона bgpd. Запускаем демон, подключаемся к bgpd по telnet и входим в режим глобального конфигурирования:

```
[root@alt216 ~]# systemctl start bgpd
[root@alt216 ~]# telnet 127.0.0.1 2605
Trying 127.0.0.1...
```

```
Connected to 127.0.0.1.
User Access Verification
Password:
router-BGP>
router-BGP> enable
router-BGP#
```

Для внесения изменений в конфигурацию, переходим в режим глобального конфигурирования:

```
router-BGP# configure terminal
router-BGP(config)#
```

Изменим имя устройства:

```
router-BGP(config)# hostname R4-BGP
```

Теперь необходимо включить процесс BGP на маршрутизаторе и назначить номер автономной системы в нашей сети, а также прописать сети, которые будет анонсировать данный маршрутизатор.

Включаем BGP и задаем номер автономной системы AS 65004, переходим к его конфигурированию;

```
R4-BGP(config)# router bgp 65004
R4-BGP(config-router)#
```

Для работы протоколу BGP сначала необходимо установить соседские отношения с подключёнными непосредственно к нему маршрутизаторами.

В нашей сети соседями маршрутизатора R4 являются маршрутизаторы R1 и R3, соответственно, указываются их IP-адреса и номера AS:

```
R4-BGP(config-router)# neighbor 10.10.14.1 remote-as 65001
R4-BGP(config-router)# neighbor 10.10.13.3 remote-as 65003
```

Далее в формате `network <x.x.x.x> mask <m.m.m.m>` прописываются сети, которые необходимо анонсировать по протоколу BGP, в нашем случае за сеть в AS отвечает loopback адрес маршрутизатора:

```
R4-BGP(config-router)# network 192.168.4.0 mask 255.255.255.252
```

Так как R4 находится в транзитной автономной системе между R1 и R3, необходимо прописать транзитные сети между ними:

```
R4-BGP(config-router)#network 10.10.13.0 mask 255.255.255.0
R4-BGP(config-router)#network 10.10.14.0 mask 255.255.255.0
```

Настройка демона bgpd завершена, необходимо сохранить конфигурацию. Отключиться от демона можно комбинации Ctrl+D.

```
R4-BGP(config)# write memory
Configuration saved to /etc/quagga/bgpd.conf
R4-BGP(config)# exit
```

4.8. Проверка работы протокола BGP

После настроек все маршрутизаторы в сети установят BGP-сессии между собой и будут с помощью анонсов BGP анонсировать друг другу префиксы (блоки адресов).

Проверим все полученные маршруты на R1 командой `show ip bgp`:

```
[root@alt219 ~]# telnet 127.0.0.1 2605
R1-BGP>
R1-BGP> enable
R1-BGP# show ip bgp
BGP table version is 0, local router ID is 192.168.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath
Network          Next Hop    Metric LocPrf Weight Path
*  10.10.11.0/24     10.10.11.2  0      0 65002 i
*>                   0.0.0.0     0          32768 i
*  10.10.12.0/24     10.10.14.4           0 65004 65003 i
*>                   10.10.11.2  0      0 65002 i
*  10.10.13.0/24     10.10.11.2           0 65002 65003 i
*>                   10.10.14.4  0      0 65004 i
*  10.10.14.0/24     10.10.14.4  0      0 65004 i
*>                   0.0.0.0     0          32768 i
*> 192.168.1.0/30   0.0.0.0     0          32768 i
```

```
*  192.168.2.0/30   10.10.11.2  0      0 65002 i
*>                  10.10.14.4  0      0 65004 i
*  192.168.10.0/30  10.10.11.2         0 65002 65003 i
*>                  10.10.14.4         0 65004 65003 i
*  192.168.20.0/30  10.10.11.2         0 65002 65003 i
*>                  10.10.14.4         0 65004 65003 i
```

Таблица маршрутов BGP показывает все известные сети, полученные по протоколу BGP (столбец `Network`), следующий маршрутизатор на маршруте к подсети (столбец `Next Hop`), путь до сети через номера автономных систем (столбец `Path`), а также остальные атрибуты пути, такие как метрика, вес и приоритет маршрута `LocPrf`.

Для того, чтобы проверить расширенную статистику BGP, используется команда `show ip bgp summary`. Выполним ее на маршрутизаторе R1:

```
R1-BGP# show ip bgp summary
BGP router identifier 192.168.1.1, local AS number
65001 RIB entries 17, using 1904 bytes of memory
Peers 2, using 14 KiB of memory
Neighbor       V     AS       MsgRcvd  MsgSent   Up/Down
State/PfxRcd
10.10.11.2     4  65002        46        59       00:11:32
8
10.10.14.4     4  65004        38        44       00:10:14
8
```

Команда показывает информацию о соседних маршрутизаторах, количество принятых и отправленных сообщений (`MsgRcvd/MsgSent`), время в активном состоянии `(Up/Down)`, а также количество принятых префиксов (маршрутов) – `State/PfxRcd`.

Чтобы проверить работу сети, организуем обмен трафиком между автономными системами AS 65001 и AS 65003. С маршрутизатора R1 проверим доступность сети 192.168.10.0/30, находящейся в автономной системе AS 65003 маршрутизатора R3:

```
[root@alt219 ~]# ping 192.168.10.1 -c 3
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
64 bytes from 192.168.10.1: icmp_req=1 ttl=63
time=0.362 ms
```

```
    64 bytes from 192.168.10.1: icmp_req=2 ttl=63
time=0.345 ms
    64 bytes from 192.168.10.1: icmp_req=3 ttl=63
time=0.356 ms
    --- 192.168.10.1 ping statistics ---
    3 packets transmitted, 3 received, 0% packet loss, time
4103ms
    rtt min/avg/max/mdev = 0.315/0.343/0.362/0.020 ms
```

Также проверим доступность сети 192.168.20.0/30:

```
    [root@alt219 ~]# ping 192.168.20.1 -c 3
    PING 192.168.20.1 (192.168.20.1) 56(84) bytes of data.
    64 bytes from 192.168.20.1: icmp_req=1 ttl=63
time=0.331 ms
    64 bytes from 192.168.20.1: icmp_req=2 ttl=63
time=0.352 ms
    64 bytes from 192.168.20.1: icmp_req=3 ttl=63
time=0.343 ms
    --- 192.168.20.1 ping statistics ---
    3 packets transmitted, 3 received, 0% packet loss, time
4103ms
    rtt min/avg/max/mdev = 0.317/0.353/0.345/0.021 ms
```

Ключ –c (count) задает количество посылаемых ICMP запросов. Сети 192.168.10.0/30 и 192.168.20.0/30 в автономной системе AS 65003 доступны, ICMP пакеты успешно возвращаются.

4.9. Распределение трафика в BGP

4.9.1. Фильтрация маршрутов с помощью Prefix-list

Prefix-list или списки префиксов представляют собой списки управления маршрутной информацией, в которых можно что-либо запретить или разрешить.

Для создания списка префиксов используется следующий синтаксис:

`ip prefix-list` <list-name> `[seq` <value>`]` <deny | permit> <network/length> `[ge` <value>`] [le` <value>`]`

List-name – название списка префиксов;

Seq – порядковый номер создаваемого правила в списке префиксов. На соответствие заданным условиям первыми проверяются правила с наименьшим

порядковым номером. Если правило одно, то параметр seq указывать не обязательно, по умолчанию равен 5;

Deny/permit – запретить или разрешить определенные маршруты;

Network/length – запись вида сеть/размер в формате <x.x.x.x>/размер;

Ge и le – не обязательные операторы, необходимы для определения диапазона фильтрации трафика. Ge (Greater or Equal), что означает больше или равно, le (Less or Equal) – меньше или равно.

Пример, запись вида:

```
ip prefix-list PL_test permit 172.16.5.0/24 ge 25 le 26
```

В данном списке префиксов создается разрешающее правило с именем PL_test, которое проверяет первые 24 бита адреса сети (первые три октета) и сравнивает их с входящим анонсом, если первые три октета совпадают с 172.16.5, а маска находится в диапазоне от 25 до 26, то правило выполняется.

Под данное правило попадают следующие сети:

Сеть 172.16.5.0/26 с диапазоном адресов 172.16.5.1 – 172.16.5.62;

Сеть 172.16.5.64/26 с диапазоном адресов 172.16.5.65 – 172.16.5.126;

Сеть 172.16.5.128/26 с диапазоном адресов 172.16.5.129 – 172.16.5.190;

Сеть 172.16.5.192/26 с диапазоном адресов 172.16.5.193 – 172.16.5.254;

Сеть 172.16.5.0/25 с диапазоном адресов 172.16.5.1 – 172.16.5.126;

Сеть 172.16.5.128/25 с диапазоном адресов 172.16.5.129 – 172.16.5.254.

После формирования списка префиксов необходимо применить его к BGP-соседу на входящий или исходящий трафик. Список применяется следующим образом:

```
router bgp <Номер автономной системы>
neighbor <IP-адрес BGP-соседа> prefix-list <Имя списка> in/out
```

После того, как прописаны префикс листы на входящий трафик, политики фильтрации еще не будут работать, необходимо перезапросить маршрутную информацию от соседа. Существует несколько способов сделать это:

Hard reset – жесткий сброс. Выполняется из привилегированного режима (enable) командой `clear ip bgp *`:

```
Router-BGP# clear ip bgp *
```

При этом разрываются все BGP-сессии со всеми соседями, очищается таблица BGP и маршрутизатору необходимо будет заново установить BGP-

сессии с соседями и выучить всю маршрутную информацию. Выполнить жесткий сброс для определенного соседа можно командой `clear ip bgp` <IP-адрес соседа>.

Soft reset – мягкий сброс. Позволяет обновить политики маршрутизации, не разрывая BGP-сессию с соседом:

```
Router-BGP# conf t
Router-BGP(config)# router bgp <Номер автономной системы>
Router-BGP(config-router)# neighbor <IP-адрес> soft-reconfiguration inbound
```

При таком способе обновления маршрутизатор сохранит BGP сессию с соседом и всю маршрутную информацию, полученную от него. При этом обновятся входящие политики фильтрации.

Если у маршрутизатора немного BGP соседей, и они анонсируют не большое количество маршрутной информации, то можно использовать жесткий сброс. Если же в сети много соседей, от которых приходит много маршрутов, то реконфигурация таким способом может сильно нагрузить маршрутизатор и занять длительное время, поэтому в таком случае лучше использовать мягкий сброс.

На данный момент маршрутизатор R1 получает анонсы сети 192.168.10.0/30 от своих соседей 10.10.11.2 (маршрутизатор R2) и 10.10.14.4 (маршрутизатор R4). Это можно увидеть из таблицы BGP в поле Next Hop:

```
[root@alt219 ~]# telnet 127.0.0.1 2605
R1-BGP>
R1-BGP> enable
R1-BGP# show ip bgp
BGP table version is 0, local router ID is 192.168.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, S Stale, R Removed
Network          Next Hop    Metric LocPrf Weight  Path
* 192.168.10.0/30 10.10.11.2         0    65002 65003 i
*>                10.10.14.4         0    65004 65003 i
```

Для настроенной лабораторной сети с помощью списка префиксов запретим принимать BGP-анонсы сети 192.168.10.0/30 от соседа R2, находящегося в автономной системе AS 65002. Для этого настроим соответствующий список префиксов:

```
R1-BGP# conf t
R1-BGP(config)# ip prefix-list from_R2 seq 5 deny
192.168.10.0/30
R1-BGP(config)# ip prefix-list from_R2 seq 10 permit
0.0.0.0/0 le 32
```

Создаётся список префиксов с именем from_R2, содержащий в себе 2 правила.

Первое правило имеет номер последовательности 5 и запрещает прием BGP-анонсов сети 192.168.10.0/30. Второе правило с номером последовательности 10 разрешает прием всех подсетей с любой длиной маски (0-32).

Когда трафик приходит на маршрутизатор, то сначала проверяется первым правилом с наименьшим номером последовательности. Если пришел анонс о запрещенной сети, то трафик, соответственно, сразу попадает под первое правило и дальнейшая обработка завершается, а запрещенные анонсы не попадают в BGP таблицу маршрутизатора.

Приходящий анонс о любой другой сети также сначала обрабатывается первым правилом и не попадает под него, после переходит на обработку вторым правилом, в котором разрешены любые сети и, соответственно, уже под него попадает, после чего обработка трафика завершается и все не запрещенные маршруты попадают в таблицу BGP.

После того как список префиксов сформирован, необходимо применить его к соседу R2. Так как маршрутизатор R1 принимает анонсы о сети 192.168.10.0/30 от R2, то применяем prefix-list на входящий трафик (in):

```
R1-BGP(config)# router bgp 65001
R1-BGP(config-router)# neighbor 10.10.11.2 prefix-list
from_R2 in
R1-BGP(config-router)# exit
```

Чтобы политики фильтрации применились, выполняем перезапрос маршрутной информации у маршрутизатора R2:

```
R1-BGP# clear ip bgp 10.10.11.2
```

После произойдет переустановление BGP сессии, R1 получит анонсы от R2 и обновит свои политики фильтрации трафика. С помощью команды `show ip bgp` проверим какие префиксы получил R1. Часть строк таблицы опущена для краткости:

```
R1-BGP# show ip bgp
```

```
BGP table version is 0, local router ID is 192.168.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath
Origin codes: i - IGP, e - EGP, ? - incomplete
 Network          Next Hop  Metric  LocPrf Weight Path
*> 192.168.1.0/30  0.0.0.0     0      32768 i
*  192.168.2.0/30  10.10.11.2   0     0 65002 i
*>                 10.10.14.4   0     0 65004 i
*> 192.168.10.0/30 10.10.14.4          0 65004 65003 i
*  192.168.20.0/30 10.10.11.2          0 65002 65003 i
*>             10.10.14.4              0 65004 65003 i
```

Как видно из таблицы BGP, маршрутизатор R1 получил анонс сети 192.168.10.0/30, но теперь эта сеть доступна только через 10.10.14.4 (маршрутизатор R4), так как анонс от R2 об этой сети пришел на маршрутизатор, но был отфильтрован списком префиксов и не добавлен в таблицу BGP.

Теперь трафик до сети 192.168.10.1 идет только через AS 65004:

```
[root@alt219 ~]# traceroute 192.168.10.1
traceroute to 192.168.10.1 (192.168.10.1), 30 hops max, 60 byte packets
 1  10.10.14.4          0.120 ms  0.078 ms  0.050 ms
 192.168.10.1       0.122 ms  0.083 ms  0.105 ms
```

4.9.2. Распределение маршрутов BGP с помощью Route map

Одним из способов распределения трафика в сетях, работающих по протоколу BGP, являются карты маршрутов route map. В контексте BGP карта маршрутов представляет собой способ контроля изменения информации о маршрутах. Распределение маршрутов применяется, например, когда необходимо организовать высокую стабильность сервисов, перенаправляя трафик через заведомо более надежные участки сети.

Синтаксис настройки карты маршрутов выглядит следующим образом:

```
route-map <имя карты> <deny | permit> <seq>
<match {критерий}>
<set {критерий}>
deny | permit – разрешающее или запрещающее правило
```

seq – порядковый номер карты маршрутов. При применении карты маршрутов к входящим или исходящим маршрутам, первый набор условий применяется через экземпляр с наименьшим номером. Если первый набор

условий не удовлетворяется, происходит переход к экземпляру карты маршрутов с более высоким номером.

Все карты маршрутов состоят из списка команд настройки `match` и `set`. Командой `match` задаются критерии соответствия, а командой `set` задаются действия, выполняемые при удовлетворении условий, задаваемых командой `match`.

В таблице 1 указаны некоторые основные критерии для команды соответствия `match`.

Таблица 1. Критерии соответствия и команды конфигурирования

Критерий соответствия	**Команда конфигурации**
Network/mask	match ip address prefix-list
AS-path	match as-path
BGP next-hop address	match ip next-hop

`Network/mask` – соответствие по сети и маске;

`AS-path` – соответствие по атрибуту пути AS-path;

`BGP next-hop address` – соответствие по следующему маршрутизатору BGP.

В таблице 2 указаны некоторые основные атрибуты пути для команды действия `set`:

Таблица 2. Атрибуты и команды конфигурации

Атрибуты пути	**Команда конфигурации**
Weight	set weight
Local Preference	set local-preference
MED	set metric
BGP next-hop	set next-hop

`Weight` – установка стоимости пути;

`Local Preference` – установка локального приоритета;

`MED` – установка метрики;

`BGP next-hop` - установка следующего BGP-маршрутизатора.

В нашей сети настроим пути трафика таким образом, чтобы трафик до подсети 192.168.10.0/30 шел через автономную систему AS 65002, а трафик до сети 192.168.20.0/30 шел через автономную систему AS 65004. Для этого воспользуемся рассмотренными ранее списками префиксов и картами маршрутов.

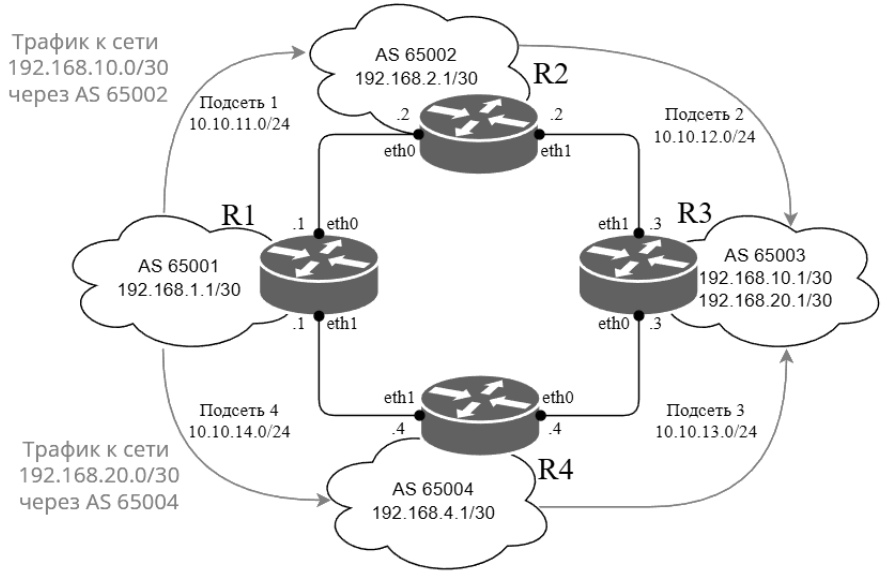


Рисунок 7. Организация распределения трафика в сети

Для начала отвяжем от соседа R2 настроенный ранее список префиксов, запрещающий принимать от него анонсы сети 192.168.10.0/30:

```
R1-BGP# conf terminal
R1-BGP(config)# router bgp 65001
R1-BGP(config-router)# no neighbor 10.10.11.2 prefix-list from_R2 in
R1-BGP(config-router)# exit
R1-BGP(config)# exit
```

Перезапросим маршруты у R2 и обновим политики фильтрации:

```
R1-BGP#
R1-BGP# clear ip bgp 10.10.11.2
```

Теперь R1 снова получает анонсы сети 192.168.10.0/30 от R2 и доступен через автономные системы 65002 и 65003:

```
R1-BGP# show ip bgp
BGP table version is 0, local router ID is 192.168.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, S Stale, R Removed
Network          Next Hop    Metric LocPrf Weight  Path

* 192.168.10.0/30 10.10.11.2           0    65002 65003 i
*>               10.10.14.4           0    65004 65003 i
*  192.168.20.0/30 10.10.11.2          0    65002 65003 i
*>                10.10.14.4           0    65004 65003 i
```

Для реализации распределения трафика создадим список префиксов, которым выделим анонсы сети 192.168.20.0/30 от R2:

```
R1-BGP# conf terminal
R1-BGP(config)# ip prefix-list PL_from_R2 seq 5 permit 192.168.20.0/30
```

Далее создадим карты маршрутов, в которых разрешим получение анонсов, попадающих под правила созданного списка префиксов. Для этого воспользуемся критерием соответствия `Network/mask`:

```
R1-BGP(config)# route-map for_AS65002 permit 10
R1-BGP(config-route-map)# match ip address prefix-list PL_from_R2
```

Воспользуемся атрибутом пути `Local preference` для установки приоритета маршрута, попадающего под правило списка префикса:

```
R1-BGP(config-route-map)# set local-preference 50
R1-BGP(config-route-map)# exit
```

Создадим второй экземпляр этой карты маршрута для всех остальных приходящих анонсов, но уже с порядковым номером 20:

```
R1-BGP(config)# route-map for_AS65002 permit 20
```

Также воспользуемся атрибутом пути `Local preference` для установки приоритета всех остальных маршрутов на R1:

```
R1-BGP(config-route-map)# set local-preference 100
R1-BGP(config-route-map)# exit
```

Рассмотрим подробнее произведенные настройки. Создана карта маршрута с именем for_AS65002, имеющая порядковый номер 10, т.е. сначала ко всему приходящему трафику будет применяться набор действий, описанный в экземпляре карты с меньшим приоритетом.

В первом экземпляре карты маршрута командой match задается критерий соответствия `Network/mask`, для которого используется команда `ip address prefix-list`, проверяющая соответствие списку префикса PL_from_R2.

Если условие `match` выполняется, то командой `set` устанавливается приоритет маршрута для сети, удовлетворяющей правилу заданного списка префиксов с помощью атрибута пути `local-preference`. Атрибут пути `local-preference` имеет числовые значения, чем выше значение атрибута пути, тем этот путь приоритетнее для трафика.

Второй экземпляр карты маршрута необходим для всех остальных анонсов, не попадающих под правила списка префиксов, этому трафику устанавливается другое значение атрибута пути `local preference`.

После того как список карта маршрута сформирована, необходимо применить ее к соседу R2. Так как маршрутизатор R1 принимает анонсы о сети 192.168.20.0/30 от R2, то применяем карту маршрута на входящий трафик (in):

```
R1-BGP(config)# router bgp 65001
R1-BGP(config-router)# neighbor 10.10.11.2 route-map
for_AS65002 in
R1-BGP(config-router)# exit
```

Рассмотрим подробнее как работает данная карта маршрута:

- Маршрутизатор R3 анонсирует сеть 192.168.20.0/30 маршрутизатору R2, а R2 анонсирует ее маршрутизатору R1;
- Приходя на маршрутизатор R1, анонс сети 192.168.20.0/30 проверяется первым экземпляром карты маршрутов и удовлетворяет правилу списка префиксов PL_from_R2, так как данный список выделяет сеть 192.168.20.0/30;

- Так как анонс попал под правило списка префиксов, то к нему применяется команда `set`, которая устанавливает приоритет данного полученного маршрута на значение 50, а дальнейшая обработка анонса завершается ввиду выполненного условия;
- Если на маршрутизатор R1 от R2 приходит анонс любой другой сети, то сначала он также проверяется первым экземпляром карты маршрута, под правила которой не попадает, так как не соответствует сети 192.168.20.0/30;
- Далее анонс проверяется вторым экземпляром карты маршрутов, в которой не производится никаких проверок, а сразу командой `set` устанавливается приоритет маршрута на значение 100.

Таким образом, данной картой маршрута мы понизили приоритет трафика до сети 192.168.20.0/30 через AS65002.

Теперь произведем аналогичные настройки для анонсов от маршрутизатора R4. Создадим список прификсов и карту маршрутов, в которой понизим приоритет трафика до сети 192.168.10.0/30 через AS65004, а приоритет всех остальных маршрутов повысим.

Настроим список префиксов, которым выделим анонсы сети 192.168.10.0/30 от маршрутизатора R4:

```
R1-BGP# conf terminal
R1-BGP(config)# ip prefix-list PL_from_R4 seq 5 permit 192.168.10.0/30
```

Далее создадим карты маршрутов, в которых разрешим получение анонсов, попадающих под правила созданного списка префиксов. Для этого воспользуемся критерием соответствия `Network/mask`:

```
R1-BGP(config)# route-map for_AS65004 permit 10
R1-BGP(config-route-map)# match ip address prefix-list PL_from_R4
```

Установка приоритета для выделенных анонсов:

```
R1-BGP(config-route-map)# set local-preference 50
R1-BGP(config-route-map)# exit
```

Создадим второй экземпляр этой карты маршрутов для всех остальных приходящих анонсов, но уже с порядковым номером 20 и установим для них приоритет 100:

```
R1-BGP(config)# route-map for_AS65004 permit 20
```

```
R1-BGP(config-route-map)# set local-preference 100
R1-BGP(config-route-map)# exit
```

Применим созданную карту маршрутов к соседу R4. Так как маршрутизатор R1 принимает анонсы о сети 192.168.10.0/30 от R4, то применяем карту маршрута на входящий трафик (in):

```
R1-BGP(config)# router bgp 65001
R1-BGP(config-router)# neighbor 10.10.14.4 route-map fpr_AS65004 in
R1-BGP(config-router)# exit
R1-BGP(config)# exit
```

После сконфигурированных карт маршрутов, перезапустим BGP-сессии с маршрутизаторами R2 и R4, обновив тем самым настроенные политики распределения трафика:

```
R1-BGP# clear ip bgp *
```

Когда BGP-сессии восстановятся, R1 получит анонсы всех сетей от своих соседей. Анонсы обработаются картами маршрутов и маршрутизатор сформирует новую таблицу BGP уже содержащую настроенные приоритеты трафика:

```
R1-BGP# show ip bgp
BGP table version is 0, local router ID is 192.168.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, S Stale, R Removed
   Network              Next Hop    Metric LocPrf Weight Path
   *> 192.168.1.0/30    0.0.0.0          0    32768 i
   *  192.168.2.0/30  10.10.11.2    0   100        0 65002 i
   *>                 10.10.14.4     0  100        0 65004 i
   *> 192.168.10.0/30 10.10.11.2        100  0 65002 65003 i
   *                  10.10.14.4         50  0 65004 65003 i
   *  192.168.20.0/30 10.10.11.2         50  0 65002 65003 i
```

```
   *>                  10.10.14.4          100  0 65004 65003
i
```

В таблице BGP появились значения в столбце локального приоритета `LocPrf` (Local Preference). Как видно, все маршруты получили свои приоритеты исходя из правил карт маршрутов.

Так маршрут к сети 192.168.10.0/30 через R2 (10.10.11.2) имеет приоритет 100, а через R4 (10.10.14.4) – приоритет 50. Для маршрута к сети 192.168.20.0/30 приоритеты имеют обратное значение. Это позволит распределить исходящий от R1 трафик к подсетям 192.168.10.0/30 и 192.168.20.0/30 через разные автономные системы.

Проверим пути трафика с помощью команды `traceroute`.

Трассировка маршрута до 192.168.10.0/30:

```
   [root@alt219 ~]# traceroute 192.168.10.1
   traceroute to 192.168.10.1 (192.168.10.1), 30 hops max,
60 byte packets
    1  10.10.11.2        0.130 ms  0.092 ms  0.065 ms
    2  192.168.10.1     0.155 ms  0.147 ms  0.152 ms
```

Трассировка маршрута до 192.168.20.0/30:

```
   [root@alt219 ~]# traceroute 192.168.20.1
   traceroute to 192.168.20.1 (192.168.20.1), 30 hops max,
60 byte packets
    1  10.10.14.4         0.125 ms  0.091 ms  0.065 ms
    2  192.168.20.1      0.115 ms  0.095 ms  0.091 ms
```

Как показывает трассировка маршрутов, трафик до сетей 192.168.10.0/30 и 192.168.20.0/30, находящихся в автономной системе AS 65003, распределяется между разными автономными системами.

Практикум № 17
Статическая маршрутизация IPv6

1. ЦЕЛЬ РАБОТЫ

Данная работа предназначена для изучения статической маршрутизации трафика по протоколу IPv6 в сети Интернет, получения практического опыта в работе с маршрутизацией трафика IPv6 в ОС Альт; работа с основными конфигурационными файлами сетевой подсистемы, настройка и управление сетевыми интерфейсами, умение работать с утилитами для диагностики сетевых соединений, работающих по протоколу IPv6.

2. ЗАДАНИЕ НА РАБОТУ

В работе используется топология сети, представленная на рисунке 1. Сеть состоит из трех подсетей, работающих по протоколу IPv6. Маршрутизаторы R2 и R3 являются транзитными маршрутизаторами и осуществляют пересылку пакетов между подсетями 1 и 3, а также выполняют роль шлюзов для маршрутизаторов R1 и R4. Маршрутизаторам R1, R2, R3 и R4 соответствуют рабочие станции под номерами PC1, PC2, PC3 и PC4, соответственно. Адресация в лабораторной сети осуществляется следующим образом:

- Подсеть 1: 2001:fb6:31c0:$X(N_{min})$::/64 или 2001:fb6:31c0:X1::/64
- Подсеть 2: 2001:fb6:31c0:$X(N_{min}+1)$::/64 или 2001:fb6:31c0:X2::/64
- Подсеть 3: 2001:fb6:31c0:$X(N_{min}+2)$::/64 или 2001:fb6:31c0:X3::/64

где $N_{min} = 1$ – наименьший номер из четырех рабочих станций;

X = номер ряда с рабочими станциями, для примера возьмём равное 1.

Наименование интерфейсов указано для примера, на практике оно может отличаться. В нашем же случае `eth0 - enp1s1f0`, а `eth1 - enp1s1f1`.

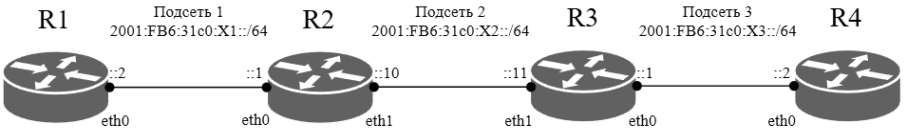


Рисунок 1. Схема сети лабораторной работы

Основные задачи работы:

- Настроить все сетевые интерфейсы машин в соответствии со схемой лабораторной сети;
- Сделать маршрутизаторы R1 и R2 шлюзами по умолчанию для маршрутизаторов R1 и R4, прописав на них соответствующие маршруты;
- На транзитных маршрутизаторах R1 и R2 настроить пересылку пакетов между интерфейсами и прописать маршруты в удаленные подсети;
- Проверить работу сети с помощью утилит диагностики сетевых соединений: ping6, traceroute6.

3. ТЕРМИНЫ И ИСПОЛЬЗУЕМЫЕ КОМАНДЫ

Используемые команды

`cd` – (Change directory) команда перемещения между директориями;

`ls` – (list) команда просмотра содержимого директории;

`nano` – текстовый редактор для создания и редактирования текстовых файлов;

`ip` – утилита для настройки и управления сетевыми интерфейсами. Для работы с протоколом IPv6 используется ключ -6:

`ip -6 address` – отображение информации о сетевых интерфейсах;

`ip -6 route` – отображение информации о маршрутах;

`ip -6 neighbor` – отображение NDP таблицы устройства.

`grep` – утилита для поиска информации;

`ping6` – утилита для проверки сетевых соединений в сетях IPv6;

`traceroute6` – утилита для трассировки маршрута в сетях IPv6;

`mtr6` – утилита для трассировки маршрута, показывает информацию в реальном времени в сетях IPv6.

4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

4.1. Настройка маршрутизатора R1

Для настройки маршрутизации в сети сначала необходимо настроить все сетевые интерфейсы на четырех маршрутизаторах.

Конфигурационные фалы всех сетевых интерфейсов находятся в директории /etc/net/ifaces. Переход в директорию осуществляется командой `cd`:

```
[user@alt219]$ cd /etc/net/ifaces/
```

Воспользуемся командой `ls` для того, чтобы просмотреть содержимое директории:

```
[user@alt219 ifaces]$ ls
```

Вывод команды `ls` покажет директории всех сетевых интерфейсов: enp1s1f0, enp1s1f1 и петлевого loopback-интерфейса lo.

Так как на R1 необходим только интерфейс enp1s1f0, то переходим в его директорию командой `cd` и выведем её содержимое командой `ls`:

```
[user@alt219 ifaces]$ cd enp1s1f0
[user@alt219 ifaces]$ ls
Options
```

В директории находится файл options - основной конфигурационный файл интерфейса enp1s1f0.

Для настройки интерфейса enp1s1f0 необходимо в конфигурационном файле /enp1s1f0/options изменить основные параметры интерфейса. Для редактирования конфигурационного файла нужны права суперпользователя root.

Вход в режим суперпользователя осуществляется командой `su -`, после чего требуется ввести пароль суперпользователя:

```
[user@alt219 enp1s1f0]$ su -
```

Отредактировать конфигурационный файл можно открыв его с помощью редактора nano:

```
[root@alt219]# nano /etc/net/ifaces/enp1s1f0/options
```

При стандартной настройке системы все сетевые интерфейсы управляются автоматически через Network Manager, необходимо включить конфигурирование интерфейсов вручную с помощью etcnet. Для этого в открывшемся окне редактора необходимо изменить основные параметры:

`BOOTPROTO=static` – использовать статически заданный IP-адрес;

`NM_CONTROLLED=no` – отключаем автоматическое управление интерфейсом через Network Manager;

`DISABLED=no` – включаем управление интерфейсом через etcnet;

`ONBOOT=yes` – включать интерфейс при загрузке системы;

`CONFIG_IPV6=yes` – режим работы с IPv6-адресами.

После настройки используется Ctrl+O чтобы сохранить изменения и Ctrl+X для выхода из файла конфигурации.

Далее необходимо назначить статический IPv6-адрес на интерфейс enp1s1f0. Для этого в /etc/net/ifaces/enp1s1f0 нужно создать файл с именем ipv6address и в него вписать необходимый IPv6-адрес.

Используя редактор nano, создается файл с именем ipv6address, в котором прописывается сетевой адрес:

```
[root@alt219 enp1s1f0]# nano ipv6address
2001:fb6:31c0:11::2/64 – прописывается статический IPv6-адрес;
```

Сохранение настроек Ctrl+O и выход из редактирования Ctrl+X.

Так как R2 является шлюзом по умолчанию для маршрутизатора R1, то на R1 необходимо прописать маршрут, по умолчанию отправляющий все пакеты на маршрутизатор R2. Для этого в /etc/net/ifaces/enp1s1f0 создается файл с именем ipv6route и в него прописывается маршрут по умолчанию:

```
[root@alt219 enp1s1f0]# nano ipv6route
default via 2001:fb6:31c0:11::1 – маршрут по умолчанию на R2;
```

Чтобы изменения применились, необходим перезапуск сетевой службы:

```
[root@alt219 enp1s1f0]#service network restart
```

Выход из суперпользователя осуществляется командой exit или Ctrl+D.

Проверяется настройка IPv6-адреса командой `ip address` или сокращенно `ip a` с ключом -6 (Показать только IPv6-адреса):

```
[user@alt219]$ ip -6 a
2: enp1s1f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qlen 1000
    inet6 2001:fb6:31c0:11::2/64 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::9aa7:b0ff:fe00:dea0/64 scope link
       valid_lft forever preferred_lft forever
```

Первая запись: `inet6 2001:fb6:31c0:11::2/64 scope global` – это global unicast адрес, сконфигурированный нами и использующийся для адресации в сетях IPv6.

Вторая запись: `inet6 fe80::9aa7:b0ff:fe00:dea0/64 scope link` – это link-local адрес, назначенный системой автоматически и предназначенный для взаимодействия между устройствами в пределах локального сегмента сети. Link-local адреса используют подсеть FE80::/64, а оставшаяся часть адреса

берется из MAC-адреса интерфейса. Данные адреса не маршрутизируются в сети Интернет и нужны для адресации на канале между двумя устройствами, а также для обеспечения работы необходимых механизмов IPv6.

Для проверки таблицы маршрутизации на R1. Используется команда `ip route` с ключом -6 (для отображения IPv6 маршрутов). Так как в работе используются IPv6-адреса подсетей, начинающиеся с 2001, то для уточнения поиска по таблице маршрутизации дополним запрос `ip -6 r` командой `grep 2001`:

```
[root@alt219 ~]# ip -6 r | grep 2001
2001:fb6:31c0:11::/64 dev enp1s1f0 proto kernel metric 256
default via 2001:fb6:31c0:1::1 dev enp1s1f0 metric 1024
```

В таблице маршрутизации присутствуют две записи. Первая запись – непосредственно подключенная напрямую к интерфейсу enp1s1f0 IPv6 сеть 2001:fb6:31c0:11::/64. Вторая запись – наш прописанный маршрут по умолчанию на маршрутизатор R2.

4.2. Настройка маршрутизатора R2

Для настройки сетевых интерфейсов на маршрутизаторе R2 редактируются их конфигурационные файлы в директории /etc/net/ifaces/.

Для интерфейса enp1s1f0 в /etc/net/ifaces/enp1s1f0 редактируется файл options:

```
[root@alt218]# nano /etc/net/ifaces/enp1s1f0/options
BOOTPROTO=static
NM_CONTROLLED=no
DISABLED=no
ONBOOT=yes
CONFIG_IPV6=yes
```

Настройка IPv6-адреса на enp1s1f0:

```
[root@alt218]# nano /etc/net/ifaces/enp1s1f0/ipv6address
2001:fb6:31c0:11::1/64
```

Для интерфейса enp1s1f1 в /etc/net/ifaces/enp1s1f1 редактируется файл options:

```
[root@alt218]# nano /etc/net/ifaces/enp1s1f1/options
```

```
BOOTPROTO=static
NM_CONTROLLED=no
DISABLED=no
ONBOOT=yes
CONFIG_IPV6=yes
```

Интерфейсу enp1s1f1 на R2 принадлежит адрес из второй подсети. Настройка IPv6-адреса на enp1s1f1:

```
[root@alt218]# nano
/etc/net/ifaces/enp1s1f1/ipv6address
2001:fb6:31c0:12::10/64
```

Так как R2 является транзитным маршрутизатором, у него должна быть возможность перенаправлять пакеты из одного интерфейса в другой. Для этого необходимо включить форвардинг пакетов на уровне ядра системы. В файле /etc/net/sysctl.conf нужно добавить строку:

```
net.ipv6.conf.all.forwarding=1
[root@alt218]# nano /etc/net/sysctl.conf
net.ipv6.conf.all.forwarding=1
```

После произведенных настроек интерфейсов и форвардинга пакетов, необходимо перезапустить сетевую службу и проверить настроенные IP-адреса:

```
[root@alt218]# service network restart
[root@alt218]# ip -6 address
2: enp1s1f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu
1500 qlen 1000
    inet6 2001:fb6:31c0:11::1/64 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::9aa7:b0ff:fe00:dd60/64 scope link
       valid_lft forever preferred_lft forever
3: enp1s1f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu
1500 qlen 1000
    inet6 2001:fb6:31c0:12::10/64 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::9aa7:b0ff:fe00:dd61/64 scope link
       valid_lft forever preferred_lft forever
```

После настроек необходима проверка сетевой связности между подсетями 1 и 2. Для этого с маршрутизатора R1 проверяется доступность подсети

2001:fb6:31c0:12::/64, находящейся за интерфейсом enp1s1f1 маршрутизатора R2.

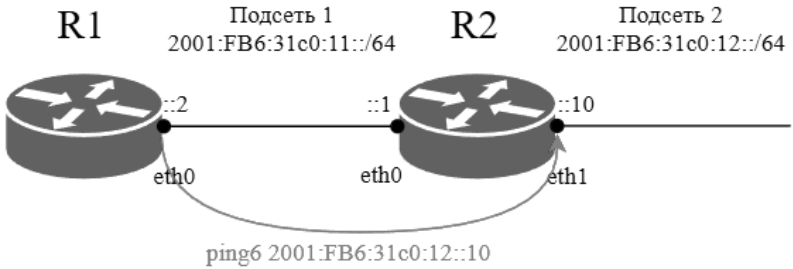


Рисунок 2. Проверка связности сети командой ping6

Чтобы проверить работу сети IPv6, используется утилита ping6, работающая по протоколу ICMPv6, специально разработанного для работы в сетях IPv6.

На R1 выполняется команда `ping6` и указывается адрес интерфейса enp1s1f1 маршрутизатора R2:

```
[root@alt219 ~]# ping6 2001:fb6:31c0:12::10 -c 3
PING 2001:fb6:31c0:12::10(2001:fb6:31c0:12::10) 56 data bytes
64 bytes from 2001:fb6:31c0:12::10: icmp_seq=1 ttl=64 time=0.184
ms
64 bytes from 2001:fb6:31c0:12::10: icmp_seq=2 ttl=64 time=0.132
ms
64 bytes from 2001:fb6:31c0:12::10: icmp_seq=3 ttl=64 time=0.102
ms
--- 2001:fb6:31c0:12::10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.102/0.139/0.184/0.035 ms
```

Ключ –c (count) задает количество посылаемых ICMP запросов. Интерфейс enp1s1f1 маршрутизатора R2 доступен, ICMP пакеты успешно возвращаются.

После того как взаимодействие между R1 и R2 установлено, устройства добавят друг друга в свою таблицу соответствия сетевых и физических адресов (ARP таблица для IPv4), установив соответствие между MAC-адресами и IPv6-адресами интерфейсов маршрутизаторов.

В сетях IPv6 протокол преобразования адресов ARP заменен на протокол обнаружения соседних устройств NDP (Neighbor Discovery Protocol).

Проверка таблицы NDP на маршрутизаторе R1 осуществляется командой `ip n` (neighbor) с ключом -6 (для протокола IPv6):

```
[root@alt219 ~]# ip -6 n
fe80::9aa7:b0ff:fe00:dd60 dev enp1s1f0 lladdr 98:a7:b0:00:dd:60
REACHABLE
2001:fb6:31c0:11::1   dev   enp1s1f0   lladdr   98:a7:b0:00:dd:60
REACHABLE
```

NDP таблица содержит две записи, устанавливающие соответствие между MAC-адресами и IPv6-адресами соседнего узла, т.е. маршрутизатора R2. Первая запись устанавливает соответствие для link-local адреса R2, а вторая для global unicast IPv6-адреса R2.

В NDP таблице маршрутизатора R2 также будут находиться IPv6-адреса и соответствующие им MAC-адреса маршрутизатора R1:

```
[root@alt218 ~]# ip -6 n
fe80::9aa7:b0ff:fe00:dea0 dev enp1s1f0 lladdr 98:a7:b0:00:de:a0
REACHABLE
2001:fb6:31c0:11::2   dev   enp1s1f0   lladdr   98:a7:b0:00:de:a0
REACHABLE
```

Отметка `REACHABLE` показывает, что запись о соответствии появилась недавно, и IPv6-адреса на R1 доступны.

Отметка `STALE` означает, что запись о доступности соседнего устройства устарела.

Отметка `UNREACHABLE` означает недоступность соседнего устройства по этому IP-адресу.

Поскольку R2 является транзитным маршрутизатором между подсетями 1 и 3, у него должен быть маршрут в третью подсеть. Взаимосвязь с подсетью 3 происходит через интерфейс enp1s1f1, поэтому маршрут добавляется в конфигурационном файле этого интерфейса:

```
[root@alt218]# nano /etc/net/ifaces/enp1s1f1/ipv6route
2001:fb6:31c0:13::/64 via 2001:fb6:31c0:12::11
```

Запись добавляет маршрут в подсеть 3 через интерфейс enp1s1f0 маршрутизатора R3.

На R2, после настройки маршрута, необходима перезагрузка сетевой службы и проверка таблицы маршрутизации. Так же дополняем запрос `ip -6 r` командой `grep 2001`:

```
[root@alt218]# service network restart
[root@alt218]# ip -6 r | grep 2001
2001:fb6:31c0:11::/64 dev enp1s1f0 proto kernel metric
256
```

```
    2001:fb6:31c0:12::/64 dev enp1s1f1 proto kernel metric
256
    2001:fb6:31c0:13::/64 via 2001:fb6:31c0:12::11 dev
enp1s1f1 metric 1024
```

В таблице маршрутизации первые две записи означают подключенные напрямую IPv6 сети к интерфейсам enp1s1f0 и enp1s1f1. Последняя запись – прописанный маршрут в подсеть 3 с интерфейса enp1s1f1 через транзитный маршрутизатор R3.

4.3. Настройка маршрутизатора R3

Для настройки сетевых интерфейсов на маршрутизаторе R3 редактируются их конфигурационные файлы в директории /etc/net/ifaces/.

Для интерфейса enp1s1f0 в /etc/net/ifaces/enp1s1f1 редактируется файл options:

```
    [root@alt217]# nano /etc/net/ifaces/enp1s1f1/options
    BOOTPROTO=static
    NM_CONTROLLED=no
    DISABLED=no
    ONBOOT=yes
    CONFIG_IPV6=yes
```

Настройка IPv6-адреса на enp1s1f1:

```
    [root@alt217]# nano
/etc/net/ifaces/enp1s1f1/ipv6address
    2001:fb6:31c0:12::11/64
```

Для интерфейса enp1s1f0 в /etc/net/ifaces/enp1s1f0 редактируется файл options:

```
    [root@alt217]# nano /etc/net/ifaces/enp1s1f0/options
    BOOTPROTO=static
    NM_CONTROLLED=no
    DISABLED=no
    ONBOOT=yes
    CONFIG_IPV6=yes
```

Интерфейсу enp1s1f0 на R3 принадлежит адрес из третьей подсети. Настройка IPv6-адреса на enp1s1f0:

```
    [root@alt217]# nano
/etc/net/ifaces/enp1s1f0/ipv6address
    2001:fb6:31c0:13::1/64
```

Поскольку R3 является транзитным маршрутизатором, у него должна быть возможность перенаправлять пакеты из одного интерфейса в другой. Для этого, как и на R2, включается форвардинг пакетов на уровне ядра системы. В файле /etc/net/sysctl.conf нужно добавить строку:

```
    net.ipv6.conf.all.forwarding=1
    [root@alt217]# nano /etc/net/sysctl.conf
    net.ipv6.conf.all.forwarding=1
```

Аналогично транзитному маршрутизатору R2, у маршрутизатора R3 должен быть маршрут в первую подсеть. Взаимосвязь с подсетью 1 происходит через интерфейс enp1s1f1, поэтому маршрут добавляется в конфигурационном файле этого интерфейса:

```
    [root@alt217]# nano /etc/net/ifaces/enp1s1f1/ipv6route
    2001:fb6:31c0:11::/64 via 2001:fb6:31c0:12::10
```

Запись добавляет маршрут в подсеть 1 через маршрутизатор R2.

После произведенных настроек интерфейсов и форвардинга пакетов, необходим перезапуск сетевой службы и проверка настроенных IPv6-адресов и маршрутов:

```
    [root@alt217]# service network restart
    [root@alt217]# ip -6 address
    2:  enp1s1f0:  <BROADCAST,MULTICAST,UP,LOWER_UP>  mtu
1500 qlen 1000
        inet6 2001:fb6:31c0:13::1/64 scope global
           valid_lft forever preferred_lft forever
        inet6 fe80::9aa7:b0ff:fe00:de20/64 scope link
           valid_lft forever preferred_lft forever
    3:  enp1s1f1:  <BROADCAST,MULTICAST,UP,LOWER_UP>  mtu
1500 qlen 1000
        inet6 2001:fb6:31c0:12::11/64 scope global
           valid_lft forever preferred_lft forever
        inet6 fe80::9aa7:b0ff:fe00:de21/64 scope link
           valid_lft forever preferred_lft forever
```

Таблица маршрутизации на R3. В таблице маршрутизации информация о подключенных напрямую подсетях, а также о том, что первая подсеть 2001:fb6:31c0:11::/64 доступна через маршрутизатор R2:

```
[root@alt217 ~]# ip -6 r | grep 2001
2001:fb6:31c0:11::/64 via 2001:fb6:31c0:12::10 dev enp1s1f1 metric 1024
2001:fb6:31c0:12::/64 dev enp1s1f1 proto kernel metric 256
2001:fb6:31c0:13::/64 dev enp1s1f0 proto kernel metric 256
```

После настроек необходима проверка сетевой связности между подсетями 1 и 3 командой `ping6`. Для этого с маршрутизатора R1 проверяется доступность подсети 2001:fb6:31c0:13::/64, находящейся за интерфейсом enp1s1f1 маршрутизатора R3:

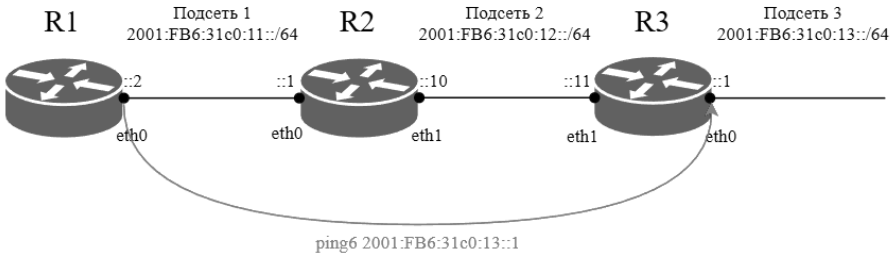


Рисунок 3. Проверка связности сети командой ping6

```
[root@alt219 ~]# ping6 2001:fb6:31c0:13::1 -c 3
PING 2001:fb6:31c0:13::1(2001:fb6:31c0:13::1) 56 data bytes
64 bytes from 2001:fb6:31c0:13::1: icmp_seq=1 ttl=63 time=0.145 ms
64 bytes from 2001:fb6:31c0:13::1: icmp_seq=2 ttl=63 time=0.132 ms
64 bytes from 2001:fb6:31c0:13::1: icmp_seq=3 ttl=63 time=0.130 ms
--- 2001:fb6:31c0:13::1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.130/0.135/0.145/0.015 ms
```

ICMP пакеты успешно возвращаются, интерфейс enp1s1f0 маршрутизатора R3 доступен.

4.4. Настройка маршрутизатора R4

Для настройки сетевых интерфейсов на маршрутизаторе R4 редактируются их конфигурационные файлы в директории /etc/net/ifaces/.

Для интерфейса enp1s1f0 в /etc/net/ifaces/enp1s1f0 редактируется файл options:

```
[root@alt216]# nano /etc/net/ifaces/enp1s1f0/options
BOOTPROTO=static
NM_CONTROLLED=no
DISABLED=no
ONBOOT=yes
CONFIG_IPV6=yes
```

Настройка IPv6-адреса на enp1s1f0:

```
[root@alt216]# nano
/etc/net/ifaces/enp1s1f0/ipv6address
2001:fb6:31c0:13::2/64
```

Так как R3 является шлюзом по умолчанию для маршрутизатора R4, то на R4 необходимо прописать маршрут, по умолчанию отправляющий все пакеты на маршрутизатор R3. Для этого в /etc/net/ifaces/enp1s1f0 создается файл с именем ipv6route и в него прописывается маршрут по умолчанию:

```
[root@alt216 enp1s1f0]# nano ipv6route
default via 2001:fb6:31c0:13::1 –
```

маршрут по умолчанию на R3;

После произведенных настроек необходим перезапуск сетевой службы и проверка настроенных IPv6-адресов и маршрутов:

```
[root@alt216]# service network restart
[root@alt216]# ip -6 a
2: enp1s1f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu
1500 qlen 1000
       inet6 2001:fb6:31c0:13::2/64 scope global
          valid_lft forever preferred_lft forever
       inet6 fe80::9aa7:b0ff:fe00:df20/64 scope link
          valid_lft forever preferred_lft forever
```

Проверка маршрута по умолчанию на R4 с помощью команды `ip -6 route`:

```
[root@alt216 ~]# ip -6 r | grep 2001
2001:fb6:31c0:13::/64 dev enp1s1f0 proto kernel metric 256
default via 2001:fb6:31c0:3::1 dev enp1s1f0 metric 1024
```

В таблице маршрутизации в дополнении к непосредственно подключенной напрямую сети, появился маршрут по умолчанию на R3, прописанный последней строкой.

4.5. Проверка работы построенной сети

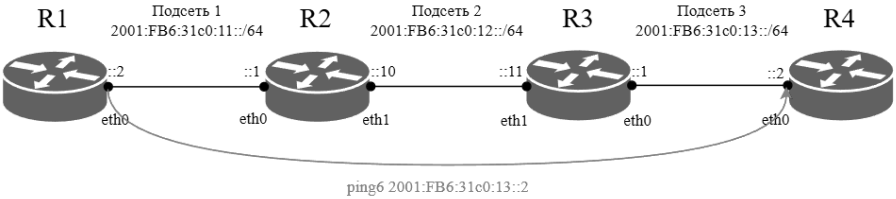


Рисунок 4. Проверка работы сети командой ping6

Проверка работы настроенной сети осуществляется командой `ping6` с маршрутизатора R1 на маршрутизатор R4:

```
[root@alt219 ~]# ping6 2001:fb6:31c0:13::2 -c 5
PING 2001:fb6:31c0:13::2(2001:fb6:31c0:13::2) 56 data bytes
64 bytes from 2001:fb6:31c0:13::2: icmp_seq=1 ttl=62 time=0.180 ms
64 bytes from 2001:fb6:31c0:13::2: icmp_seq=2 ttl=62 time=0.190 ms
64 bytes from 2001:fb6:31c0:13::2: icmp_seq=3 ttl=62 time=0.162 ms
64 bytes from 2001:fb6:31c0:13::2: icmp_seq=4 ttl=62 time=0.174 ms
64 bytes from 2001:fb6:31c0:13::2: icmp_seq=5 ttl=62 time=0.216 ms
--- 2001:fb6:31c0:13::2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3996ms
```

```
    rtt min/avg/max/mdev = 0.162/0.184/0.216/0.021 ms
```

Отправленные ICMP пакеты успешно возвращаются, маршрутизатор R4 доступен, настроенная сеть функционирует верно.

Проверить каким путем проходят пакеты между подсетями можно с помощью утилиты traceroute6, предназначенной для работы с IPv6:

```
    [root@alt219 ~]# traceroute6  2001:fb6:31c0:13::2

    traceroute to 2001:fb6:31c0:13::2
(2001:fb6:31c0:13::2), 30 hops max, 80 byte packets
     1  gateway (2001:fb6:31c0:11::1)  0.148 ms  0.088 ms
0.055 ms
     2  2001:fb6:31c0:12::11           0.196 ms  0.092 ms
0.099 ms
     3  2001:fb6:31c0:13::2            0.187 ms  0.158 ms
0.135 ms
```

Маршрутизатор R1, не зная маршрута до R4, отправляет свои пакеты на шлюз по умолчанию – маршрутизатор R2. Транзитные маршрутизаторы R2 и R3 знают о всех подсетях, и используя свою маршрутную информацию, перенаправляют пакеты между подсетями. Получив пакеты от R1, R2 отправляет их на R3, а R3 на R4. Так как R4 также не знает о подсети 2001:fb6:31c0:11::/64, в которой находится R1, он отправляет пакеты на свой шлюз по умолчанию – маршрутизатор R3, а тот уже через R2 на R1.

4.6. Утилита mtr6 для диагностики сетевых соединений

Путь прохождения пакетов можно также отследить с помощью утилиты mtr6. Утилита mtr6 является аналогом traceroute6 и в реальном времени показывает потери и задержки пакетов на определенных участках сети:

```
    [root@alt219]# mtr6 2001:fb6:31c0:13::2
    My traceroute [v0.95]
    elbrus.localdomain (::)                  23:30:15 2025
Host                        Packets                 Pings
                      loss%  Snt   Last   Avg  Best  Wrst
StDev
1. 2001:fb6:31c0:11::1  0.0%   11    0.4    1.1   0.4  1.4
0.3
2. 2001:fb6:31c0:12::11 0.0%   10    2.2    2.3   2.2  2.4
0.1
```

```
3. 2001:fb6:31c0:13::2  0.0%   10    2.2    2.3   2.2  2.4
0.1
```

Loss% – процент потерь пакетов;
Snt – количество отправленных пакетов;
Last – время задержки последнего полученного пакета, мс;
Avg – среднее время задержки, мс;
Best – наименьшее зафиксированное время задержки, мс;
Wrst – наибольшее зафиксированное время задержки, мс;
StDev – среднеквадратичное отклонение времени задержки, мс.

Практикум № 18
Протокол динамической маршрутизации OSPFv3

1. ЦЕЛЬ РАБОТЫ

Изучение особенностей работы и настройки протокола поиска первого кратчайшего пути OSPFv3 (Open Shortest Path First) версии 3, предназначенного для работы в сетях IPv6, с помощью отечественных программно-аппаратных средств. Изучение пакета quagga, реализующего протоколы маршрутизации, анализ его функционала в рамках настройки IPv6 сети, работающей по протоколу динамической маршрутизации OSPFv3.

2. ЗАДАНИЕ НА РАБОТУ

В работе используется топология сети, представленная на рисунке 1. Маршрутизаторам R1, R2, R3 и R4 соответствуют рабочие станции под номерами PC1, PC2, PC3 и PC4, соответственно. Адресация в моделируемой сети осуществляется следующим образом:

- Подсеть 1: $2001{:}\,\mathrm{fb6}{:}\,31\mathrm{c0}{:}\,\mathrm{X}(N_{min})::/64$ или $2001{:}\,\mathrm{fb6}{:}\,31\mathrm{c0}{:}\,\mathrm{X1}::/64$
- Подсеть 2: $2001{:}\,\mathrm{fb6}{:}\,31\mathrm{c0}{:}\,X(N_{min}+1)::/64$ или $2001{:}\,\mathrm{fb6}{:}\,31\mathrm{c0}{:}\,\mathrm{X2}::/64$
- Подсеть 3: $2001{:}\,\mathrm{fb6}{:}\,31\mathrm{c0}{:}\,X(N_{min}+2)::/64$ или $2001{:}\,\mathrm{fb6}{:}\,31\mathrm{c0}{:}\,\mathrm{X3}::/64$
- Подсеть 4: $2001{:}\,\mathrm{fb6}{:}\,31\mathrm{c0}{:}\,X(N_{min}+3)::/64$ или $2001{:}\,\mathrm{fb6}{:}\,31\mathrm{c0}{:}\,\mathrm{X4}::/64$

где $N_{min} = 1$ – наименьший номер из четырех рабочих станций;

X = номер ряда с рабочими станциями, для примера возьмём равное 1.

Наименование интерфейсов указано для примера, на практике оно может отличаться. В нашем же случае eth0 – enp1s1f0, а eth1 – enp1s1f1.

Последние цифры IPv6-адресов интерфейсов соответствуют номеру машины.

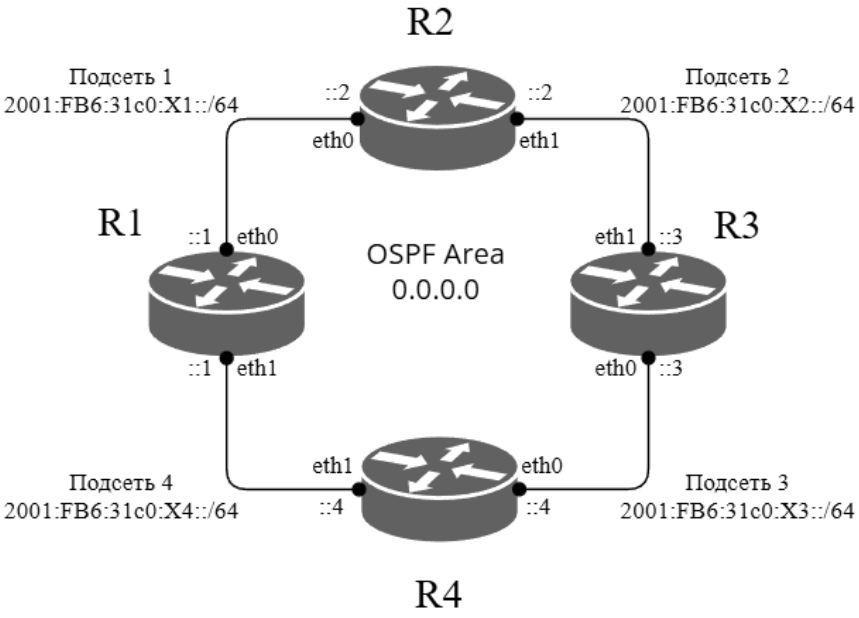


Рисунок 1. Схема сети.

Основные задачи работы:

- Подготовить машины к работе в сети IPv6 и установить необходимые пакеты;
- Изменить имя устройств и настроить сетевые интерфейсы, задействованные в работе протокола OSPFv3 с помощью демона zebra;
- С помощью демона ospf6d настроить работу протокола OSPFv3 на всех устройствах, настроить анонсирование необходимых подсетей и проверить работу протокола с помощью анализатора трафика Wireshark;
- Проверить отказоустойчивость построенной сети OSPFv3, смоделировав отказ канала связи между R1 и R2 путем отключения интерфейса enp1s1f0 на R1;
- Проверить работу протокола OSPFv3 при изменении полосы пропускания каналов связи между маршрутизаторами R1, R2 и R4.

3. ТЕРМИНЫ И ИСПОЛЬЗУЕМЫЕ КОМАНДЫ

Используемые команды

`apt-get install <имя пакета>` – команда пакетного менеджера Apt, используемого в ОС Альт. Инициирует установку указанных пакетов;

`chown` – команда смены владельца (change owner), когда также необходимо сменить права вложенные файлы и папки, используется ключ -R (рекурсивно);

`nano` – текстовый редактор, также может создавать текстовые файлы;

`systemctl <start/stop>` – команда управления системными сервисами

`netstat` – команда отображения сетевых соединений системы. Основные ключи: -t (TCP-соединения), -u (UDP-соединения), -l (слушающие сокеты), -p (номер процесса в системе), -n (не переводить IP-адреса в доменные имена);

`grep` – утилита для поиска информации;

`ip` – утилита для настройки и управления сетевыми интерфейсами. Для работы с протоколом IPv6 используется ключ -6:

`ip -6 address` – отображение информации о сетевых интерфейсах;

`ip -6 route` – отображение информации о маршрутах;

`telnet` – утилита для подключения к сетевым узлам по протоколу telnet;

`ping6` – утилита для проверки сетевых соединений в сетях IPv6;

`traceroute6` – утилита для трассировки маршрута в сетях IPv6.

Основные команды сетевых демонов quagga

`Show running config` – показать текущую конфигурацию устройства;

`Hostname <имя>` – смена имени устройства;

`Ipv6 forwarding` – режим работы маршрутизатора в сетях IPv6;

`Interface <имя>` – переход к конфигурированию интерфейса;

`Ipv6 address <IPv6-адрес/маска>` – Настройка IPv6-адреса интерфейса;

`Bandwidth <1-10000000 кбит/с>` – настройка скорости интерфейса;

`Router ospf6` – запуск протокола OSPFv3 на устройстве и переход к его конфигурированию;

`Router-id <x.x.x.x>` – настройка идентификатора маршрутизатора OSPF;

`Network <интерфейс> area <x.x.x.x>` – настройка анонсирования сети на определенном интерфейсе и присвоение ей конкретной зоны;

`Show ipv6 route ospf6` – показать таблицу маршрутизации OSPFv3;

`Show ipv6 ospf6 neighbor` – информация о соседях OSPF;

`Write memory` – сохранение конфигурации устройства.

4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

4.1. Установка пакетов и подготовка машин к работе

Для установки пакета quagga, как и любых дополнительных пакетов, необходимы права суперпользователя root, поэтому переходим в режим работы от root командой su – и вводим пароль суперпользователя:

```
[user@elbrus ~]$ su -
Password:
```

Используя пакетный менеджер apt, командой `apt-get install` устанавливаем необходимые пакеты. Кроме пакета quagga, для выполнения лабораторной работы понадобятся пакеты telnet и wireshark. Используем ключ –y, чтобы автоматически подтверждать установку пакетов:

```
[root@alt219 ~]# apt-get install -y quagga telnet wireshark
```

Конфигурационные файлы пакета quagga находятся в директории /etc/quagga/, а логи – в /var/log/quagga/.

Чтобы пакет мог корректно работать, необходимо сменить права на файлы конфигурации и логов, которые он использует. Командой `chown` делаем пользователя quagga и его одноименную группу владельцами файлов:

```
[root@alt219 ~]# chown -R quagga:quagga /etc/quagga/
[root@alt219 ~]# chown -R quagga:quagga /var/log/quagga/
```

После того, как пакеты установлены, необходимо на всех машинах включить форвардинг IPv6 пакетов на уровне ядра системы. Это необходимо для того, чтобы маршрутизаторы могли пересылать пакеты между своими интерфейсами.

Включение форвадинга производится редактированием конфигурационного файла системы sysctl.conf, находящегося в директории /etc/net/. В конфигурационном файле необходимо добавить строку net.ipv6.conf.all.ip_forwarding=1:

```
[root@alt219 ~]# nano /etc/net/sysctl.conf
net.ipv6.conf.all.ip_forwarding=1
```

Сохраняем изменения Ctrl+O и выходим с помощью Ctrl+X.

Также, для запуска демонов, работающих по протоколу IPv6, в системе необходимо включить поддержку IPv6, для этого в файле /etc/sysconfig/network необходимо добавить строку NETWORKING_IPV6=yes

```
[root@alt219 ~]# nano /etc/sysconfig/network
NETWORKING_IPV6=yes
```

Сохраняем изменения Ctrl+O и выходим с помощью Ctrl+X. После настроек перезапускаются сетевые сервисы системы командой `service network restart`:

```
[root@alt219 ~]# service network restart
```

4.2. Подключение к демонам и работа с конфигурационными файлами пакета quagga

Для настройки динамической маршрутизации OSPFv3 нужны управляющий демон zebra и демон протокола OSPFv3 – ospf6d:

Запускаем необходимые демоны:

```
[root@alt219 ~]# systemctl start zebra
[root@alt219 ~]# systemctl start ospf6d
```

Настраивать сетевые демоны можно как редактируя их конфигурационные файлы, находящиеся в директории /etc/quagga/, так и подключаясь к их консоли по telnet.

После запуска демонов на нашей машине появятся сокеты (IP-адрес + порт), прослушивающие входящие соединения, с помощью которых и осуществляется подключение к консоли демонов.

Проверим появление необходимых сокетов командой `netstat`, так как демоны quagga используют порты, начинающиеся от 2601, то для точного поиска дополняем запрос командой `grep 260:`

```
[root@alt219 ~]# netstat -tlnp | grep 260
Proto        Local  Address   Foreign  Address   State
PID/Program name
tcp         127.0.0.1:2601          0.0.0.0:*     LISTEN
4026/zebra
tcp                ::1:2606              :::*     LISTEN
4113/ospf6d
```

На локальном хосте IPv6-демон ospf6d по loopback IPv6-адресу слушает порт 2606, а zebra по loopback IPv4-адресу – порт 2601.

Используем telnet для подключения к консоли zebra или ospf6d. Указывается IP-адрес и порт необходимого демона:

`[root@alt219 ~]# telnet 127.0.0.1 2601` – подключение к zebra

`[root@alt219 ~]# telnet ::1 2606` – подключение к ospf6d

Подключимся к zebra. Вводим пароль zebra (по умолчанию) и попадаем в его консоль:

```
[root@alt219 ~]# telnet 127.0.0.1 2601
Trying 127.0.0.1...
Connected to 127.0.0.1.
User Access Verification
Password:
Router>
```

Для того чтобы начать настройку, сначала нужно войти в привилегированный режим командой `enable`. Привилегированный режим также защищен паролем (по умолчанию zebra):

```
Router> enable
Password:
Router#
```

Чтобы посмотреть текущую конфигурацию маршрутизатора, используется команда `show running-config`, или сокращенно `show run`:

```
Router# show run
```

Вывод команды покажет конфигурацию, записанную в файле /etc/quagga/zebra.conf, рассмотрим ее подробнее:

Имя хоста:

```
hostname Router
```

Зашифрованный пароль для подключения к нашему маршрутизатору:

```
password 8 LrjDz/a2KALVQ
```

Зашифрованный пароль для входа в привилегированный режим enable:

```
enable password 8 LrjDz/a2KALVQ
```

Команда, включающая шифрование паролей при просмотре конфигурации:

```
service password-encryption
```

Путь к файлу с логами:

```
log file /var/log/quagga/zebra.log
```

Список контроля доступа (ACL) с именем localhost, разрешающий доступ к консоли демона zebra только с локального хоста, и запрещающий все остальные соединения:

```
access-list localhost permit 127.0.0.1/32
access-list localhost deny any
```

Команды, указывающие применять список доступа localhost для входящих соединений по виртуальным терминальным линиям vty. Подключаясь по telnet, мы занимаем одну виртуальную терминальную линию vty:

```
line vty
access-class localhost
```

Внесение изменений в конфигурацию устройства производится в режиме глобального конфигурирования (config). Вход в режим осуществляется командой `configure terminal` или сокращённо `conf t`:

```
Router# configure terminal
Router(config)#
```

4.3. Настройка маршрутизатора R1

Для работы протокола OSPFv3 на маршрутизаторе R1 с помощью демона zebra настраиваются все сетевые интерфейсы, задействованные в работе протокола OSPFv3. Подключаемся к zebra и входим в режим глобального конфигурирования:

```
[root@alt219 ~]# telnet 127.0.0.1 2601
Trying 127.0.0.1...
Connected to 127.0.0.1.
User Access Verification
Password:
Router>
Router> enable
Router#
Router# configure terminal
Router(config)#
```

Изменим имя устройства:

```
Router(config)# hostname R1
```

Далее необходимо включить поддержку передачи пакетов в IPv6-сетях:

```
R1(config)# ipv6 forwarding
```

В процессе работы протокола OSPFv3 будут задействованы два интерфейса маршрутизатора: enp1s1f0 и enp1s1f1.

Настройка интерфейса enp1s1f0:

`R1(config)#interface enp1s1f0` – переход к конфигурированию интерфейса;

`R1(config-if)#ipv6 address 2001:fb6:31c0:11::1/64` – присваиваем адрес и маску подсети;

`R1(config-if)# bandwidth 100000` – задается скорость работы интерфейса в кбит/с (100 Мбит/с). Команда необходима для работы механизмов QoS (Quality of Service) и не меняет реальную физическую скорость интерфейса, а лишь заставляет маршрутизатор думать, что скорость является такой.

`R1(config-if)#exit` – выход из режима настройки интерфейса enp1s1f0.

```
R1(config)#
```

Настройка интерфейса enp1s1f1:

```
R1(config)#interface enp1s1f1
R1(config-if)#ipv6 address 2001:fb6:31c0:14::1/64
R1(config-if)# bandwidth 100000
R1(config-if)#exit
R1(config)#
```

Чтобы сохранить все произведенные настройки, нужно сохранить текущую конфигурацию zebra командой `write memory` или сокращенно `wr`:

```
R1(config)# write memory
Configuration saved to /etc/quagga/zebra.conf
R1(config)# exit
```

Вся конфигурация была перезаписана в /etc/quagga/zebra.conf.

Настройка демона zebra завершена, отключиться от демона можно с помощью комбинации Ctrl+D.

Далее на маршрутизаторе R1 необходимо настроить работу протокола OSPFv3 с помощью демона ospf6d. Подключаемся к ospf6d по telnet:

```
[root@alt219 ~]# telnet ::1 2606
Trying ::1...
Connected to ::1.
User Access Verification
Password:
router-OSPF6>
router-OSPF6> enable
router-OSPF6#
```

Командой `show running-config` можно посмотреть текущую конфигурацию, записанную в /etc/quagga/ospf6d.conf. Для внесения изменений в конфигурацию, переходим в режим глобального конфигурирования, используя команду `configure terminal`:

```
router-OSPF6# configure terminal
router-OSPF6(config)#
```

Изменим имя устройства:

```
router-OSPF6(config)# hostname R1-OSPF6
```

Теперь необходимо включить процесс OSPFv3 на маршрутизаторе:

`R1-OSPF6(config)# router ospf6` – включаем OSPFv3 на устройстве и переходим к его конфигурированию;

Для правильной работы у маршрутизатора OSPF должен быть задан идентификатор маршрутизатора Router ID (RID).

Изначально, при запуске процесса OSPF, маршрутизаторы в качестве Router ID используют самый большой IP-адрес физического интерфейса. Оставлять настройку по умолчанию не рекомендуется, так как в случае отключения физического интерфейса нарушится и процесс OSPF, идентификатор которого был привязан к IP-адресу физического интерфейса, это может привести к нарушению работы протокола OSPF на маршрутизаторе.

Router ID назначается следующим образом в порядке приоритета:

- с помощью команды `router-id`. Если команда задана, то маршрутизатор использует ее значение;
- если на каком-либо loopback-интерфейсе маршрутизатора сконфигурирован IP-адрес и этот интерфейс в состоянии up, то

маршрутизатор выбирает loopback-интерфейс с самым большим IP-адресом;

- маршрутизатор сам выбирает самый большой IP-адрес из имеющихся физических интерфейсов.

Таким образом, лучше назначать RID вручную командой `router-id` или использовать логические loopback-интерфейсы, т.к. они не полагаются на аппаратные средства, и переходят в рабочее состояние сразу при запуске устройства.

Зададим Router ID с помощью команды `router-id`:

```
R1-OSPF6(config-router)# router-id 1.1.1.1
```

Далее необходимо прописать все сети, которые будет анонсировать маршрутизатор по OSPF. При настройке сетей в OSPFv3 необходимо указывать к какой зоне `area` они принадлежат.

В лабораторной работе будет использоваться одна нулевая зона, называемая магистральной (backbone area). Если в сети существует несколько зон, все они должны быть подсоединены к магистральной зоне.

Пропишем все сети, подключенные к R1, которые необходимо анонсировать, и определим их в нулевую зону. Анонс сетей производится путем указания необходимых интерфейсов, находящихся в соответствующих сетях:

```
R1-OSPF6(config-router)# network enp1s1f0 area 0.0.0.0
R1-OSPF6(config-router)# network enp1s1f1 area 0.0.0.0
R1-OSPF6(config-router)# exit
```

Настройка демона ospf6d завершена. Сохраняем конфигурацию и отключаемся от ospf6d с помощью Ctrl+D.

```
R1-OSPF6(config)# write memory
Configuration saved to /etc/quagga/ospf6d.conf
R1-OSPF6(config)# exit
```

4.4. Проверка работы протокола OSPFv3

Проверим настройку IPv6-адресов на маршрутизаторе R1 командой ip adress с ключом -6 (для протокола IPv6):

```
[root@alt219 ~]# ip -6 address
2: enp1s1f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu
1500 qlen 1000
    inet6 2001:fb6:31c0:11::1/64 scope global
       valid_lft forever preferred_lft forever
```

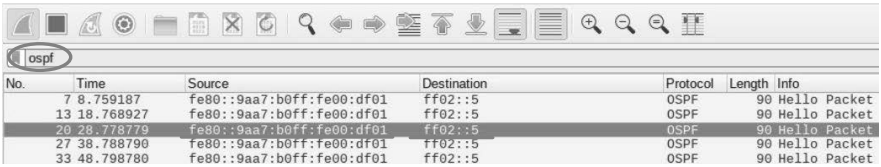
```
       inet6 fe80::9aa7:b0ff:fe00:df00/64 scope link
          valid_lft forever preferred_lft forever
   3: enp1s1f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu
1500 qlen 1000
       inet6 2001:fb6:31c0:14::1/64 scope global
          valid_lft forever preferred_lft forever
       inet6 fe80::9aa7:b0ff:fe00:df01/64 scope link
          valid_lft forever preferred_lft forever
```

Первая запись: `inet6 2001:fb6:31c0:11::1/64 scope global` – это global unicast адрес, сконфигурированный нами и использующийся для адресации в сетях IPv6.

Вторая запись: `inet6 fe80::9aa7:b0ff:fe00:df00/64 scope link` – это link-local адрес, назначенный системой автоматически и предназначенный для взаимодействия между устройствами в пределах локального сегмента сети. Link-local адреса используют подсеть FE80::/64, а оставшаяся часть адреса берется из MAC-адреса интерфейса. Данные адреса не маршрутизируются в сети Интернет и нужны для адресации на канале между двумя устройствами, а также для обеспечения работы необходимых механизмов IPv6.

Как только маршрутизатор настроен, он начинает поиск соседей, рассылая пакеты «Hello» на групповой IPv6-адрес FF02::5, предназначенный для всех маршрутизаторов, поддерживающих протокол OSPFv3. Для отправки пакетов Hello в качестве источника пакета, маршрутизаторы используют link-local адреса интерфейсов. Маршрутизаторы OSPFv3 получают пакеты Hello, посылаемые на этот IPv6-адрес и узнают из них о других соседях в сети.

Проверим работу протокола OSPFv3 на R1 с помощью анализатора трафика, для этого на интерфейсе enp1s1f1 запустим Wireshark и зададим фильтр по протоколу OSPF:



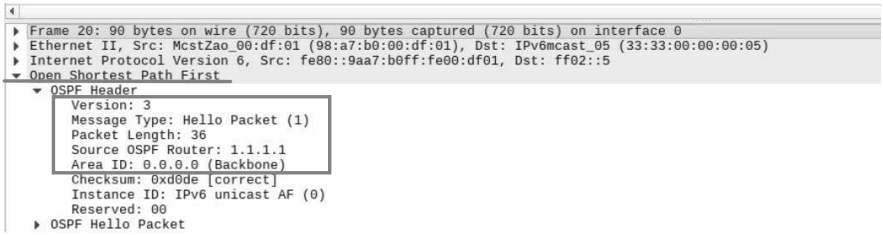


Рисунок 2. Анализ работы протокола OSPFv3 на R1

Анализатор трафика показывает, как маршрутизатор R1 с Анализатор показывает, как с link-local адреса интерфейса enp1s1f1, который имеет адрес `fe80::9aa7:b0ff:fe00:df01`, отправляет пакеты OSPFv3 Hello на групповой IPv6-адрес FF02::5. Заголовок пакета OSPFv3 Hello содержит основные параметры маршрутизатора, а именно: версию протокола OSPF (Version 3, версия OSPF для протокола IPv6), тип пакета (Hello Packet), Router ID маршрутизатора (Source OSPF Router) и идентификатор зоны (Area ID).

Hello пакеты используются для обнаружения соседей, а также выступают в роли keepalive-пакетов для проверки доступности, и отправляются по умолчанию каждые 10 секунд. Пакет несет в себе параметры, о которых маршрутизаторы OSPFv3 должны договориться перед тем, как становиться соседями.

4.5. Настройка маршрутизатора R2

Теперь производится настройка интерфейсов маршрутизатора R2. На R2 запускаем демон zebra, подключаемся к нему по telnet и входим в режим глобального конфигурирования:

```
[root@alt218 ~]# systemctl start zebra
[root@alt218 ~]# telnet 127.0.0.1 2601
Trying 127.0.0.1...
Connected to 127.0.0.1.
User Access Verification
Password:
Router>
Router enable
Router#
Router# configure terminal
Router(config)#
```

Изменим имя устройства:

```
Router(config)# hostname R2
```

Включение поддержки передачи пакетов в IPv6-сетях:

```
 R2(config)# ipv6 forwarding
```

Настройка интерфейса enp1s1f0:

```
R2(config)#interface enp1s1f0
R2(config-if)# ipv6 address 2001:fb6:31c0:11::2/64
R2(config-if)# bandwidth 100000
R2(config-if)# exit
R2(config)#
```

Настройка интерфейса enp1s1f1:

```
R2(config)#interface enp1s1f1
R2(config-if)#ipv6 address 2001:fb6:31c0:12::2/64
R2(config-if)# bandwidth 100000
R2(config-if)#exit
```

Настройка демона zebra завершена, необходимо сохранить конфигурацию. Отключиться от демона можно с помощью комбинации Ctrl+D.

```
R2(config)# write memory
Configuration saved to /etc/quagga/zebra.conf
R2(config)# exit
```

Далее на маршрутизаторе R2 необходимо настроить работу протокола OSPFv3 с помощью демона ospf6d. Запускаем демон, подключаемся к ospf6d по telnet и входим в режим глобального конфигурирования:

```
[root@alt218 ~]# systemctl start ospf6d
[root@alt218 ~]# telnet ::1 2606
Trying ::1...
Connected to ::1.
User Access Verification
Password:
router-OSPF6>
router-OSPF6> enable
router-OSPF6#
router-OSPF6# configure terminal
router-OSPF6(config)#
```

Изменим имя устройства:

```
router-OSPF6(config)# hostname R2-OSPF6
```

Теперь необходимо включить процесс OSPFv3, задать идентификатор Router ID и прописать сети, которые будет анонсировать наш маршрутизатор:

```
R2-OSPF6(config)# router ospf6
R2-OSPF6(config-router)# router-id 2.2.2.2
```

Пропишем все нужные для анонсирования сети, подключенные к маршрутизатору, путем добавления соответствующих интерфейсов и определим их в нулевую зону:

```
R2-OSPF6(config-router)# network enp1s1f0 area 0.0.0.0
R2-OSPF6(config-router)# network enp1s1f1 area 0.0.0.0
```

Проверим текущую конфигурацию демона ospf6d командой `show run`, если все настроено верно, то основная конфигурация ospf6d должна содержать следующие строки:

```
R2-OSPF6(config)# show run
!
router ospf6
 router-id 2.2.2.2
 interface enp1s1f0 area 0.0.0.0
 interface enp1s1f1 area 0.0.0.0
!
```

Настройка демона ospf6d завершена, необходимо сохранить конфигурацию:

```
R2-OSPF6(config)# write memory
Configuration saved to /etc/quagga/ospf6d.conf
R2-OSPF6(config)# exit
```

Отключиться от демона можно с помощью комбинации Ctrl+D.

4.6. Настройка маршрутизатора R3

Настройка интерфейсов маршрутизатора R3 производится с помощью демона zebra. На R3 запускаем демон zebra, подключаемся к нему по telnet и входим в режим глобального конфигурирования:

```
[root@alt217 ~]# systemctl start zebra
[root@alt217 ~]# telnet 127.0.0.1 2601
Trying 127.0.0.1...
```

```
Connected to 127.0.0.1.
User Access Verification
Password:
Router>
Router> enable
Router#
Router# configure terminal
Router(config)#
```

Изменим имя устройства:

```
Router(config)# hostname R3
```

Включение поддержки передачи пакетов в IPv6-сетях:

```
 R3(config)# ipv6 forwarding
```

Настройка интерфейса enp1s1f0:

```
R3(config)#interface enp1s1f0
R3(config-if)#ipv6 address 2001:fb6:31c0:13::3/64
R3(config-if)# bandwidth 100000
R3(config-if)#exit
```

Настройка интерфейса enp1s1f1:

```
R3(config)#interface enp1s1f1
R3(config-if)#ip address 2001:fb6:31c0:12::3/64
R3(config-if)# bandwidth 100000
R3(config-if)#exit
R3(config)#
```

Настройка демона zebra завершена, необходимо сохранить конфигурацию. Отключиться от демона можно с помощью комбинации Ctrl+D:

```
R3(config)# write memory
Configuration saved to /etc/quagga/zebra.conf
R3(config)# exit
```

Далее на маршрутизаторе R3 необходимо настроить работу протокола OSPFv3 с помощью демона ospf6d. Запускаем демон, подключаемся к ospf6d по telnet и входим в режим глобального конфигурирования:

```
[root@alt217 ~]# systemctl start ospf6d
[root@alt217 ~]# telnet ::1 2606
```

```
User Access Verification
Password:
router-OSPF6>
router-OSPF6> enable
router-OSPF6#
router-OSPF6# configure terminal
router-OSPF6(config)#
```

Изменим имя устройства:

```
router-OSPF6(config)# hostname R3-OSPF6
```

Теперь необходимо включить процесс OSPFv3 и прописать сети, которые будет анонсировать маршрутизатор R3:

```
R3-OSPF6(config)# router ospf6
```

Теперь необходимо включить процесс OSPFv3, задать идентификатор Router ID и прописать сети, которые будет анонсировать наш маршрутизатор:

```
R3-OSPF6(config)# router ospf6
R3-OSPF6(config-router)# router-id 3.3.3.3
```

Пропишем все нужные для анонсирования сети, подключенные к маршрутизатору, путем добавления соответствующих интерфейсов и определим их в нулевую зону:

```
R3-OSPF6(config-router)# network enp1s1f0 area 0.0.0.0
R3-OSPF6(config-router)# network enp1s1f1 area 0.0.0.0
```

Настройка демона ospf6d завершена, необходимо сохранить конфигурацию. Отключиться от демона можно с помощью комбинации Ctrl+D.

```
R3-OSPF6(config)# write memory
Configuration saved to /etc/quagga/ospf6d.conf
R3-OSPF6(config)# exit
```

4.7. Настройка маршрутизатора R4

Настройка интерфейсов маршрутизатора R4 производится с помощью демона zebra. На R4 запускаем демон zebra, подключаемся к нему по telnet и входим в режим глобального конфигурирования:

```
[root@alt216 ~]# systemctl start zebra
[root@alt216 ~]# telnet 127.0.0.1 2601
```

```
Trying 127.0.0.1...
Connected to 127.0.0.1.
User Access Verification
Password:
Router>
Router> enable
Router#
Router# configure terminal
Router(config)#
```

Изменим имя устройства:

```
Router(config)# hostname R4
```

Включение поддержки передачи пакетов в IPv6-сетях:

```
R4(config)# ipv6 forwarding
```

Настройка интерфейса enp1s1f0:

```
R4(config)#interface enp1s1f0
R4(config-if)#ipv6 address 2001:fb6:31c0:13::4/64
R4(config-if)# bandwidth 100000
R4(config-if)#exit
```

Настройка интерфейса enp1s1f1:

```
R4(config)#interface enp1s1f1
R4(config-if)#ip address 2001:fb6:31c0:14::4/64
R4(config-if)# bandwidth 100000
R4(config-if)#exit
R4(config)#
```

Настройка демона zebra завершена, необходимо сохранить конфигурацию. Отключиться от демона можно с помощью комбинации Ctrl+D:

```
R4(config)# write memory
Configuration saved to /etc/quagga/zebra.conf
R4(config)# exit
```

Далее на маршрутизаторе R4 необходимо настроить работу протокола OSPFv3 с помощью демона ospf6d. Запускаем демон, подключаемся к ospf6d по telnet и входим в режим глобального конфигурирования:

```
[root@alt216 ~]# systemctl start ospf6d
```

```
[root@alt216 ~]# telnet ::1 2606
User Access Verification
Password:
router-OSPF6>
router-OSPF6> enable
router-OSPF6#
router-OSPF6# configure terminal
router-OSPF6(config)#
```

Изменим имя устройства:

```
router-OSPF6(config)# hostname R4-OSPF6
```

Теперь необходимо включить процесс OSPFv3 и прописать сети, которые будет анонсировать маршрутизатор R3:

```
R4-OSPF6(config)# router ospf6
```

Теперь необходимо включить процесс OSPFv3, задать идентификатор Router ID и прописать сети, которые будет анонсировать наш маршрутизатор:

```
R4-OSPF6(config)# router ospf6
R4-OSPF6(config-router)# router-id 4.4.4.4
```

Пропишем все нужные для анонсирования сети, подключенные к маршрутизатору, путем добавления соответствующих интерфейсов и определим их в нулевую зону:

```
R4-OSPF6(config-router)# network enp1s1f0 area 0.0.0.0
R4-OSPF6(config-router)# network enp1s1f1 area 0.0.0.0
```

Настройка демона ospf6d завершена, необходимо сохранить конфигурацию. Отключиться от демона можно с помощью комбинации Ctrl+D.

```
R4-OSPF6(config)# write memory
Configuration saved to /etc/quagga/ospf6d.conf
R4-OSPF6(config)# exit
```

4.8. Анализ работы сети, работающей по протоколу OSPF

Когда все маршрутизаторы сконфигурированы, между ними должны установиться соседские отношения с помощью пакетов OSPF Hello. Проверим наличие соседских отношений на маршрутизаторе R1. Для этого подключаемся к ospf6d и из привилегированного режима (enable) вводим команду show ipv6 ospf6 neighbor:

```
    [root@alt219 ~]# telnet 127.0.0.1 2606
    R1-OSPF6>
    R1-OSPF6> enable
    R1-OSPF6# show ipv6 ospf6 neighbor
    Neighbor ID   Pri   DeadTime   State/IfState   Duration
I/F[State]
    2.2.2.2         1     00:00:39    Full/BDR       00:00:50
enp1s1f0[DR]
    4.4.4.4         1     00:00:38    Full/BDR       00:00:30
enp1s1f1[DR]
```

Вывод команды показывает, что R1 установил соседские отношения с R2 и R4. Таблица содержит следующие обозначения:

- Neighbor ID: Идентификатор (Router ID) маршрутизатора, с которым установлено соседство;
- Pri: Поле приоритета, указывает приоритет соседнего маршрутизатора. По умолчанию приоритеты установлены на 1;
- Dead Time: Оставшееся время, в течении которого маршрутизатор будет ждать пакет OSPF Hello от соседа, прежде чем объявить его недоступным (по умолчанию 40 секунд, т.е. 4 интервала Hello);
- State: Поле State показывает функциональное состояние соседнего маршрутизатора. Состояние Full означает полную синхронизацию с соседом;
- Duration: Поле Duration указывает время, прошедшее с установления соседских отношений с другими маршрутизаторами;
- I/F[State]: Поле Интерфейса указывает локальный интерфейс, на котором было установлено OSPF соседство.

Проверим работу протокола OSPFv3 проверкой доступности сети 2001:fb6:31c0:12::0/64 с маршрутизатора R1. Так как данная сеть не подключена напрямую к маршрутизатору R1, то с помощью протокола OSPFv3 он должен получить к ней маршрут от своих соседей.

Чтобы проверить работу сети IPv6, используется утилита ping6, работающая по протоколу ICMPv6, который предназначен для работы в сетях IPv6.

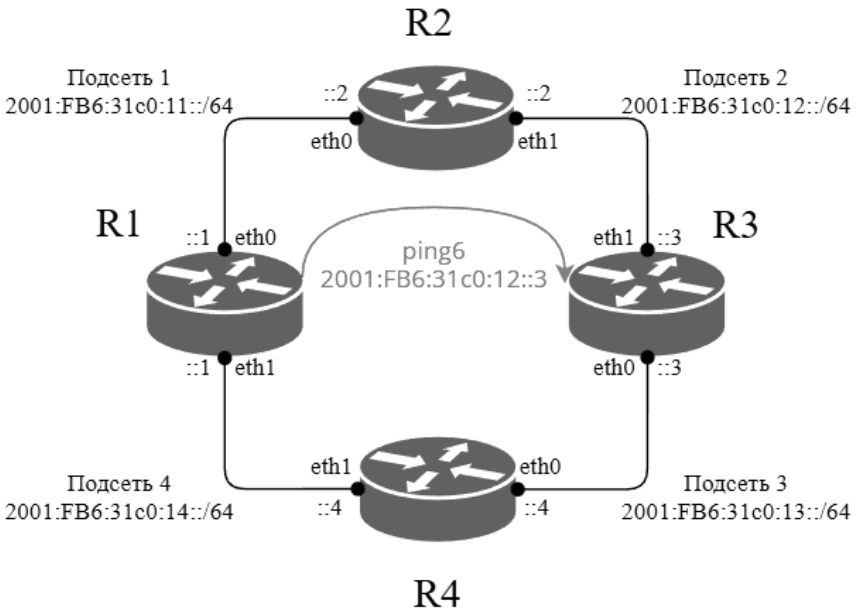


Рисунок 3. Проверка связности настроенной сети

На R1 выполняется команда `ping6` и указывается адрес интерфейса enp1s1f1 маршрутизатора R3:

```
[root@alt219 ~]# ping6 2001:fb6:31c0:12::3 -c 3
PING 2001:fb6:31c0:12::3(2001:fb6:31c0:12::3) 56 data bytes
64 bytes from 2001:fb6:31c0:12::3: icmp_seq=1 ttl=63 time=0.495 ms
64 bytes from 2001:fb6:31c0:12::3: icmp_seq=2 ttl=63 time=0.171 ms
64 bytes from 2001:fb6:31c0:12::3: icmp_seq=3 ttl=63 time=0.135 ms
--- 2001:fb6:31c0:12::3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.135/0.267/0.495/0.161 ms
```

Ключ –c (count) задает количество посылаемых ICMP запросов. Интерфейс enp1s1f1 маршрутизатора R3 доступен, ICMP пакеты успешно возвращаются, значит R1 получил необходимый маршрут и сеть работает корректно.

Проверим таблицу маршрутизации OSPFv3 на R1. Подключаемся по telnet к zebra, и в привилегированном режиме (enable) используем команду show ipv6 route ospf6:

```
[root@alt219 ~]# telnet 127.0.0.1 2601
R1>
R1> enable
R1# show ipv6 route ospf6
Codes: K - kernel route, C - connected, S - static, R - RIPng,O - OSPFv6, I - IS-IS, B - BGP, A - Babel, > - selected route, * - FIB route
O       2001:fb6:31c0:11::/64   [110/1]   is   directly connected, enp1s1f0, 00:01:48
O>*       2001:fb6:31c0:12::/64       [110/2]       via fe80::9aa7:b0ff:fe00:ddc0, enp1s1f0, 00:00:59
O>*       2001:fb6:31c0:13::/64       [110/2]       via fe80::9aa7:b0ff:fe00:dd21, enp1s1f1, 00:00:50
O       2001:fb6:31c0:14::/64   [110/1]   is   directly connected, enp1s1f1, 00:01:28
```

В таблице маршрутизации указаны маршруты, полученные R1 по OSPFv3 от своих соседей R2 и R4. Сеть 2001:fb6:31c0:12::/64 доступна через fe80::9aa7:b0ff:fe00:ddc0 (link-local адрес маршрутизатора R2), а сеть 2001:fb6:31c0:13::/64 доступна через fe80::9aa7:b0ff:fe00:dd21 (link-local адрес маршрутизатора R4).

Согласно таблице маршрутизации, сеть 2001:fb6:31c0:12::/64 доступна через R2. Проверим это, выполнив с R1 трассировку маршрута утилитой traceroute6 до адреса 2001:fb6:31c0:12::3, находящегося в подсети 2001:fb6:31c0:12::/64:

```
[root@alt219 ~]# traceroute6 2001:fb6:31c0:12::3
traceroute to 2001:fb6:31c0:12::3 (2001:fb6:31c0:12::3), 30 hops max, 80 byte packets
 1  2001:fb6:31c0:11::2  0.123 ms  0.077 ms  0.052 ms
 2  2001:fb6:31c0:12::3  0.140 ms  0.098 ms  0.088 ms
```

Как показывает трассировка, пакеты сначала попадают на адрес 2001:fb6:31c0:11::2 маршрутизатора R2, а потом достигают адреса назначения 2001:fb6:31c0:12::3 на маршрутизаторе R3.

Проверим как протокол OSPFv3 перестраивает маршруты до подсетей при изменении сетевой топологии, путем отключения интерфейса enp1s1f0 на маршрутизаторе R1. После отключения порта, протокол OSPFv3 перестроит маршрут к подсети 2001:fb6:31c0:12::/64 через маршрутизатор R4.

Как только происходит отключение интерфейса, алгоритм OSPF создает анонсы LSA (Link State Advertisement), в которых содержатся изменения в базе данных LSDB (Link State Data Base) маршрутизатора, а именно – отключенный интерфейс enp1s1f0, и рассылает их своим соседям, а они своим соседям, пока у всех маршрутизаторов не появится одинаковая копия базы LSDB, содержащая информацию об измененной топологии сети.

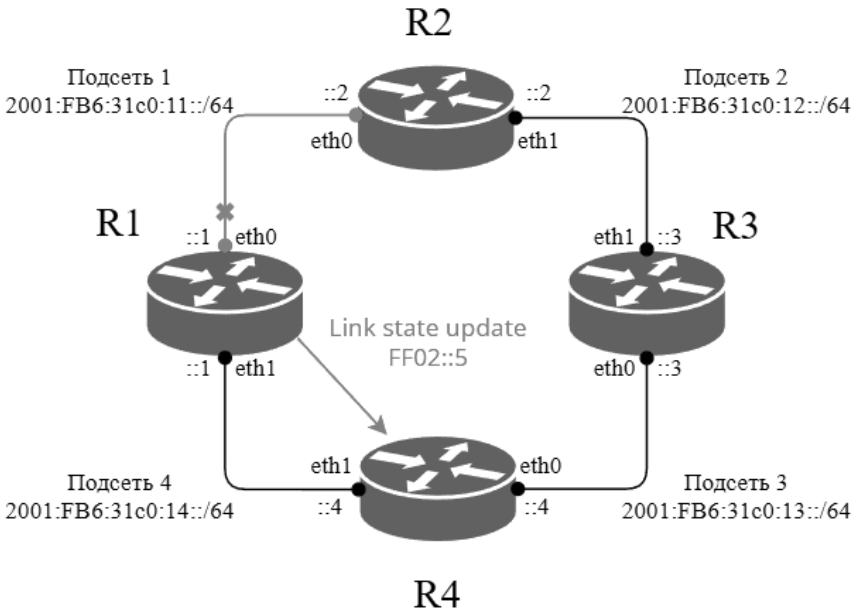


Рисунок 4. Схема сети при отключении интерфейса enp1s1f0

Запустим анализатор трафика Wireshark на интерфейсе enp1s1f1 маршрутизатора R1, и зададим фильтр по протоколу OSPF. После этого отключаем интерфейс enp1s1f0 и с помощью Wireshark смотрим анонсируемую информацию с интерфейса enp1s1f1 маршрутизатора R1:

```
[root@alt219 ~]# ip link set dev enp1s1f0 down
```

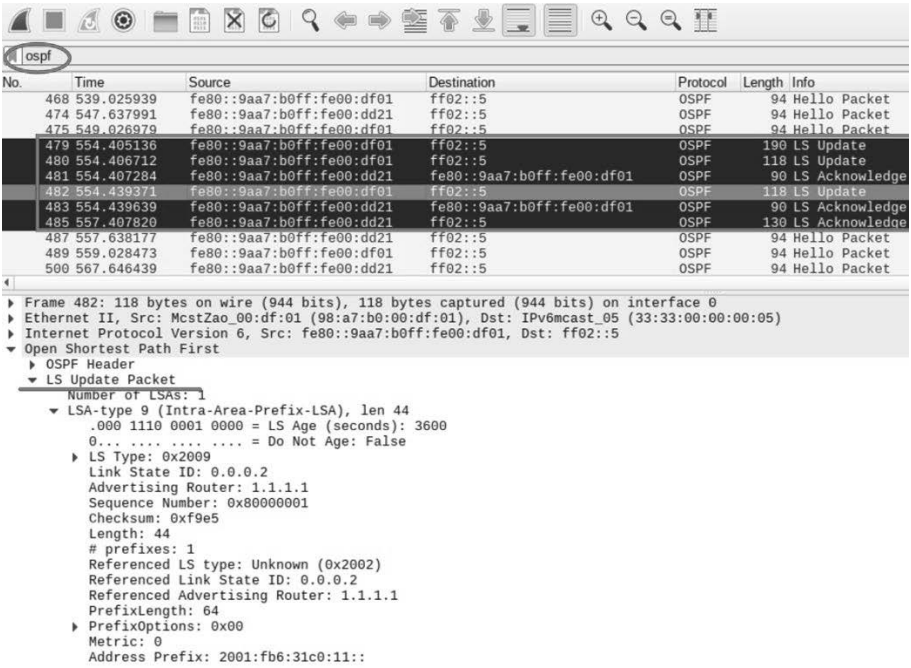


Рисунок 5. Анализ работы протокола OSPFv3 при отключении интерфейса

В частности, сообщения LSA, передающие информацию об изменении, называются обновлениями состояния канала LSU (Link State Update). Как показывает анализатор пакетов, при отключении интерфейса маршрутизатор R1 сразу начал отправку своим соседям на групповой адрес Update сообщений LSU, содержащих изменения базы данных LSDB. Маршрутизаторы R1 и R4, используя link-local адреса интерфейсов, обмениваются пакетами LS Update и подтверждениями LS Acknowledgment, после чего происходит перестроение маршрутов в сети.

После перестроения маршрутов, используем утилиту ping6 с маршрутизатора R1, проверим доступность адреса 2001:fb6:31c0:11::2, который находится в подсети 2001:fb6:31c0:11::/64:

```
[root@alt219 ~]# ping6 2001:fb6:31c0:11::2 -c 5
PING 2001:fb6:31c0:11::2(2001:fb6:31c0:11::2) 56 data bytes
64 bytes from 2001:fb6:31c0:11::2: icmp_seq=1 ttl=62 time=0.178 ms
```

```
    64 bytes from 2001:fb6:31c0:11::2: icmp_seq=2 ttl=62
time=0.221 ms
    64 bytes from 2001:fb6:31c0:11::2: icmp_seq=3 ttl=62
time=0.202 ms
    64 bytes from 2001:fb6:31c0:11::2: icmp_seq=4 ttl=62
time=0.197 ms
    64 bytes from 2001:fb6:31c0:11::2: icmp_seq=5 ttl=62
time=0.200 ms

    --- 2001:fb6:31c0:11::2 ping statistics ---
    5 packets transmitted, 5 received, 0% packet loss, time
3998ms
    rtt min/avg/max/mdev = 0.178/0.199/0.221/0.020 ms
```

Интерфейс enp1s1f0 маршрутизатора R2 доступен, ICMP пакеты успешно возвращаются, протокол OSPFv3 перестроил маршрут к сети 2001:fb6:31c0:11::0/64 через маршрутизаторы R4 и R3.

После перестроения маршрута, используя утилиту traceroute6 с маршрутизатора R1 проверим путь трафика до адреса 2001:fb6:31c0:12::2:

```
    [root@alt219 ~]# traceroute6 2001:fb6:31c0:11::2
    traceroute to 2001:fb6:31c0:11::2 (2001:fb6:31c0:11::2), 30
hops max, 80 byte packets
     1  2001:fb6:31c0:14::4          0.193 ms  0.122 ms  0.092
ms
     2  2001:fb6:31c0:13::3          0.193 ms  0.168 ms  0.149
ms
     3  2001:fb6:31c0:11::2          0.263 ms  0.245 ms  0.222
ms
```

Трассировка показывает, что пакеты сначала попадают на узел 2001:fb6:31c0:14::4 (маршрутизатор R4), а потом через маршрутизатор R3 (2001:fb6:31c0:13::3) попадают в нужную подсеть на заданный IPv6-адрес 2001:fb6:31c0:11::2.

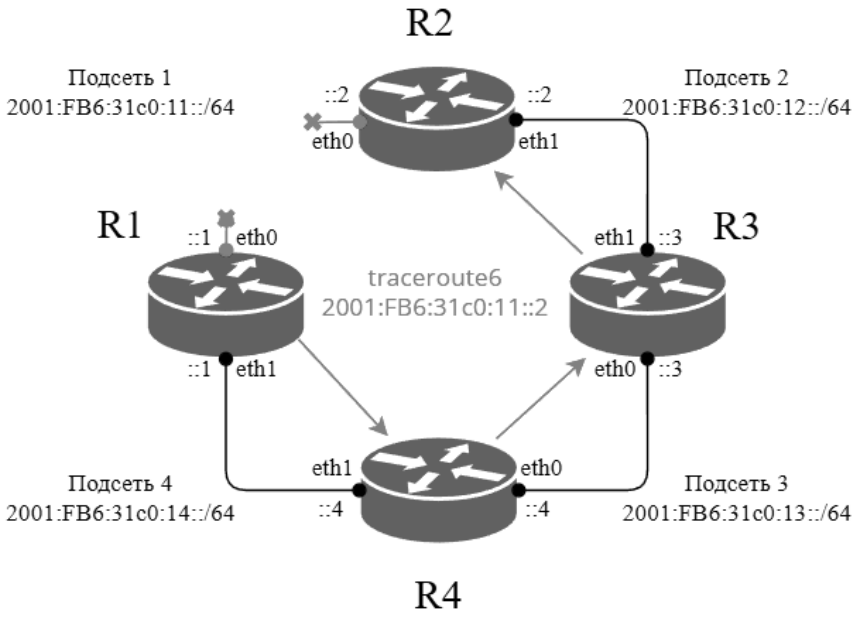


Рисунок 6. Схема прохождения трафика в сети

Когда маршруты перестроены, проверим таблицу маршрутизации OSPFv3 на R1. Для этого подключимся к zebra и в привилегированном режиме (enable) используем команду `show ipv6 route ospf6`:

```
R1# show ipv6 route ospf6
Codes: K - kernel route, C - connected, S - static, R - RIPng, O - OSPFv6, I - IS-IS, B - BGP, A - Babel, > - selected route, * - FIB route
O>*       2001:fb6:31c0:11::/64       [110/4]       via fe80::9aa7:b0ff:fe00:dd21, enp1s1f1, 00:03:58
O>*       2001:fb6:31c0:12::/64       [110/3]       via fe80::9aa7:b0ff:fe00:dd21, enp1s1f1, 00:04:33
O>*       2001:fb6:31c0:13::/64       [110/2]       via fe80::9aa7:b0ff:fe00:dd21, enp1s1f1, 00:08:30
O        2001:fb6:31c0:14::/64   [110/1]   is   directly connected, enp1s1f1, 00:09:08
```

Теперь в таблице маршрутизации указано, что все существующие сети доступны через fe80::9aa7:b0ff:fe00:df21 (link-local адрес маршрутизатора R4) с интерфейса enp1s1f1.

Протокол OSPFv3 также может выбирать наилучший маршрут до подсетей исходя их полосы пропускания канала на пути прохождения трафика. Если снизить скорость линии между маршрутизаторами, то протокол OSPFv3 также сначала начнет рассылку своим соседям Update сообщений, содержащих изменения в топологии сети, а после перестроит маршрут до сети назначения по более скоростным каналам.

На маршрутизаторе R1 вернем в активное состояние интерфейс enp1s1f0 и для правильной работы перезапустим демон zebra:

```
[root@alt219 ~]# ip link set dev enp1s1f0 up
[root@alt219 ~]# systemctl restart zebra
```

Проверим таблицу маршрутизации OSPFv3 на R1. Подключаемся по telnet к zebra, и в привилегированном режиме (enable) используем команду `show ipv6 route ospf6`. Подсеть 2001:fb6:31c0:11::/64 появилась в статусе подключенной напрямую (directly connected), а подсеть 2001:fb6:31c0:12::/64 снова доступна через R2:

```
[root@alt219 ~]# telnet 127.0.0.1 2601
R1>
R1> enable
R1# show ipv6 route ospf6
Codes: K - kernel route, C - connected, S - static, R - RIPng,O - OSPFv6, I - IS-IS, B - BGP, A - Babel, > - selected route, * - FIB route
O       2001:fb6:31c0:11::/64   [110/1]   is   directly connected, enp1s1f0, 00:01:35
O>*       2001:fb6:31c0:12::/64       [110/2]       via fe80::9aa7:b0ff:fe00:ddc0, enp1s1f0, 00:01:35
O>*       2001:fb6:31c0:13::/64       [110/2]       via fe80::9aa7:b0ff:fe00:dd21, enp1s1f1, 00:01:35
O       2001:fb6:31c0:14::/64   [110/1]   is   directly connected, enp1s1f1, 00:01:35
```

Изменим пропускную способность канала связи между R1 и R2. На интерфейсе enp1s1f0 маршрутизатора R1 программно снизим скорость интерфейса со 100 Мбит/с до 5 Мбит/с, в zebra в режиме глобального

конфигурирования используем команду `bandwidth`, полоса указывается в кбит/с:

```
R1# configure terminal
R1(config)# interface enp1s1f0
R1(config-if)# bandwidth 5000
R1(config-if)# exit
```

Физически скорость интерфейса не изменилась, но маршрутизатор теперь считает скорость интерфейса равной 5 Мбит/с. После этого протокол OSPF снова перестроит маршрут ко всем подсетям через R4. Несмотря на три транзитных участка по новому маршруту через R4 и R3, линия между R1 и R2 все равно будет иметь меньший приоритет из-за низкой скорости передачи, хотя и имеет всего один транзитный участок.

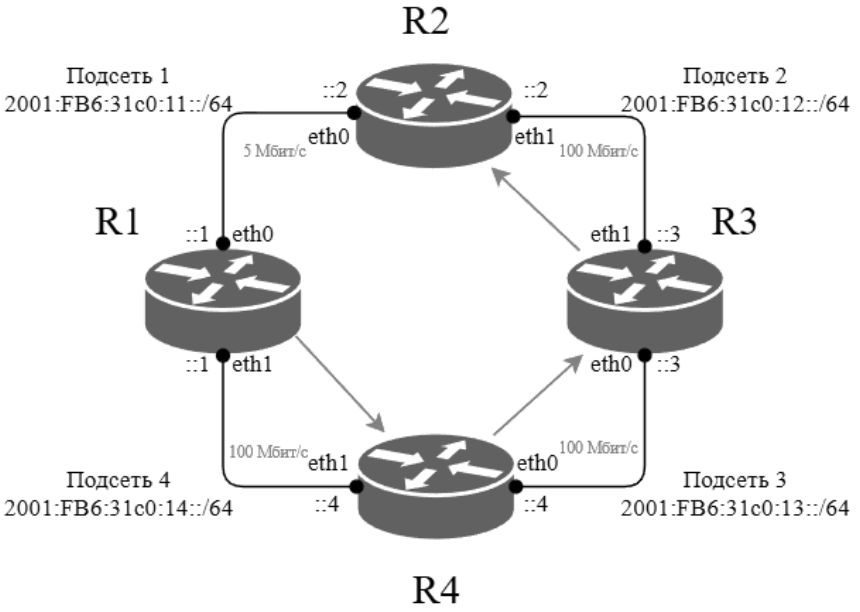


Рисунок 7. Изменение пропускной способности канала между R1 и R2

Таблица маршрутизации на R1 показывает изменения, произошедшие в сети. Теперь маршрут до всех подсетей идет снова идет через fe80::9aa7:b0ff:fe00:dd21, т.е. через маршрутизатор R4:

```
R1# show ipv6 route ospf6
```

```
   Codes: K - kernel route, C - connected, S - static, R
- RIPng, O - OSPFv6, I - IS-IS, B - BGP, A - Babel,> -
selected route, * - FIB route
   O>*       2001:fb6:31c0:11::/64       [110/4]       via
fe80::9aa7:b0ff:fe00:dd21, enp1s1f1, 00:00:33
   O>*       2001:fb6:31c0:12::/64       [110/3]       via
fe80::9aa7:b0ff:fe00:dd21, enp1s1f1, 00:00:33
   O>*       2001:fb6:31c0:13::/64       [110/2]       via
fe80::9aa7:b0ff:fe00:dd21, enp1s1f1, 00:00:33
   O       2001:fb6:31c0:14::/64   [110/1]   is   directly
connected, enp1s1f1, 00:15:12
```

Управление передачей трафика в сети происходит с помощью метрик маршрутов, в таблице маршрутизации метрика маршрута указана вторым числом в квадратных скобках, первое число — это административная дистанция протокола маршрутизации.

Административная дистанция используется маршрутизатором для того, чтобы определить какому протоколу маршрутизации доверять больше, если у обоих протоколов есть маршрут до подсети назначения. Например, более современный протокол OSPFv3 имеет административную дистанцию 110, а протокол RIPng – административную дистанцию 120. Таким образом, если в сети используются два и более протокола маршрутизации, у которых есть свои маршруты к сети назначения, то маршрутизатор выберет маршрут по протоколу с наименьшей административной дистанцией. У статических маршрутов административная дистанция равна единице, так как они прописываются вручную, а значит степень доверия к ним выше.

Если же в сети используется один протокол маршрутизации, и у него есть несколько различных маршрутов до подсети назначения, то маршрутизатор выбирает маршрут уже на основании метрик. Чем меньше метрика, тем больше маршрутизатор доверяет этому маршруту.

У разных протоколов маршрутизации метрика рассчитывается по-разному. Например, у протокола OSPFv3 метрика рассчитывается на основании суммарной цены маршрута, которая в свою очередь зависит от ширины полосы пропускания интерфейсов и каналов связи на маршруте до подсети назначения. А у протокола RIPng метрика рассчитывается исходя из числа транзитных участков между маршрутизатором и подсетью назначения.

Чтобы проверить, как меняется метрика маршрута в сети, на маршрутизаторе R1 программно изменим цену интерфейса enp1s1f1 путем

уменьшения его скорости со 100 Мбит/с до 10 Мбит/с и проанализируем таблицу маршрутизации:

```
R1(config)#
R1(config)# interface enp1s1f1
R1(config-if)# bandwidth 10000
R1(config-if)# exit
```

После изменений трафик все равно будет проходить через маршрутизатор R3, так как суммарная цена маршрута через R3 ниже из-за более высокой пропускной способности каналов, проходящих через него.

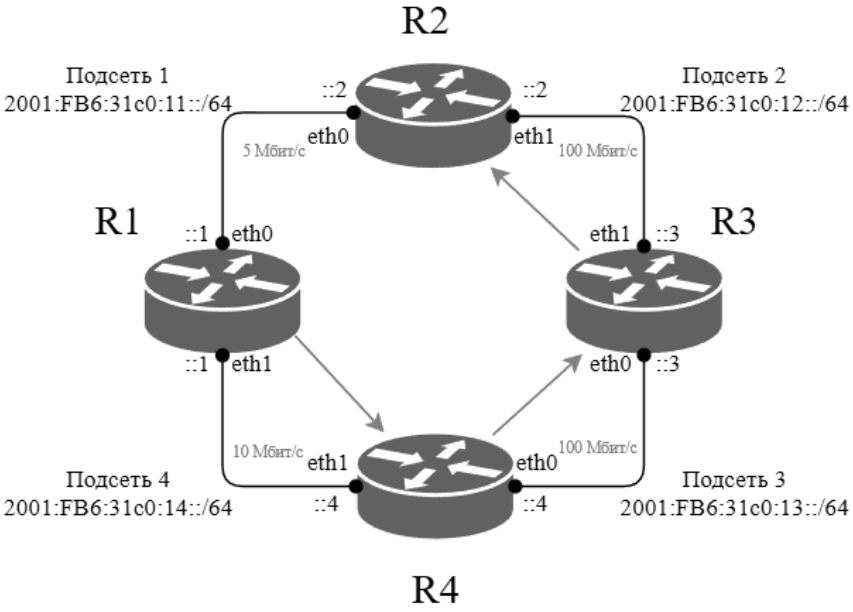


Рисунок 8. Изменение пропускной способности канала между R1 и R4

Уменьшая скорость интерфейса enp1s1f1 маршрутизатора R1, мы увеличиваем цену маршрута на участке сети между R1 и R4, что приводит к увеличению метрик на 9 единиц. Сделанные изменения будут видны в таблице маршрутизации R1:

```
[root@alt219 ~]# telnet 127.0.0.1 2601
R1>
R1> enable
R1# show ipv6 route ospf6
```

```
   Codes: K - kernel route, C - connected, S - static, R
- RIPng, O - OSPFv6, I - IS-IS, B - BGP, A - Babel, > -
selected route, * - FIB route
   O            2001:fb6:31c0:11::/64    [110/13]    via
fe80::9aa7:b0ff:fe00:dd21, enp1s1f1, 00:00:45
   O>*      2001:fb6:31c0:12::/64      [110/12]      via
fe80::9aa7:b0ff:fe00:dd21, enp1s1f1, 00:00:45
   O>*      2001:fb6:31c0:13::/64      [110/11]      via
fe80::9aa7:b0ff:fe00:dd21, enp1s1f1, 00:00:45
   O       2001:fb6:31c0:14::/64  [110/10]  is  directly
connected, enp1s1f1, 00:00:45
```

После снижения скорости интерфейса enp1s1f1, метрики всех маршрутов увеличились, но трафик по-прежнему идет через R4:

```
   [root@alt219 ~]# traceroute6 2001:fb6:31c0:12::2
   traceroute            to            2001:fb6:31c0:12::2
(2001:fb6:31c0:12::2), 30 hops max, 80 byte packets
    1  2001:fb6:31c0:14::4      0.190 ms  0.110 ms  0.074
ms
    2  2001:fb6:31c0:13::3      0.181 ms  0.157 ms  0.150
ms
    3  2001:fb6:31c0:12::2      0.244 ms  0.222 ms  0.219
ms
```

Практикум № 19
Настройка статической маршрутизации с использованием командной строки утилиты nmcli

1. ЦЕЛЬ РАБОТЫ

Познакомиться с понятиями публичных и частных сетей. Изучить механизмы функционирования маршрутизации в ОС Альт. Научиться настраивать статическую маршрутизацию.

2. ЗАДАНИЕ НА РАБОТУ

Соедините три компьютера по схеме, изображённой на рисунке 1.

Наименование интерфейсов указано для примера, на практике оно может отличаться. В нашем же случае `eth0 – enp1s1f0`, а `eth1 – enp1s1f1`.

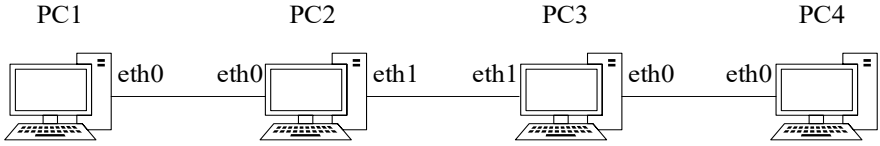


Рисунок 1. Схема соединения компьютеров

Настройте передачу пакетов между сетевыми интерфейсами.

Проверьте правильность настройки маршрутов.

3. ТЕРМИНЫ И ИСПОЛЬЗУЕМЫЕ КОМАНДЫ

Используемые команды

`nmcli` – утилита для управления сетевыми подключениями

`nano` – текстовый редактор.

`ping` – команда для отправки пакетов по указанному сетевому адресу.

`ip r` – команда для настройки маршрутизации.

4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

Для выполнения всех команд необходимо зайти в режим суперпользователя

```
[user@alt218]# su -
Password:12345678
[root@alt218]#
```

Определим, какой адрес необходимо указать для каждой из машин. На всех рабочих местах указан номер. Согласно этому номеру выбирается подсеть по следующему правилу – подсеть для двух компьютеров выбирается в соответствии с минимальным номером рабочего места. Общая формула выбора

`10.10.XY.Z/24`

где

`X` – номер ряда с рабочими местами;

`Y` – минимальный номер между двумя рабочими местами;

`Z` – номер рабочего места.

Пусть схема подключения выглядит следующим образом

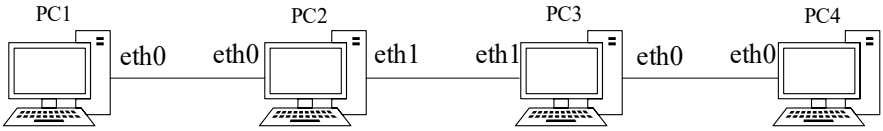


Рисунок 2. Пример схемы нумерации компьютеров

Предположим, что мы проводим настройку 1 ряда. Согласно рисунку 2 и описанному выше правилу, компьютеры получат следующие ip адреса:

PC1:
- `10.10.11.1` на интерфейсе `enp1s1f0`

PC2:
- `10.10.11.2` на интерфейсе `enp1s1f0`
- `10.10.12.2` на интерфейсе `enp1s1f1`

PC3:
- `10.10.12.3` на интерфейсе `enp1s1f1`
- `10.10.13.3` на интерфейсе `enp1s1f0`

PC4:
- `10.10.13.4` на интерфейсе `enp1s1f0`

Для присвоения ip – адресов на необходимые интерфейсы, с последующим их подключением, воспользуемся утилитой “nmcli”

Добавим необходимые интерфейсы и настроим согласно рисунку 2. Рассмотрим на примере PC1 и PC2:

PC1:

```
[root@alt219]# nmcli c add type Ethernet con-name enp1s1f0 ifname enp1s1f0 ip4 10.10.11.1/24
```

Теперь активируем настроенный интерфейс

```
[root@alt219]# nmcli device connect enp1s1f0
```

PC2:

```
[root@alt218]# nmcli c add type Ethernet con-name enp1s1f0 ifname enp1s1f0 ip4 10.10.11.2/24
[root@alt218]# nmcli c add type Ethernet con-name enp1s1f1 ifname enp1s1f1 ip4 10.10.12.2/24
```

Теперь активируем настроенные интерфейсы:

```
[root@alt218]# nmcli device connect enp1s1f0
[root@alt218]# nmcli device connect enp1s1f1
```

Проверить правильность присвоения ip – адресов соответствующим интерфейсам можно при помощи команды:

```
[root@alt218]# ip a
```

Для PC3 и PC4 проделайте аналогичные операции самостоятельно.

На компьютерах PC2 и PC3 включим пересылку пакетов между сетевыми интерфейсами, изменив в файле `sysctl.conf` параметр `net.ipv4.ip_forward` на 1. Для этого откроем файл, введя команду

```
[root@alt218]# nano /etc/net/sysctl.conf
```

И изменим значение параметра `net.ipv4.ip_forward` на 1, если значение уже не равно единице (рисунок 3).

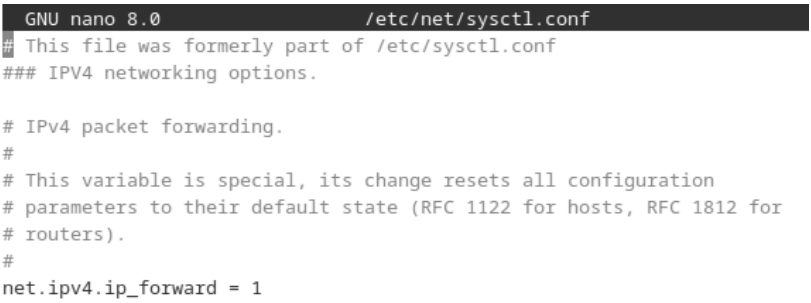


Рисунок 3. Фрагмент файла `sysctl.conf`

Далее, после записи в файле, сохраним введённые данные с помощью комбинации `CTRL+O` и выходим из редактирования – `CTRL+X`.

Пропишем следующие маршруты в консоли для каждого из компьютеров в соответствии с примером из рисунка 2.

PC1: `ip r add 10.10.12.0/24 via 10.10.11.2 dev enp1s1f0`
PC1: `ip r add 10.10.13.0/24 via 10.10.11.2 dev enp1s1f0`
PC2: `ip r add 10.10.13.0/24 via 10.10.12.3 dev enp1s1f1`
PC3: `ip r add 10.10.11.0/24 via 10.10.12.2 dev enp1s1f1`
PC4: `ip r add 10.10.12.0/24 via 10.10.13.3 dev enp1s1f0`
PC4: `ip r add 10.10.11.0/24 via 10.10.13.3 dev enp1s1f0`

Перезапустим сетевую службу на всех PC для применения параметров

```
[root@alt218]# service network restart
```

Для проверки правильности выполнения работы используем команду `ping`:

Для PC1:

```
[root@alt219]# ping 10.10.13.4
```

Для PC4:

```
[root@alt216]# ping 10.10.11.1
```

Список литературы

Основная литература

1. Олифер В.Г. Компьютерные сети. Принципы, технологии, протоколы, 4-е издание. - М.: Питер, 2014. - 943с.
2. Семенов Ю.А. Протоколы и алгоритмы маршрутизации в Интернет [Электронный ресурс]/ Семенов Ю.А.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 998 с.— Режим доступа: http://www.iprbookshop.ru/62826.html.— ЭБС «IPRbooks

Дополнительная литература

3. Берлин А.Н. Основные протоколы Интернет [Электронный ресурс]/ Берлин А.Н.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 602 с.— Режим доступа: http://www.iprbookshop.ru/52181.— ЭБС «IPRbooks»
4. Зиглер Роберт Л., Брандмауэры в Linux. : Пер. с англ. : М. : Издательский дом “Вильямс”, 2001. – 384 с.
5. Oskar Andreasson, Руководство по iptables (Iptables Tutorial 1.1.19), : Пер. С англ. : Андрей Киселёв, 2003. – 137 с.
6. Сетевые средства Linux [Электронный ресурс] // ИНТУИТ, Национальный открытый университет. URL: https://www.intuit.ru/studies/courses/681/537/info
7. Основные протоколы интернет [Электронный ресурс] // ИНТУИТ, Национальный открытый университет. URL: https://www.intuit.ru/studies/courses/2/2/info
8. Операционные системы ОС Альт [Электронный ресурс] // Базальт СПО. URL: https://www.basealt.ru/products/alt-server/
9. Postel, J. (ed.), "Internet Protocol - DARPA Internet Program Protocol Specification," RFC 791, USC/Information Sciences Institute, September 1981. - 45 p. https://tools.ietf.org/html/rfc791
10. J. Postel. “User Datagram Protocol”, RFC 768, 1980, - 3 p. https://tools.ietf.org/html/rfc768
11. TRANSMISSION CONTROL PROTOCOL - DARPA Internet Program Protocol Specification, RFC 793, 1981. – 85 p. https://tools.ietf.org/html/rfc793
12. INTERNET CONTROL MESSAGE PROTOCOL - DARPA Internet Program Protocol Specification, RFC 792, 1981, - 21 p. https://tools.ietf.org/html/rfc792
13. Internet Protocol, Version 6 (IPv6) Specification Internet Engineering Task Force (IETF) RFC 8200, 2017. – 41 p. https://tools.ietf.org/html/rfc8200