Применение программного комплекса «IMUNES» для моделирования компьютерных сетей и сетей IoT

Общая информация об IMUNES

Integrated Multiprotocol Network Emulator/Simulator — интегрированный многопротокольный комплекс эмуляции/симуляции сетевых топологий. IMUNES разработан в Загребском Университете (University of Zagreb) и представляет собой альтернативу реальным экспериментальным сетям. К основным преимуществам этого инструмента относятся: высокая масштабируемость, производительность и точность. Инфраструктура связи (звенья, коммутаторы и концентраторы) выполнена с помощью *netgraph*, API уровня ядра для манипулирования трафиком, включенным в ядро FreeBSD. В Linux IMUNES работает на инструментах Docker (ПО для автоматизации управления программ в средах с поддержкой контейнеризации) и Open vSwitch (программный многоуровневый коммутатор с открытым исходным кодом). Все виртуальные узлы совместно используют один системный тактовый генератор, что обеспечивает корреляцию захвата сети между ними [11].

Проведение моделирования на основе IMUNES подразумевает отделение спецификации топологии и оболочки графического интерфейса от утилиты управления. Данный подход способствует реалистичному и детальному эмулированию сетевых маршрутизаторов. Также он позволяет каждому виртуальному узлу запускать личную копию любого немодифицированного приложения пользовательского уровня, включая демоны протокола маршрутизации, генераторы и анализаторы трафика [12].

Ключевые особенности IMUNES:

- открытый исходный код и бесплатное приложение;
- моделирование IP-сетей в реальном времени на гигабитных скоростях;
- масштабируемая архитектура для крупных экспериментов;
- до 1000 виртуальных узлов на одном физическом компьютере, каждый из которых способен выполнять приложения UNIX;

- все узлы в IMUNES работают как реальные физические узлы, которые имеют собственные: IPv4, IPv6, IPsec, сокет, брандмауэр, конфигурации;
- возможность манипулирования трафиком: задержка, коэффициент битовых ошибок, дублирование, полоса пропускания;
- эффективное использование аппаратных ресурсов процессора.

Установка программного комплекса «IMUNES»

Перед выполнением установки важно учесть, что построить модель сети с помощью IMUNES возможно на любой системе с поддержкой Tcl/Tk (FreeBSD, Linux, Windows, macOS). Однако эксперимент (выполнение модели) может быть запущен только на операционных системах FreeBSD и Linux с правами суперпользователя (root permission). При несоблюдении данного условия возникают ошибки, указанные на рисунках 1 и 2.

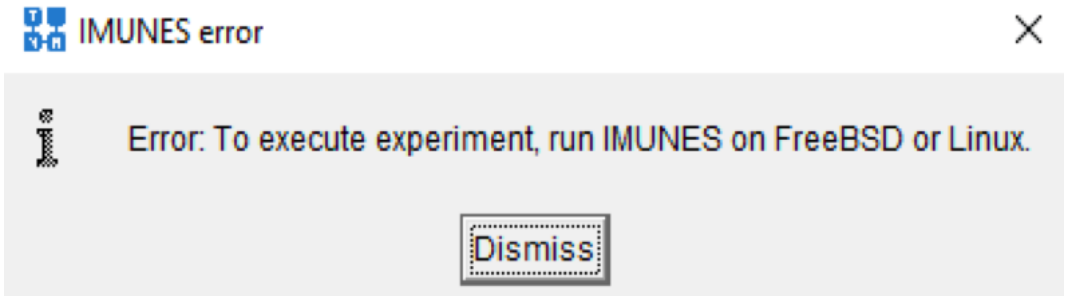

Рисунок 1. Ошибка при запуске эксперимента на Windows

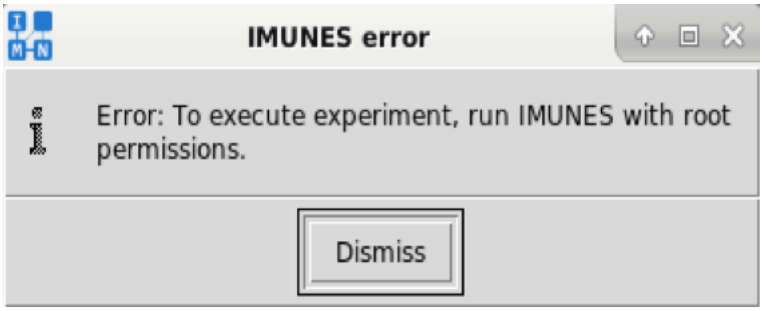

Рисунок 2. Ошибка при запуске эксперимента на FreeBSD без прав суперпользователя

Установка на FreeBSD

Процесс установки IMUNES начинается с загрузки пакетов, необходимых для работы программного комплекса. Вводится команда:

```
# pkg install tk86 ImageMagick tcllib wireshark socat git
gmake
```

Загрузка последней версии IMUNES через общедоступный репозиторий GitHub происходит с помощью команды:

```
# git clone https://github.com/imunes/imunes.git
```

Затем нужно установить IMUNES и заполнить виртуальную файловую систему заранее заданными данными. Для этого в режиме суперпользователя выполняются команды:

```
# cd imunes
# make install
# imunes -p
```

Установка на Linux

Перед загрузкой IMUNES на Linux следует установить следующие пакеты: tcl (версия 8.6 или выше), tk (версия 8.6 или выше), tcllib, Wireshark (с графическим интерфейсом), ImageMagick, Docker (версия 1.6 или выше), Open vSwitch, nsenter, xterm, make. В зависимости от дистрибутива Linux последовательно выполняются представленные ниже команды [13].

Для Arch

Установка пакетов Tcl/Tk, Wireshark, ImageMagick, Docker, Open vSwitch, make, xterm:

```
# pacman -S tk tcllib wireshark-gtk imagemagick docker
\make openvswitch xterm
```

Для Debian 9

Обновление информации о пакетах в репозиториях:

```
# apt-get update
```

Подготовка к загрузке пакетов с веб-узла:

```
# apt-get install apt-transport-https ca-certificates
curl gnupg2 software-properties-common
```

Загрузка пакетов Docker:

```
# curl -fsSL https://download.docker.com/linux/debian/gpg
| apt-key add -
```

Добавление нового репозитория:

```
# add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/debian $(lsb_release -
cs) stable"
```

Обновление информации о пакетах в репозиториях и установка пакетов Docker, Open vSwitch, xterm, Wireshark, ImageMagick, Tcl/Tk, служебных утилит Linux, make:

```
# apt-get update

# apt-get install docker-ce openvswitch-switch xterm
wireshark \ imagemagick tk tcllib util-linux make
```

Для Ubuntu 18.04 LTS (Mint 19, 19.1)

Установка пакетов Open vSwitch, Docker, xterm, Wireshark, ImageMagick, Tcl/Tk, служебных утилит Linux:

```
# apt install openvswitch-switch docker.io xterm
wireshark \ make imagemagick tk tcllib util-linux
```

Для Ubuntu 15.04

Установка пакетов Open vSwitch, Docker, xterm, Wireshark, ImageMagick, Tcl/Tk, служебных утилит Linux:

```
# apt-get install openvswitch-switch docker.io xterm
wireshark \ make ImageMagick tk tcllib user-mode-linux
util-linux
```

После загрузки и установки на Linux необходимых пакетов нужно загрузить файлы IMUNES с GitHub с помощью команды:

```
# git clone https://github.com/imunes/imunes.git
```

Затем происходит установка IMUNES на Linux по аналогии с FreeBSD. Для этого в режиме суперпользователя выполняются команды:

```
# cd imunes
# make install
# imunes -p
```

Установка на Windows и macOS

Чтобы программный комплекс полноценно работал на Windows и macOS, необходимо загрузить виртуальную машину FreeBSD с предустановленным IMUNES с веб-узла: http://imunes.net/download. Для установки необходимо наличие программы Oracle VM VirtualBox и 4 Гб свободного пространства на жестком диске.

Запуск IMUNES

После корректной установки IMUNES готов к работе. ПО запускается через терминал операционной системы в режиме суперпользователя. Для входа в данный режим на исследуемой ОС Debian GNU Linux 9 вводим:

```
# sudo su
```

Затем вводим пароль и нажимаем клавишу Enter. Чтобы запустить IMUNES выполним команду:

```
# imunes
```

Для более удобного запуска программы создадим ярлык на рабочем столе. Сначала нужно активировать показ иконок в разделе Desktop утилиты Tweak Tool. Напротив надписи Icons on Desktop переводим ползунок в положение ON. Затем в приложении Text Editor создаем новый текстовый файл в который прописываем следующую команду:

```
# xterm -e sudo imunes
```

Сохраняем файл на рабочем столе под именем «IMUNES». При щелчке ПКМ на файле открывается его меню конфигурации, в котором нужно выбрать пункт Properties. Во вкладке Permissions открывшегося окна IMUNES Properties ставим галочку напротив пункта Allow executing file as program. Чтобы установить иконку IMUNES на созданный файл, переходим во вкладку Basic и выбираем нужное изображение. Закрываем окно свойств.

Теперь при двойном щелчке ПКМ по иконке IMUNES на рабочем столе открывается окно эмулятора терминала в котором происходит запуск программного комплекса в режиме суперпользователя.

Пользовательский интерфейс

IMUNES GUI — это простая консоль управления на основе Tcl/Tk, позволяющая строить топологии виртуальных сетей и управлять ими. Составляющие окна IMUNES: рабочее пространство, называемое Canvas, меню в верхней части, панель инструментов слева и строка состояния внизу.

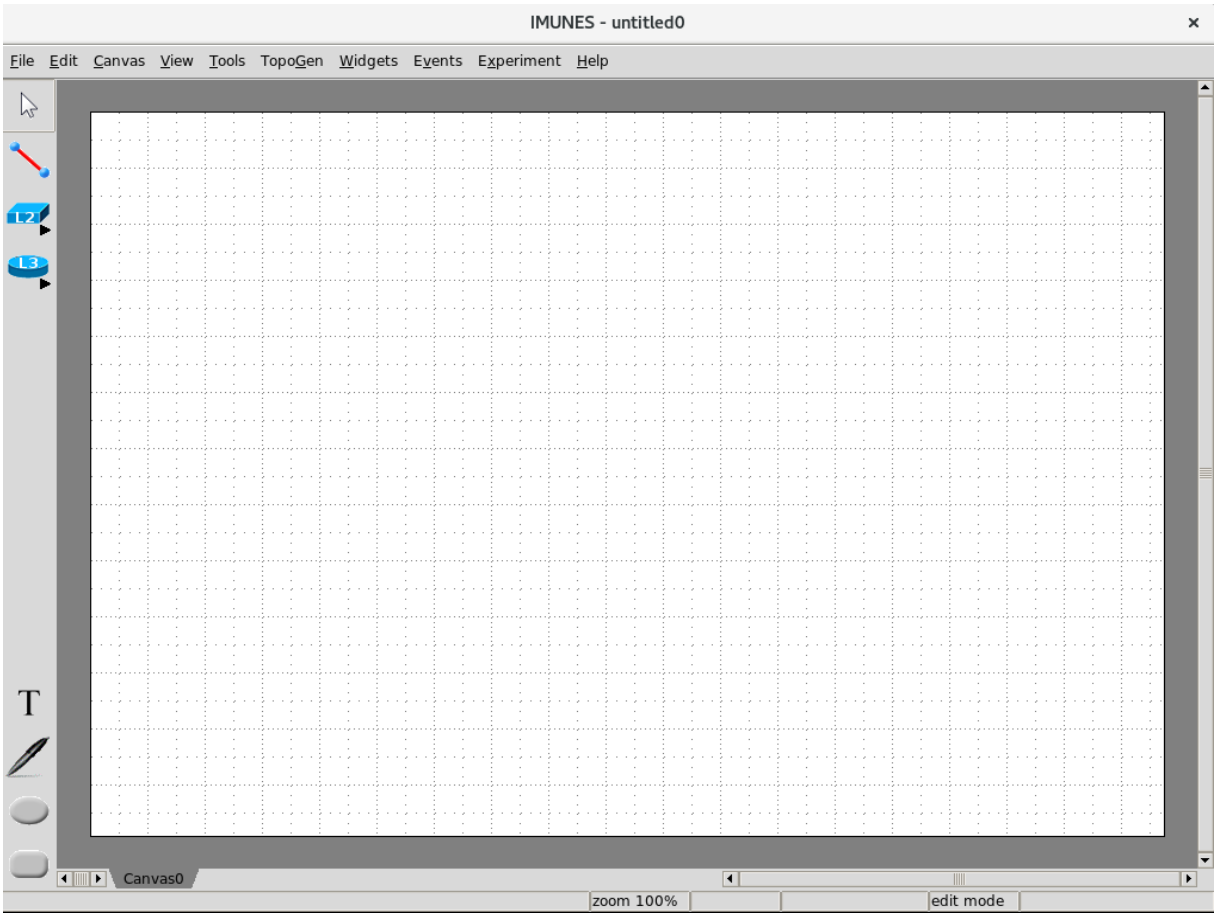

Рисунок 3. Скриншот окна IMUNES

После запуска программы или после создания нового рабочего пространства пользователь находится в режиме редактирования. Режим используется для построения и настройки сети, в отличие от режима выполнения, целью которого является моделирование сети.

Панель инструментов расположена в левой части графического интерфейса пользователя. Она содержит инструменты для проектирования сетевых топологий и для добавления аннотаций. Ниже приведено описание каждого элемента.

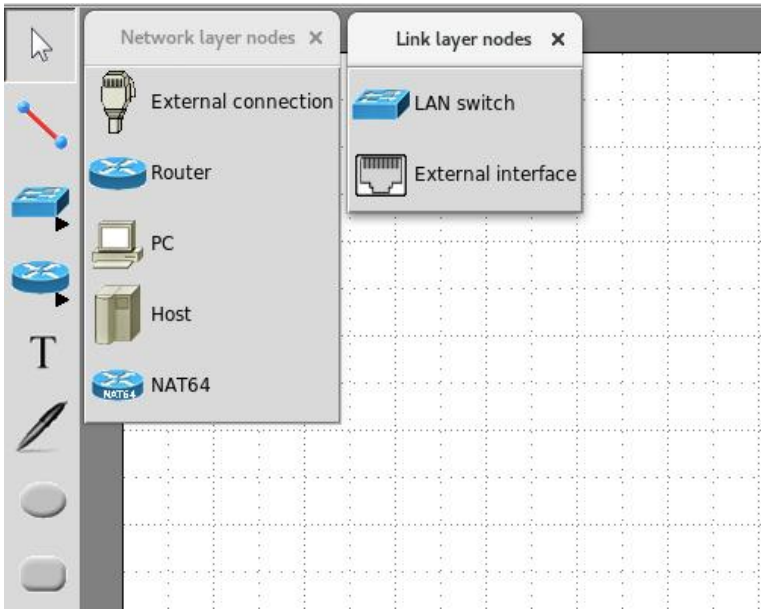

Рисунок 4. Панель инструментов IMUNES

Select tool (Инструмент выбора) — инструмент по умолчанию для выбора и перемещения элементов. Чтобы выбрать необходимый узел, нужно щелкнуть по нему ЛКМ. Чтобы переместить — нажать на нем ЛКМ и, не отпуская, перетащить в нужную часть рабочей области. По щелчку ПКМ открывается меню конфигурации элемента.

Link (Соединение) — инструмент, который используется для создания сетевых связей между узлами. Чтобы соединить узлы, нужно щелкнуть по инструмент ЛКМ, затем нажать на один из соединяемых узлов и перетащить линию в направлении другого.

External connection (Внешнее подключение) — инструмент, который дает возможность соединить хост-компьютер пользователя с виртуальным узлом путем создания интерфейса на компьютере пользователя.

Router (Маршрутизатор) — данный элемент сетевого уровня создает виртуальную машину, настроенную на выполнение роли маршрутизатора в сети. Router сконфигурирован для работы с протоколами динамической маршрутизации и перенаправления принятых пакетов на основе информации заголовка IP в каждом пакете.

PC (Персональный компьютер) — элемент сетевого уровня, который создает виртуальную машину моделирующую клиентский компьютер. PC сконфигурирован со статическим маршрутом по умолчанию, не пересылает пакеты и не выполняет никаких протоколов маршрутизации.

Host (Хост) — элемент сетевого уровня, создающий виртуальную машину для симуляции хост-сервера в тестовой сети. Он также не пересылает пакеты и имеет статические маршруты. Однако, в отличие от PC, Host запускает сетевые сервисы, такие как SSH.

NAT64 — узел маршрутизатора, который способен осуществлять трансляцию между протоколами IPv4 и IPv6.

LAN switch (Коммутатор локальной сети) — элемент канального уровня, который пересылает входящие пакеты на подключенные узлы, используя таблицу адресов назначения и ее порты.

External interface (Внешний интерфейс) — инструмент, который обеспечивает возможность подключения виртуального узла к физическому интерфейсу.

Text (Текст) — инструмент для добавления текста на рабочую область.

Freeform (Свободная форма) — инструмент для добавления новой формы на рабочую область.

Oval (Овал) — инструмент для добавления новой овальной формы.

Rectangle (Прямоугольник) — инструмент для добавления новой прямоугольной формы на рабочую область.

На рисунке 5 изображено меню программы, которое обеспечивает доступ к различным функциям. Некоторые параметры в строке меню автоматически отключаются в режиме выполнения [14].

Рисунок 5. Меню IMUNES

Пример моделирование сети

- Персональные компьютеры (test-pc1 и test-pc2) из сети 192.168.1.0/24 подключены к коммутатору локальной сети (test-switch), который подключен к маршрутизатору (test-router).
- Хост (test-host) из сети 192.168.2.0/24 напрямую подключен к маршрутизатору (test-router).
- Персональные компьютеры из первой сети имеют маршрут только к сети 192.168.2.0/24. Сервер из второй сети имеет маршрут по умолчанию.
- Маршрутизатор сконфигурирован с помощью пакета Quagga, чтобы иметь возможность обслуживать и получать динамические обновления маршрута.

Построение данной сети будет производиться на операционной системе Debian GNU Linux 9 с использованием инструментов, описанных в предыдущем разделе.

Построение топологии

Чтобы разместить узел, выберем соответствующий инструмент узла, а затем щелкнем ЛКМ на рабочем пространстве. Таким образом добавим маршрутизатор, коммутатор локальной сети, хост и два компьютера. С помощью инструмента Link подключаем коммутатор к маршрутизатору, каждый компьютер к коммутатору, а затем хост напрямую к роутеру.

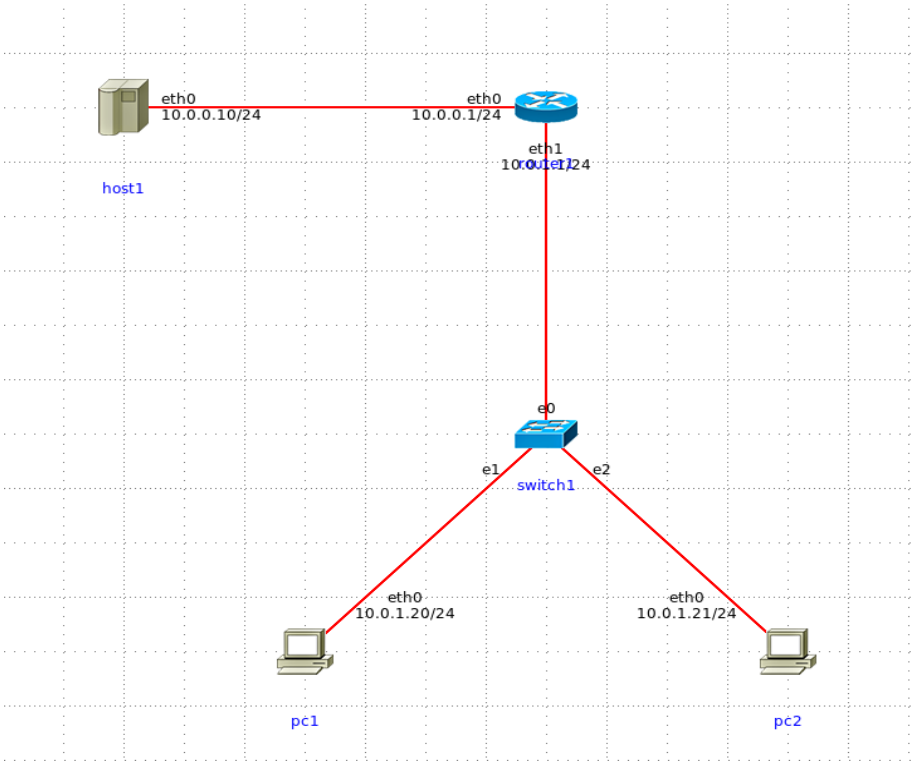

Рисунок 6. Топология экспериментальной локальной сети

Когда узлы соединены с помощью инструмента Link, исходный узел, узел назначения и соединение автоматически получают предварительно сконфигурированные параметры. Если указатель мыши находится над узлом или соединением, некоторые из настроенных параметров отображаются в левой части строки состояния в нижней части окна.

Рисунок 7. Строка состояния окна IMUNES

Имена интерфейсов, IPv4 / IPv6-адреса элементов сетевого уровня, имена узлов видны на рабочей области. Свойства видимости параметров узлов и ссылок задаются в меню View.

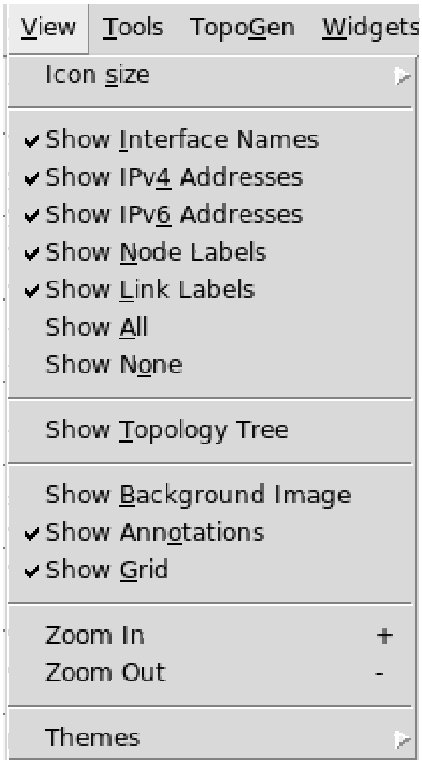

Рисунок 8. Параметры меню View

Чтобы удалить элемент сети, нужно выбрать его с помощью инструмента Select tool, а затем нажать клавишу Delete. Так же произвести удаление можно, щелкнув по элементу ПКМ и щелкнув по пункту Delete в появившемся меню. После удаления узла автоматически следует удаление соединений с ним.

Чтобы переместить сетевой элемент, выделяем его инструментом выбора и перетаскиваем в назначенную позицию. Для изменения положения имени узла нужно также выделить его и перетащить. С помощью инструмента Select tool возможно перемещение группы связанных узлов, которые выбираются нажатием ЛКМ в дополнение к нажатию клавиши Ctrl. За выбор топологии всей сети отвечает пункт Select all в меню Edit.

Настроим параметры моделируемой сети. Меню конфигурации сетевого элемента открывается по двойному щелчку ЛКМ на этом элементе. Или же щелчком ПКМ по элементу с последующим выбором пункта Configure в появившемся меню.

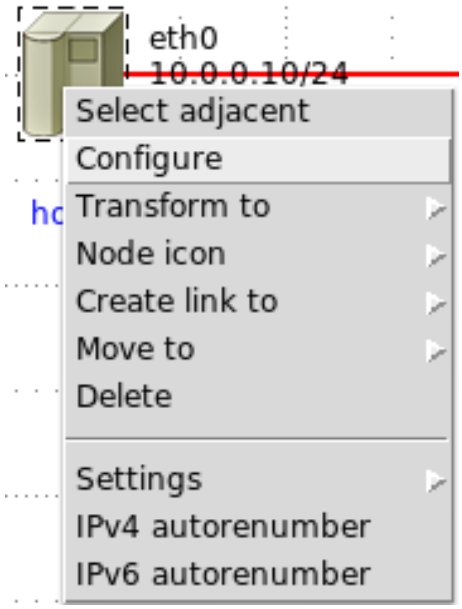

Рисунок 9. Меню конфигурации сетевого элемента в режиме редактирования

Для начала переименуем узлы на test-host, test-router, test-switch, test-pc1 и test-pc2 соответственно. Затем для коммутатора изменим способ планирования пакетов данных с FIFO («первым пришел — первым обслужен») на WFQ (метод взвешенной справедливой очереди). Выделяем интерфейс канального уровня e0 из списка интерфейсов, выбираем пункт WFQ в меню Queue и щелкаем по кнопке нажимаем Apply. Повторяем процедуру для других интерфейсов канального уровня.

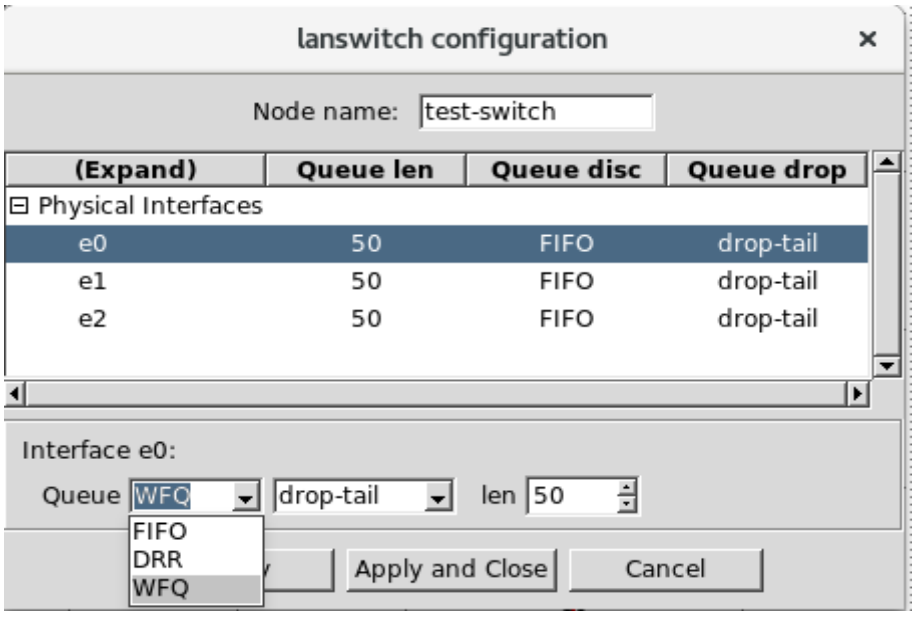

Рисунок 10. Окно конфигурации коммутатора локальной сети

Окно конфигурации ПК/Хоста/Роутера состоит из двух вкладок: Configuration и Interfaces. Изменим параметры сетевого интерфейса и параметры маршрутизации. Чтобы изменить адрес IP, на вкладке Interfaces выбираем интерфейс eth0 в поле адреса IPv4 вводим нужные значения и щелкаем по кнопке Apply. Меняем IP-адрес хоста на 192.168.2.5/24, а IP-адреса ПК на 192.168.1.5/24 и 192.168.1.7/24.

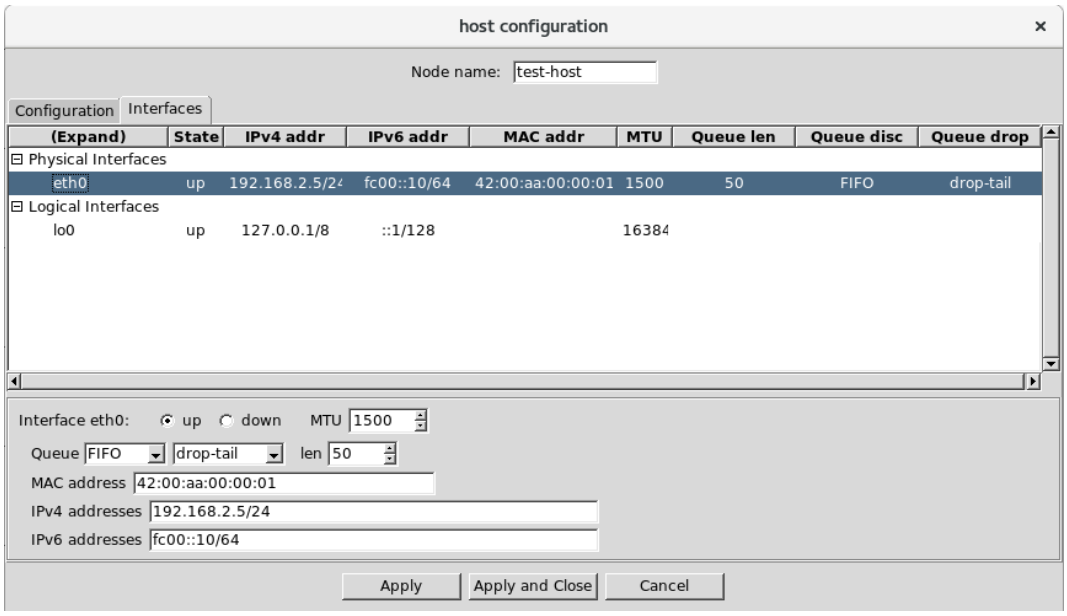

Рисунок 11. Окно конфигурации хоста локальной сети

Компьютеры и хосты используют статическую маршрутизацию. Предварительно сконфигурированная таблица маршрутизации содержит только маршрут по умолчанию. Если синтаксис маршрута неверен, этот маршрут будет игнорироваться. Статические маршруты и маршруты по умолчанию состоят из:

1) Сети назначения: IP-адрес, за которым следует косая черта и префикс сети.
2) IP-адреса сетевого интерфейса следующего перехода (без косой черты и без префикса сети).

Добавим статический маршрут на test-pc1 и test-pc2 для сети 192.168.2.0/24 через шлюз 192.168.1.1.

Рисунок 12. Окно со статическими маршрутами ПК

На office-host мы изменим адрес шлюза по умолчанию на 192.168.2.1.

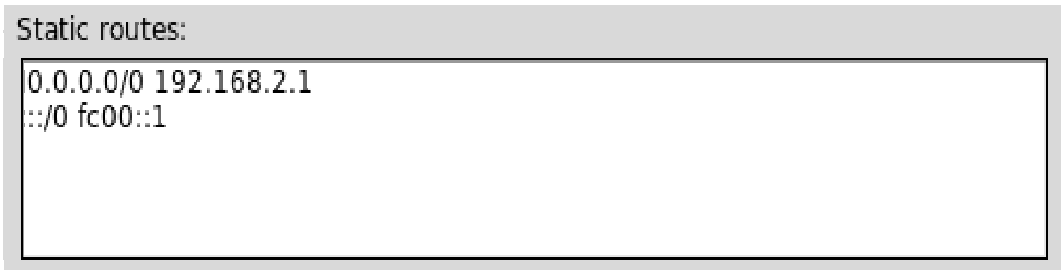

Рисунок 13. Окно со статическими маршрутами хоста

Чтобы применить измененную конфигурацию и закрыть окно конфигурации, щелкаем по кнопке Apply and Close.

Окно конфигурации маршрутизатора, помимо вкладок описанных выше, содержит область для выбора модели маршрутизации и протоколов, а также вкладку IP Security. Изменим IPv4-адреса на обоих сетевых интерфейсах: 192.168.1.1/24 на интерфейсе eth1 и 192.168.2.1/24 на сетевом интерфейсе eth0. Модель маршрутизации Quagga и протоколы оставим установленными по умолчанию.

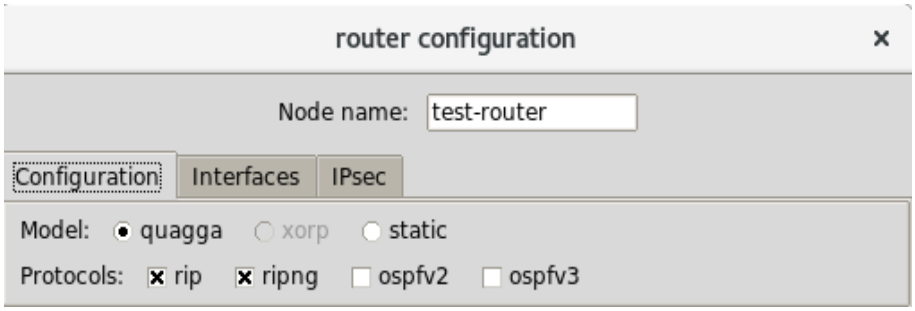

Рисунок 14. Окно конфигурации маршрутизатора локальной сети

Окно конфигурации соединения предлагает возможность задавать полосу пропускания канала, задержку распространения, частоту ошибок по битам и вероятность дублирования пакетов. Существуют также свойства отображения: ширина линии соединения и цвет линии.

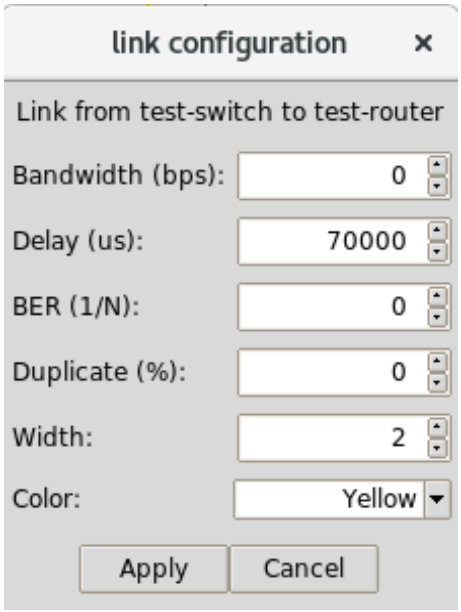

Рисунок 15. Окно конфигурации соединения

Поменяем цвет линий на желтый. Значения остальных параметров оставим по умолчанию, кроме параметров соединения между test-router и test-switch. Для этого соединения установим задержку 70000 мкс. Задержка будет проверена во время моделирования сети с помощью команды traceroute. Настроенная сеть выглядит так, как показано на рисунке 16.

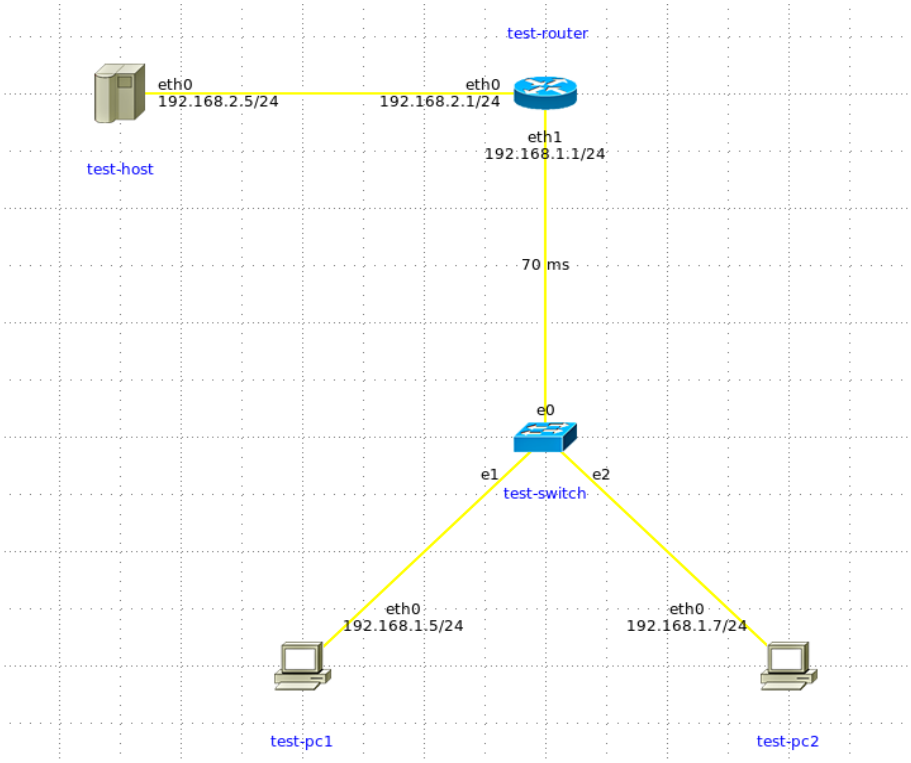

Рисунок 16. Топология локальной сети с заданными параметрами

Выполнение эксперимента

После построения и правильной настройки топологии сети выполним моделирование. В меню Experiment выбираем команду Execute. IMUNES переключается из режима редактирования в режим выполнения. В процессе запуска эксперимента программный комплекс создает и конфигурирует виртуальную сеть. Информация о времени, потраченном на создание топологии сети, и уникальный идентификатор эксперимента отображаются в строке состояния.

Рисунок 17. Строка состояния IMUNES в режиме выполнения

В правом углу строки можно увидеть, что IMUNES теперь работает в режиме выполнения. Инструменты добавления узлов и соединений перестают

быть активными. Меню элементов в режиме выполнения отличается от меню в режиме редактирования. Оно предлагает следующие возможности:

- выбрать узел, подключенный к данному узлу;
- просмотреть текущую конфигурацию;
- запустить / остановить / перезапустить элемент сети;
- запустить / остановить / перезапустить любую из возможных служб или импортировать запуск конфигурации из меню настроек;
- открыть консоль, анализаторы сети Wireshark или tcpdump на любом из интерфейсов, в браузере Firefox или в почтовом клиенте.

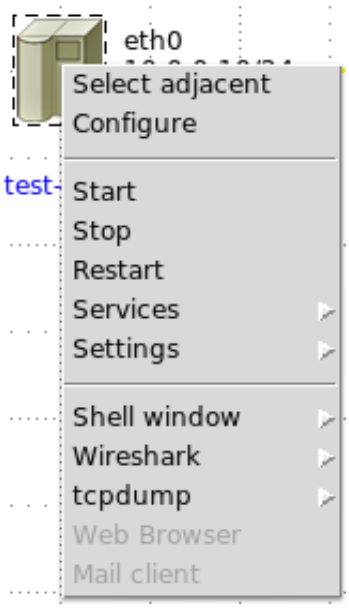

Рисунок 18. Меню конфигурации сетевого элемента в режиме выполнения

Из меню конфигурации узла в режиме выполнения можно изменить только имя узла. Параметры интерфейса канального уровня, параметры сетевого интерфейса и параметры маршрутизации, могут быть изменены из консоли на каждом узле. Чтобы изменить эти параметры из меню конфигурации узла, нужно остановить узел (с помощью пункта Stop), изменить его параметры и запустить узел (щелчком по пункту Start).

Теперь мы проверим правильность настройки топологии виртуальной сети. Открываем консоль двойным щелчком ЛКМ по сетевому элементу, например по test-host.

Для проверки параметров сетевого интерфейса eth0, вводим команду:

```
# ifconfig eth0
```

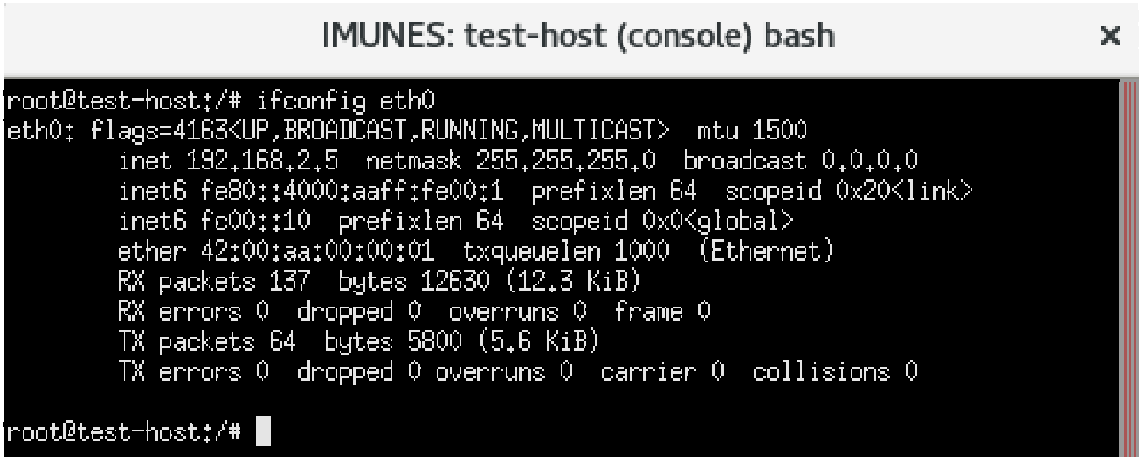

Рисунок 19. Результат выполнения команды *ifconfig* в консоли хоста

Для проверки доступности конкретного сетевого элемента, например, test-pc1 выполняем следующую команду:

```
# ping 192.168.1.5
```

Результат показан на рисунке 20. Для остановки передачи пакетов нужно нажать сочетание клавиш Ctrl+C.

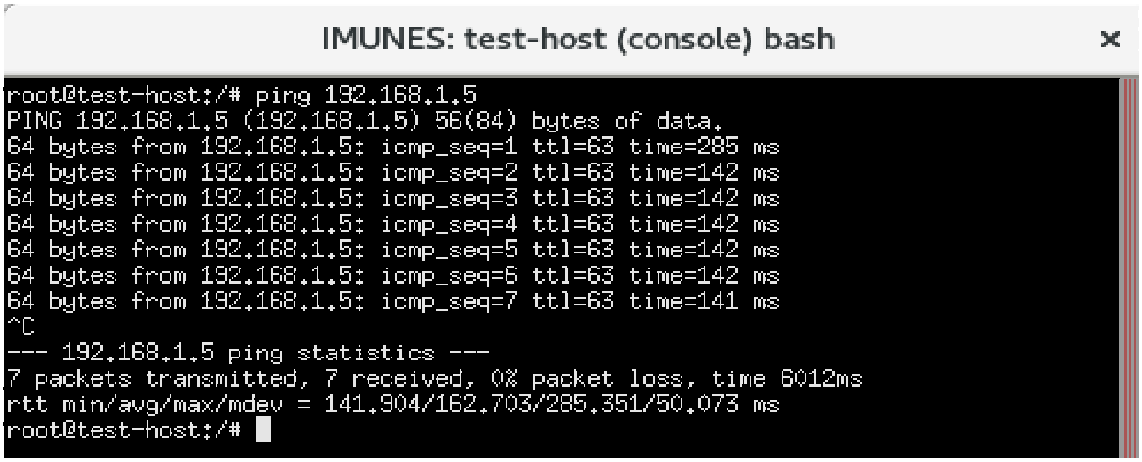

Рисунок 20. Результат выполнения команды *ping* в консоли хоста

Проверим задержку соединения между маршрутизатором и коммутатором, которая установлена на 70000 мкс (70 мс). В консоли на test-pc1 вводим команду:

```
# traceroute 192.168.1.1
```

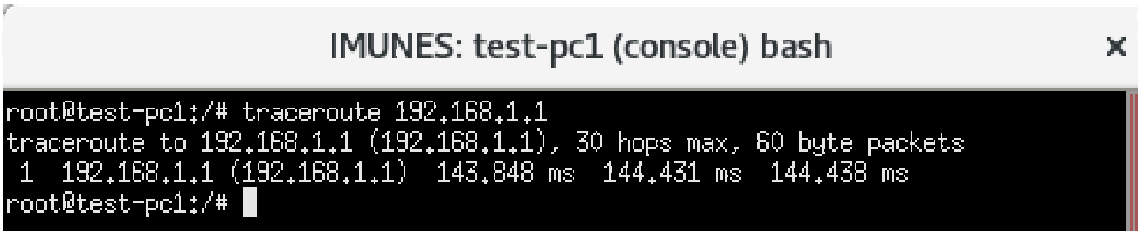

Рисунок 21. Результат выполнения команды *traceroute* в консоли ПК

В консоли на test-host вводим следующую команду:

```
# traceroute 192.168.2.1
```

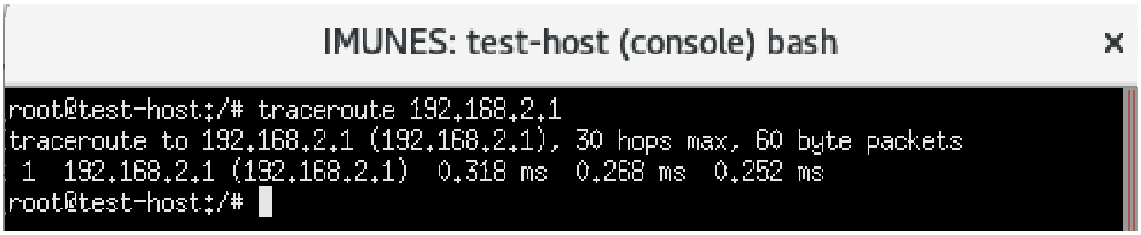

Рисунок 22. Результат выполнения команды *traceroute* в консоли хоста

Чтобы завершить эксперимент и переключиться из режима выполнения в режим редактирования, открываем вкладку меню Experiment и щелкаем по пункту Execute. IMUNES завершает активные сервисы на каждом узле и закрывает все элементы сети. Процесс прекращается, когда в строке состояния появляется сообщение об успешной очистке.

Рисунок 23. Строка состояния IMUNES при переходе в режим редактирования

После построения, настройки и тестирования виртуальной сети ее можно сохранить щелчком по пункту Save вкладки меню File. Топология виртуальной сети сохраняется в формате файла конфигурации IMUNES (.imn). Чтобы в дальнейшем открыть существующий файл конфигурации, нужно щелкнуть по пункту Open вкладки меню File, выбрать его в открывшемся диалоговом окне и щелкнуть по кнопке Open file.

Лабораторная работа №1. Ознакомление с интерфейсом программы «IMUNES» и построение простейшей сети

1 Цель работы

Ознакомление студентов с основными возможностями программного комплекса посредством моделирования простой сетевой топологии.

2 Задачи

Изучить структуру окна и назначение инструментов. Построить сеть из 4 персональных компьютеров и 1 коммутатора. Задать адреса сетевых узлов, проверить их доступность с помощью утилиты *ping*.

3 Порядок выполнения лабораторной работы

1. Запустить программный комплекс «IMUNES», выполнив в терминале команду *imunes* либо дважды щелкнув ЛКМ по ярлыку программы.
2. В произвольном порядке ознакомиться с пунктами меню, попробовать работу каждого инструмента, просмотреть параметры конфигурации сетевых элементов.
3. Создать новую рабочую область: Canvas ➡ New.
4. Изменить имя рабочей области с «Canvas1» на «Simple Network», дважды щелкнув по соответствующей вкладке в нижней части окна.
5. В правый верхний угол рабочей области добавить фигуру произвольного цвета с помощью инструмента Rectangle. Размер рабочей области можно изменить: Canvas ➡ Resize.
6. Используя инструмент Text, добавить на фигуру подпись с ФИО студента и номером группы.
7. С помощью инструмента PC добавить на рабочую область 4 компьютера.

8. С помощью инструмента LAN Switch добавить на рабочую область коммутатор.
9. Соединить все компьютеры с напрямую с коммутатором, используя инструмент Link.
10. Открыть окно конфигурации ПК двойным щелчком ЛКМ по его иконке и во вкладке Interfaces задать каждому компьютеру соответствующий IPv4-адрес и маску подсети:
    - pc1: 192.168.1.1/24;
    - pc2: 192.168.1.2/24;
    - pc3: 192.168.1.3/24;
    - pc4: 192.168.1.4/24.
11. Перевести IMUNES в режим выполнения: Experiment ➞ Execute.
12. Проверить, соответствует ли построенная модель сети той топологии, которая изображена на рисунке 24.

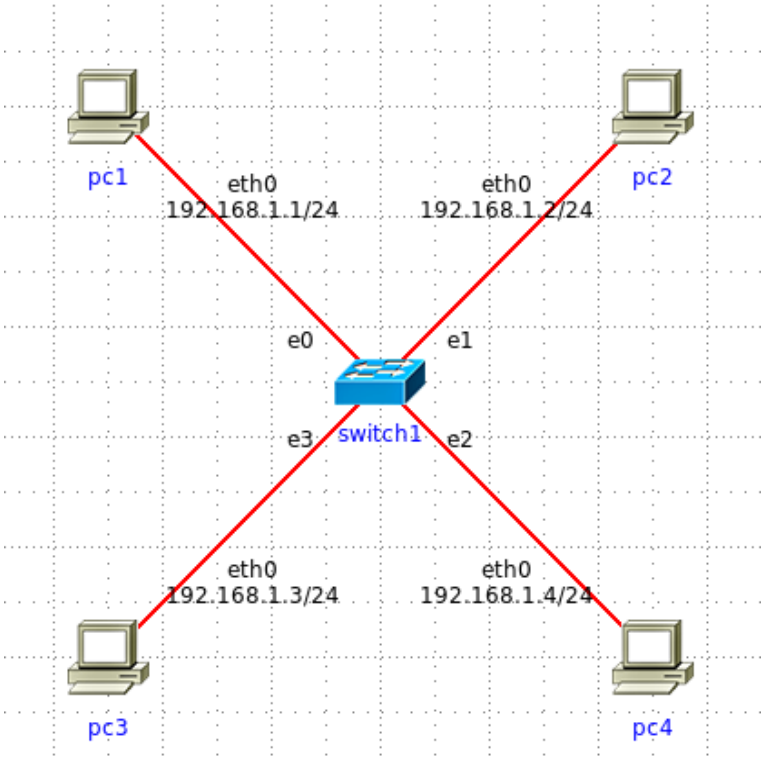

Рисунок 24. Топология сети для лабораторной работы №1

13. На узле pc1 открыть консоль двойным щелчком ЛКМ по изображению компьютера.

14. Для проверки доступности узла pc2 ввести команду:

```
# ping 192.168.1.2 -c7
```

Утилита *ping* в данном случае отправляет на pc2 7 эхо-запросов (список возможных параметров утилиты отображается по команде *man ping)*. Если пакеты возвращаются без потерь, аналогично проверить доступность остальных компьютеров.

15. После проверки завершить эксперимент: Experiment ➡ Terminate.

16. Сохранить созданную топологию: File ➡ Save as.

4 Содержание отчета

- титульный лист;
- задачи лабораторной работы;
- скриншот построенной топологии с подписью;
- скриншоты выполнения команды *ping*.

5 Контрольные вопросы

1) Какие инструменты содержит окно программы «IMUNES»?
2) В чем отличия режима редактирования от режима выполнения? Как переключаться между ними?
3) Как добавлять сетевые элементы на рабочую область? Как устанавливать соединение между элементами?
4) Каким образом задается адрес узла?
5) Для чего нужна утилита *ping*?

Лабораторная работа №2. Моделирование сети со статической маршрутизацией

1 Цель работы

Ознакомление с функциями маршрутизаторов. Формирование навыка организации статических маршрутов и занесения их в таблицы маршрутизации сетевых устрои□ств.

2 Задачи

Построить сетевую топологию согласно приведенной схеме подключения. Задать адресацию узлов и сетевых интерфейсов в соответствии с вариантом. Настроить роутеры в режиме статической маршрутизации таким образом, чтобы обеспечить возможность прохождение сигнала между всеми узлами. Воспользовавшись утилитой *traceroute*, проверить маршруты следования данных в сети.

3 Теоретический материал

Статическая маршрутизация — вид маршрутизации, при котором информация о маршрутах заносится в таблицы маршрутизации каждого роутера вручную администратором сети. Маршрутизация происходит без использования протоколов маршрутизации. При изменениях в топологии сети, системный администратор каждый раз должен корректировать статические маршруты. С другой стороны, данный способ является наиболее стабильным и требующим минимума аппаратных ресурсов.

Статическая маршрутизация применяется в организации работы вычислительных сетей небольшого размера, в силу легкости конфигурации и отсутствия дополнительной нагрузки на сеть в виде широковещательного служебного трафика. Также статическая маршрутизация используется на компьютерах внутри сети. В таком случае обычно задается маршрут шлюза по

умолчанию — шлюза, на который отправляется пакет тогда, когда маршрут к сети назначения не задан явным образом в таблице маршрутизации [3].

При задании статического маршрута указывается:

- адрес сети, в которую направлен трафик, и маска сети;
- адрес шлюза (узла), который отвечает за дальнейшую маршрутизацию или подключен к сети назначения напрямую.

Пример записи статического маршрута: 172.23.34.0/24 10.57.0.9

Утилита *traceroute* предназначена для определения маршрутов следования данных в сетях TCP/IP. Она основана на использовании протоколов ICMP и UDP, а также механизма TTL.

Для фиксации промежуточных маршрутизаторов *traceroute* отправляет серию UDP датаграмм целевому узлу, при этом каждыи раз увеличивая на 1 значение поля TTL в заголовке IP-пакета. Это поле указывает максимальное количество маршрутизаторов, которое может быть прои дено пакетом. Первая серия пакетов отправляется с TTL, равным 1, и поэтому первыи же маршрутизатор возвращает обратно сообщение ICMP «время жизни истекло», указывающее на невозможность доставки данных. Traceroute сохраняет адрес маршрутизатора, а также время между отправкои пакета и получением ответа. Затем утилита повторяет отправку серии пакетов, но уже с TTL, равным 2, что позволяет первому маршрутизатору пропустить их дальше. Процесс повторяется до тех пор, пока при определенном значении TTL пакет не достигнет целевого узла. При получении ответа от этого узла процесс трассировки считается завершенным, результат выводится на экран [15].

4 Порядок выполнения лабораторной работы

1. Запустить программный комплекс «IMUNES», выполнив в терминале команду *imunes* либо дважды щелкнув ЛКМ по ярлыку программы.

2. В правый верхний угол рабочей области добавить фигуру произвольного цвета с помощью инструмента Rectangle. Размер рабочей области можно изменить: Canvas ➡ Resize.

3. Используя инструмент Text, добавить на фигуру подпись с ФИО студента и номером группы.

4. Используя соответствующие инструменты, построить сеть из 5 маршрутизаторов, 1 коммутатора, 5 ПК и 2 хостов в соответствии с топологией, изображенной на рисунке 24.

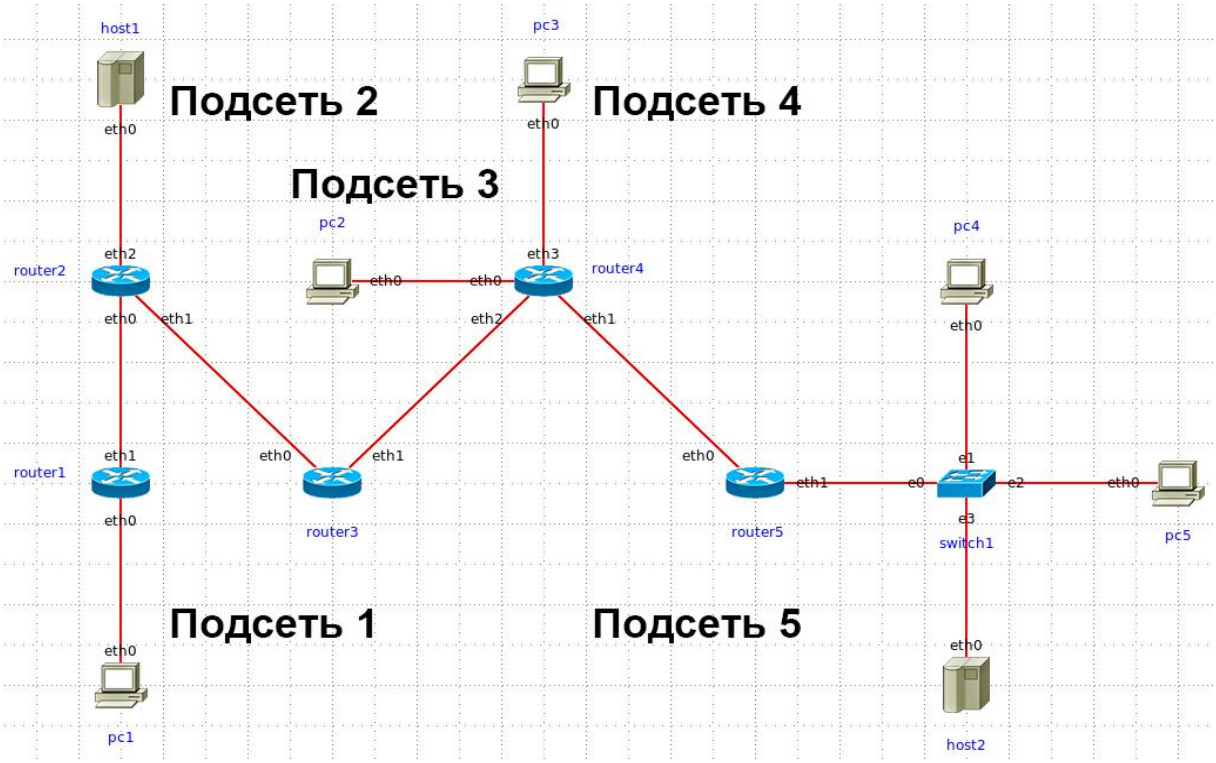

Рисунок 25. Топология сети для лабораторной работы №2

5. По умолчанию IP-адреса узлов и сетевых интерфейсов задаются автоматически из пула адресов, который редактируется в меню: Tools ➡ IPv4 address pool. Студентам необходимо задать IP-адреса вручную согласно варианту, который соответствует последней

цифре номера студенческого билета. Для подсетей между роутерами использовать сети с маской /30 из диапазона 10.10.0.0/24, а для остальных подсетей диапазоны из таблицы 3.

Таблица 3. Исходные данные для лабораторной работы №2

№	Подсеть 1	Подсеть 2	Подсеть 3	Подсеть 4	Подсеть 5
0	192.168.64.0/24	172.24.7.0/16	192.168.39.0/24	10.0.37.0/8	172.30.0.0/8
1	10.13.0.0/16	10.72.0.0/16	192.168.0.0/16	172.21.34.0/24	10.56.0.0/16
2	10.57.0.0/16	172.21.11.0/24	192.168.78.0/24	192.168.9.0/24	172.23.73.0/24
3	172.21.4.0/24	172.19.34.0/24	10.52.0.0/24	192.168.41.0/24	10.15.0.0/24
4	192.168.52.0/24	192.168.53.0/24	172.25.53.0/24	192.168.0.0/16	172.29.34.0/24
5	172.27.57.0/16	172.31.0.0/24	192.168.67.0/24	172.30.0.0/8	10.73.0.0/16
6	10.29.0.0/16	172.27.0.0/16	172.21.1.0/24	10.81.0.0/16	192.168.6.0/24
7	10.3.0.0/12	172.27.17.0/24	10.14.0.0/16	192.168.0.0/16	192.168.8.0/24
8	172.0.0.0/24	172.17.7.0/24	172.7.0.0/12	192.168.24.0/24	172.30.0.0/24
9	10.40.0.0/16	172.29.30.0/24	172.23.22.0/24	172.27.0.0/16	192.168.4.0/24

6. Установить на router1, router2, router3, router4, router5 статический режим маршрутизации. Для этого дважды щелкнуть ЛКМ по иконке роутера и в открывшемся окне конфигурации выбрать пункт static.

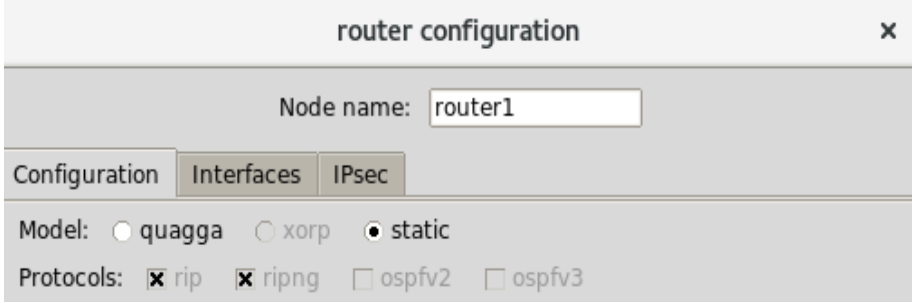

Рисунок 26. Окно конфигурации маршрутизатора

7. Убедиться в том, что в поле Static routes окна конфигурации ПК и хостов указаны корректные IP-адреса шлюзов по умолчанию (ближайших роутеров).
8. Задать роутерам статические маршруты таким образом, чтобы все компьютеры и хосты в моделируемой топологии могли обмениваться данными друг с другом. Это можно сделать в поле Static routes окна конфигурации маршрутизаторов.

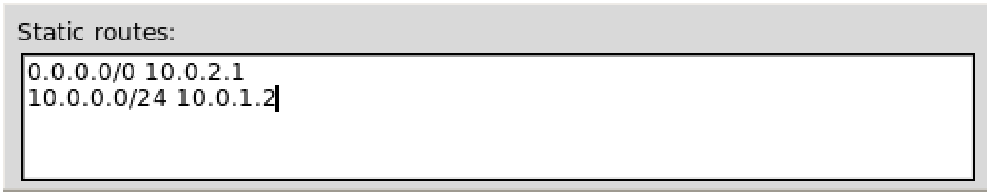

Рисунок 27. Запись адреса шлюза по умолчанию и статического маршрута в настройках маршрутизатора

Также добавление статического маршрута возможно по команде *ip route add*, которая прописывается в консоли сетевого устройства, когда IMUNES работает в режиме выполнения (прочие опции данной команды: *ip route help)*. Например:

```
# ip route add default via 10.0.2.1
# ip route add 10.0.0.0/24 10.0.1.2
```

9. Сохранить настроенную сеть: File ➞ Save as.
10. Перевести IMUNES в режим выполнения: Experiment ➞ Execute.
11. С помощью утилиты *traceroute* проверить маршрут следования данных от компьютера pc1 до хоста host2. В консоли pc1 выполнить:

```
# traceroute IP-адрес вашего host2
```

12. Таким же образом проверить маршрут следования данных от компьютера pc3 до хоста host1. В консоли pc3 выполнить:

```
# traceroute IP-адрес вашего host1
```

5 Содержание отчета

- титульный лист;
- задачи лабораторной работы;
- скриншот построенной топологии с подписью;
- скриншоты выполнения команды *traceroute*;
- выводы о проделанной работе.

6 Контрольные вопросы

1) В каких случаях применяется статическая маршрутизация?
2) Как добавлять или удалять статические маршруты в IMUNES?
3) Что такое шлюз по умолчанию?
4) Принцип работы утилиты *traceroute.*

Лабораторная работа №3. Изучение работы протокола ARP

1 Цель работы

Изучение особенностей работы протокола ARP в IP-сети. Формирование навыка использования утилит командной строки, позволяющих проконтролировать работу протокола ARP.

2 Задачи

Построить сетевую топологию согласно приведенной схеме подключения. Настроить роутер в режиме статической маршрутизации таким образом, чтобы обеспечить возможность прохождение сигнала между всеми узлами сети. Проверить работу протокола ARP в построенной сети.

3 Теоретический материал

ARP (Address Resolution Protocol) — протокол в компьютерных сетях, который служит для определения MAC-адреса сетевого устрои□ства по известному IP-адресу. Описание протокола было опубликовано в ноябре 1982 г. в RFC 826. ARP спроектирован для случая передачи IP-пакетов через сегмент Ethernet, однако принцип данного протокола используется и для сетеи□ других типов. В семеи□стве IPv6 протокола ARP нет, его функции выполняет ICMPv6. Существует 2 типы сообщении□ ARP: запрос ARP (ARP-request) и ответ ARP (ARP-reply) [16].

Принцип работы протокола:

1) В начале передачи данных от узла A к узлу B первый узел должен определить MAC-адрес второго. Для этого узел A формирует запрос ARP, указывая в нем известныи□ IP-адрес B, и рассылает запрос широковещательно по протоколу канального уровня.

2) Все узлы локальной сети получают ARP и сравнивают указанный в нем IP-адрес с собственным.
3) Определив, что IP-адрес в запросе совпадает с IP-адресом узла B, он формирует ARP-ответ, в котором указывает IP-адрес и локальный MAC-адрес. Ответ отправляется напрямую, так как ARP-запрос от A указал MAC-адрес.
4) Получив ARP-ответ, передающий узел A записывает соответствие MAC-адреса B его IP-адресу в специальную ARP-таблицу. Теперь, пока запись хранится в ARP-таблице, данные могут передаваться от A к B. Для осуществления передачи от B к A требуется выполнить обратную процедуру.
5) Когда данные от A к B все еще передаются, но время жизни записи в ARP-таблице истекло, узел A должен обновить ее. Для этого он посылает повторный ARP-запрос, который отправляется с целью извлечь MAC-адрес узла B. Если ответ на повторный запрос не приходит, процедура разрешения адреса повторяется с самого начала (посредством широковещательного ARP-запроса).

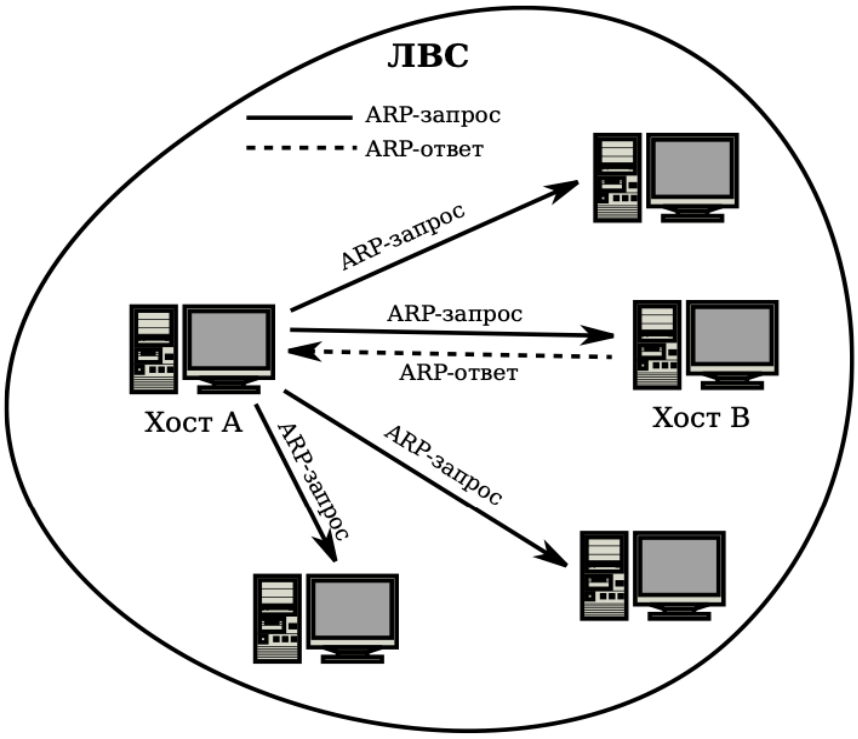

Рисунок 28. Схема работы протокола ARP

Самопроизвольныи ARP (gratuitous ARP) — такое поведение ARP, при котором ARP-ответ присылается, когда в этом нет необходимости с точки зрения получателя. Самопроизвольныи ARP-ответ — это пакет ARP, присланныи без запроса. Он применяется для определения конфликтов IP-адресов в сети: как только станция получает адрес по DHCP или адрес присваивается вручную, рассылается ARP-ответ gratuitous ARP. Самопроизвольный ARP является особенно небезопасным, поскольку с его помощью можно уверить удаленныи узел в том, что MAC-адрес какои-либо системы в его сети изменился, и указать вместо него ложный адрес [16].

4 Порядок выполнения лабораторной работы

1. Запустить программный комплекс «IMUNES», выполнив в терминале команду *imunes* либо дважды щелкнув ЛКМ по ярлыку программы.
2. В правый верхний угол рабочей области добавить фигуру произвольного цвета с помощью инструмента Rectangle Размер рабочей области можно изменить: Canvas ➞ Resize.
3. Используя инструмент Text, добавить на фигуру подпись с ФИО студента и номером группы.
4. Используя соответствующие инструменты, построить сеть из 5 ПК, 2 коммутаторов и 1 маршрутизатора в соответствии с топологией, изображенной на рисунке 30. Предварительно задать пул IPv4-адресов: Tools ➞ IPv4 address pool ➞ 172.25.0.0/16 ➞ Apply.

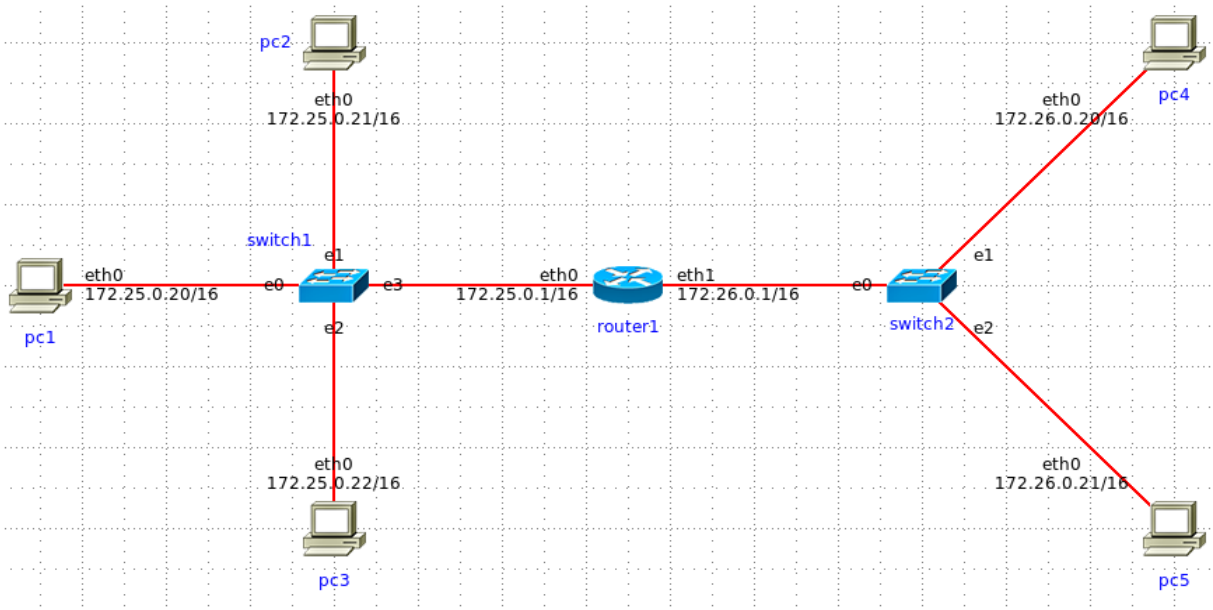

Рисунок 29. Топология сети для лабораторной работы №3

5. Сохранить созданную топологию: File ➞ Save as.

6. Перевести IMUNES в режим выполнения: Experiment ➞ Execute.

7. Открыть консоль компьютера pc1 и выполнить команду *arp*. Отсутствие вывода результатов означает, что ARP-таблица данного узла пуста.

8. На компьютере pc3 запустить анализатор трафика Wireshark: щелкнуть ПКМ по изображению компьютера, навести курсор на соответствующий пункт, щелкнуть по строке с интерфейсом eth0.

9. В открывшемся окне ввести команду *arp*. Теперь Wireshark будет отображать трафик протокола ARP на узле pc3.

10. В консоли компьютера pc1 с помощью утилиты *ping* отправить 1 эхо-запрос на узел pc3:

```
# ping 175.25.0.22 -c1
```

11. В окне программы Wireshark зафиксировать ARP сообщения, полученные и отправленные pc3 в процессе работы утилиты *ping*. Перед отправкой эхо-запроса pc1 широковещательно узнает, какой MAC-адрес соответствует IP-адресу pc3, и получает от pc3 ответ. Затем происходит

передача сообщения ICMP на узел pc3 и их последующая отправка в обратном направлении с аналогичными предварительными действиями.

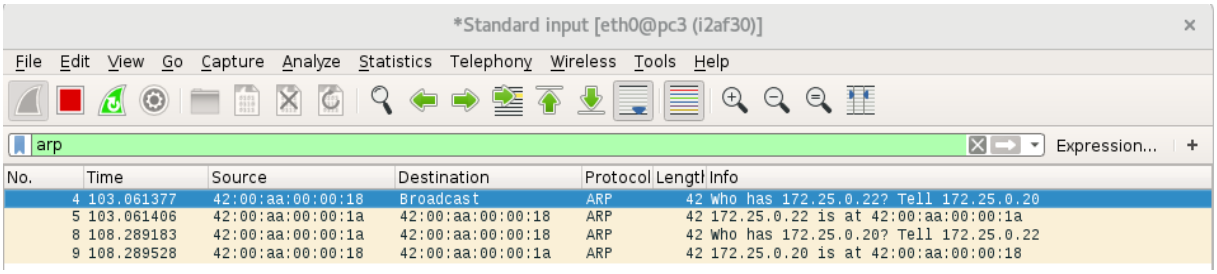

Рисунок 30. Скриншот окна Wireshark с отображением ARP пакетов

12. Заново ввести команду *arp* в консоль компьютера pc1 и убедиться в том, что в ARP-таблице появилась запись о соответствии IP-адреса pc3 его MAC-адресу (для pc3 аналогично).

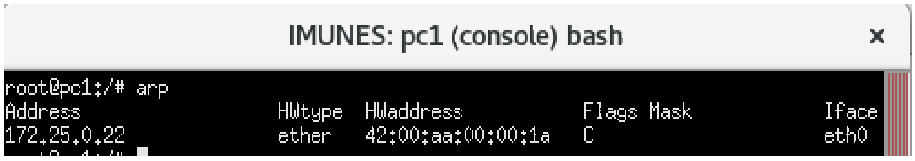

Рисунок 31. Скриншот консоли pc3 с ARP-таблицей

13. Повторить описанные выше действия для компьютеров pc4 и pc5, начиная с пункта 6. Сохранить скриншоты программы Wireshark и консолей с ARP-таблицами.

5 Содержание отчета

- титульный лист;
- задачи лабораторной работы;
- скриншот построенной топологии с подписью;
- скриншоты программы Wireshark и ARP-таблиц;
- выводы о проделанной работе.

6 Контрольные вопросы

1) Назначение протокола ARP.
2) Что такое ARP-запрос и ARP-ответ?
3) Самопроизвольный ARP.

Лабораторная работа №4. Динамическая маршрутизация по протоколам RIP и OSPF

1 Цель работы

Ознакомление с механизмом динамическои маршрутизации по протоколам RIP и OSPF.

2 Задачи

Построить сетевую топологию согласно приведенной схеме подключения. Задать адресацию узлов и сетевых интерфейсов в соответствии с вариантом. Сравнить работу роутеров в режиме динамической маршрутизации по протоколам RIP и OSPF с помощью анализатора трафика Wireshark, утилит *arp* и *traceroute*.

3 Теоретический материал

Протоколы динамической маршрутизации служат для автоматизации процесса составления маршрутных таблиц. Принцип их использования состоит в том, что роутеры рассылают по сети определенную информацию из своей таблицы маршрутизации и корректируют ее на основе полученных ответов. Данный подход оптимизирует администрирование сетей, в которых могут происходить частые или непрогнозируемые изменения. При использовании динамической маршрутизации маршруты также можно прописывать и вручную.

Выделяют две группы протоколов маршрутизации. В протоколах типа «вектор-расстояние» каждый роутер рассылает список адресов доступных ему сетей («векторов»), с каждым из которых связан параметр «расстояния» (например, количество маршрутизаторов до сети). Основным представителем данной группы является RIP (Routing Information Protocol) — протокол маршрутной информации. Максимальное значение расстояния для протокола RIP равно 15, значение 16 трактуется как «сеть недостижима». Поэтому в

масштабных сетях данный протокол не применяется. Также недостатком первой версии протокола RIP является то, что оценка расстояния происходит только с учетом числа переходов. Протокол RIP не учитывает реальную производительность каналов связи, что может оказаться неэффективным в гетерогенных сетях с использованием различных устройств и технологий. Частично данные проблемы решаются во второй версии: RIP2. Для маршрутизации IPv6-адресов используется протокол нового поколения: RIPng.

Протоколы типа «состояние канала» работают иначе. Маршрутизаторы обмениваются между собой топологической информацией о связях в сети: какие маршрутизаторы с какими сетями соединены. В результате каждый роутер имеет целостное представление о структуре топологии, на основе которого вычисляет собственную оптимальную таблицу маршрутизации. Протоколом этой группы является более новый протокол OSPF (Open Shortest Path First) — открой кратчайший путь первым. При работе по данному протоколу маршрутизаторы связывают с каждой сетью метрику — значение, характеризующее «качество» канала. Например, для сетей Ethernet со скоростью обмена 100 Мбит/с используется значение 1, а для коммутируемых соединений 56 Кбит/с – значение 1785. Таким образом, маршрутизаторы OSPF (в отличие от RIP, где все каналы равнозначны) учитывают реальную пропускную способность и выявляют эффективные маршруты. Важной отличием протокола OSPF является использование групповой, а не широковещательной рассылки.

Указанные особенности позволяют использовать OSPF в больших сетях. Однако такое использование может породить проблему большого количества циркулирующей в сети маршрутной информации и увеличения таблиц маршрутизации. Следовательно, увеличивается стоимость сетевого оборудования [17].

4 Порядок выполнения лабораторной работы

1. Запустить программный комплекс «IMUNES», выполнив в терминале команду *imunes* либо дважды щелкнув ЛКМ по ярлыку программы.
2. В правый верхний угол рабочей области добавить фигуру произвольного цвета с помощью инструмента Rectangle. Размер рабочей области можно изменить: Canvas ➞ Resize.
3. Используя инструмент Text, добавить на фигуру подпись с ФИО студента и номером группы.
4. Используя соответствующие инструменты, построить сеть из 7 ПК, 2 хостов, 3 коммутаторов и 6 маршрутизаторов в соответствии с топологией, изображенной на рисунке 33.

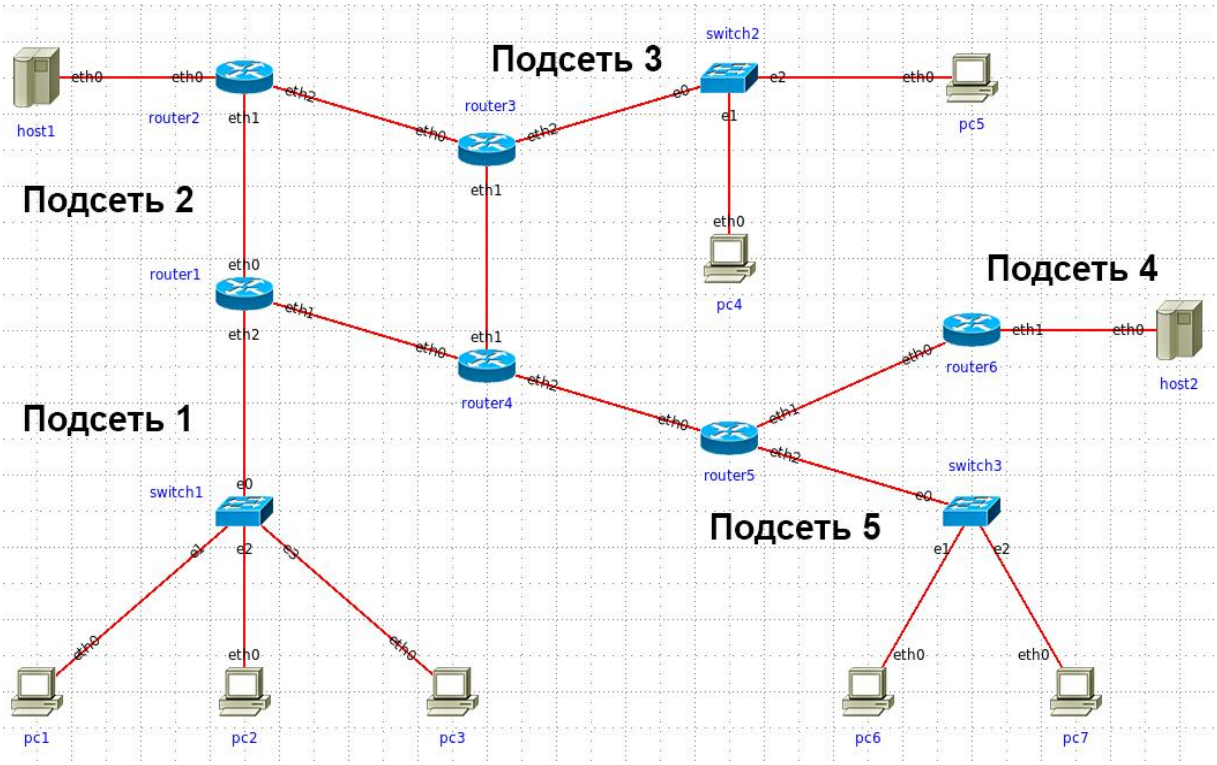

Рисунок 32. Топология сети для лабораторной работы №4

5. По умолчанию IP-адреса узлов и сетевых интерфейсов задаются автоматически из пула адресов, который редактируется в меню: Tools ➞ IPv4 address pool. Студентам необходимо задать IP-адреса

вручную согласно варианту, который соответствует последней цифре номера студенческого билета. Для подсетей между роутерами использовать подсети с маской /30 из диапазона 10.10.0.0/24, а для остальных подсетей диапазоны из таблицы 3 (лабораторная работа №2).

6. Сохранить настроенную топологию: File ➞ Save as.
7. Установить на router1, router2, router3, router4, router5, router6 протокол динамической маршрутизации RIP. Для этого дважды щелкнуть ЛКМ по иконке роутера и в открывшемся окне конфигурации выбрать только пункт rip.
8. Перевести IMUNES в режим выполнения: Experiment ➞ Execute.
9. Произвольно выбрать 3 маршрутизатора. На каждом из них с помощью Wireshark проследить за прохождением пакетов на интерфейсе, через который выбранный роутер соединен с другим роутером. Сохранить 3 скриншота окна Wireshark. Закрыть Wireshark.
10. На этих же роутерах в консоли просмотреть ARP-таблицы. Сохранить 3 скриншота.
11. С помощью утилиты *traceroute* проверить маршрут следования данных от компьютера pc4 до pc7. Затем от хоста host2 до host1 и от компьютера pc1 до pc5. Сохранить 3 скриншота результатов выполнения *taceroute.*
12. Перевести IMUNES в режим редактирования: Experiment ➞ Terminate.
13. Установить на router1, router2, router3, router4, router5, router6 протокол динамической маршрутизации OSPF. Для этого дважды щелкнуть ЛКМ

по иконке роутера и в открывшемся окне конфигурации выбрать только пункт ospfv2.

14. Перевести IMUNES в режим выполнения: Experiment ➞ Execute.
15. Повторить действия из пунктов 9-11 с 3 ранее выбранными маршрутизаторами.
16. Сравнить скриншоты Wireshark, ARP-таблиц и результатов выполнения *traceroute* при динамической маршрутизации по протоколам RIP и OSPF.
17. Составить выводы на основании проанализированных результатов.

5 Содержание отчета

- титульный лист;
- задачи лабораторной работы;
- скриншот построенной топологии с подписью;
- 9 скриншотов программы Wireshark, ARP-таблиц и результатов выполнения *traceroute* при динамической маршрутизации по протоколу RIP;
- 9 скриншотов программы Wireshark, ARP-таблиц и результатов выполнения *traceroute* при динамической маршрутизации по протоколу OSPF;
- выводы о проделанной работе.

6 Контрольные вопросы

1) Протокол RIP.
2) В чем особенность протокола RIPng?
3) Протокол OSPF.

Лабораторная работа №5. Ознакомление с утилитой *netcat*

1 Цель работы

Ознакомление с некоторыми возможностями сетевой утилиты *netcat* на базе топологии, построенной в IMUNES.

2 Задачи

Создать модель сети, согласно приведенной схеме подключения. С помощью утилиты *netcat* установить соединение между двумя компьютерами, осуществить передачу текстового файла между ними. Реализовать выполнение команд на удаленном ПК.

3 Теоретический материал

Программа Netcat — это сетевая утилита командной строки общего назначения для чтения, записи, перенаправления и шифрования данных в сети. Команда для Netcat выглядит как:

```
nc [options] host ports
```

Где *host* — имя хоста или его IP-адрес, а *ports* — это номер порта (несколько номеров) или диапазон номеров портов. Описание опций приведено ниже.

-d. Доступна только в Windows. Позволяет запустить утилиту в режиме прослушивания, не открывая окно MS-DOS.

-e <*command*>. Если Netcat скомпилирован с опцией GAPING_SECURITY_HOLE, программа может выполнять команду <*command*> всякий раз, когда с прослушиваемым портом устанавливается соединение.

-i <*seconds*>. Интервал задержки между пересылками данных. Если через конвейер Netcat проходит файл, то программа ждет <*seconds*> секунд перед тем, как передать следующую строку. Если Netcat применяется для управления

несколькими портами на одном хосте, утилита ждет <*second*> секунд перед тем, как соединится со следующим портом из перечисленных в строке.

-g <route-list>. Позволяет передавать трафик через определенные IP-адреса, которые используются для исходящего адреса.

-l. Активирует режим прослушивания Netcat. Опция используется совместно с *-p*, чтобы привязать Netcat к определенному TCP-порту и ожидать входящих соединений.

-n. При использовании данной опции не следует указывать имена хостов в качестве аргументов.

-o <hexfile>. Обеспечивает создание шестнадцатеричного дампа данных и сохранение его в файле hexfile.

-p <port>. Опция позволяет задать локальный номер порта, который следует использовать Netcat. Если опция не определена для исходящего соединения, Netcat будет использовать заданный системой порт.

-r. Локальный и удаленный порты выбираются случайным образом.

-s. Определяет исходящий IP-адрес, который Netcat использует для установки соединения.

-t. Скомпилированный с опцией TELNET, Netcat может поддерживать взаимодействие с telnet-сервером в соответствии с установленными соглашениями.

-u. Опция сообщает программе о необходимости использовать UDP-протокол вместо TCP, работая как в режиме прослушивания, так и в режиме клиента.

-v. Определяет степень подробности вывода программы. Чем больше раз используется, тем подробнее вывод.

-w <seconds> определяет промежуток времени, в течение которого Netcat ждет соединения.

-z. Опция сообщает Netcat о необходимости послать достаточно данных для поиска открытых портов в заданном диапазоне значений [18].

4 Порядок выполнения лабораторной работы

1. Запустить программный комплекс «IMUNES», выполнив в терминале команду *imunes* либо дважды щелкнув ЛКМ по ярлыку программы.
2. В правый верхний угол рабочей области добавить фигуру произвольного цвета с помощью инструмента Rectangle Размер рабочей области можно изменить: Canvas ➡ Resize.
3. Используя инструмент Text, добавить на фигуру подпись с ФИО студента и номером группы.
4. Используя соответствующие инструменты, построить сеть из 4 ПК, 2 коммутаторов и 2 маршрутизаторов в соответствии с топологией, изображенной на рисунке 34.

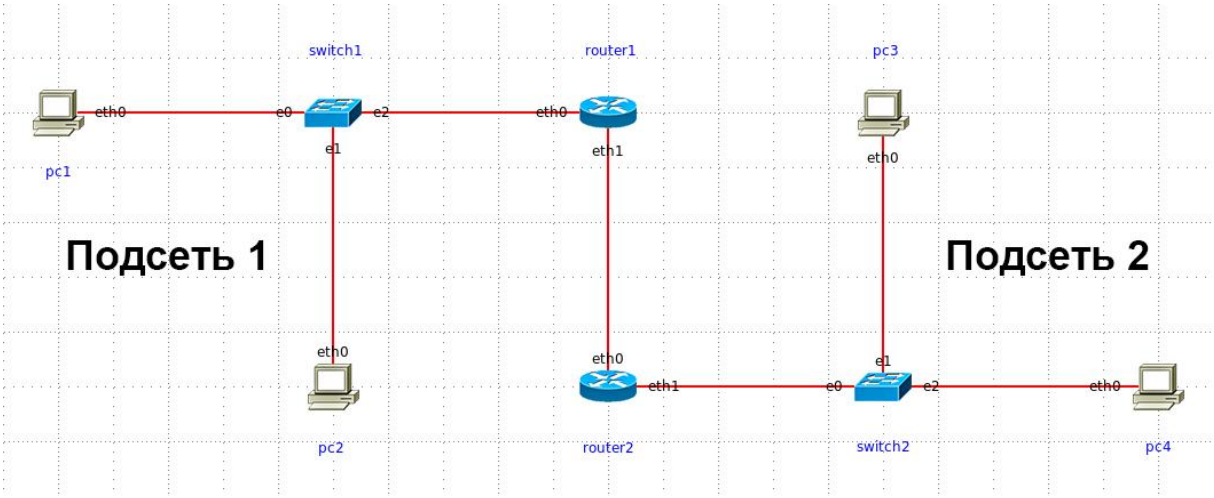

Рисунок 33. Топология сети для лабораторной работы №5

5. По умолчанию IP-адреса узлов и сетевых интерфейсов задаются автоматически из пула адресов, который редактируется в меню: Tools ➡ IPv4 address pool. Студентам необходимо задать IP-адреса вручную согласно варианту, который соответствует последней

цифре номера студенческого билета. Для подсетей между роутерами использовать подсети с маской /30 из диапазона 10.10.0.0/24, а для остальных подсетей диапазоны из таблицы 3 (лабораторная работа №2).

6. Сохранить топологию сети: File ➞ Save as.
7. Перевести IMUNES в режим выполнения: Experiment ➞ Execute.
8. Запустить на компьютере pc1 анализатор трафика Wireshark и задать фильтр по протоколу TCP.
9. В консоли pc1 запустить утилиту *netcat* в пассивном режиме (флаг -l), указав для прослушивания номер порта более 20000:

```
# nc -l -p номер порта
```

10. В консоли pc4 запустить утилиту *netcat*, установив соединение с pc1:

```
# nc IP-адрес pc1 номер порта
```

11. После установления соединения, текст, набранныи□ в консоли, по нажатию клавиши Enter будет также отображаться в консоли другого компьютера. Передать несколько фраз и сделать скриншоты консолей.
12. Завершить соединение с помощью комбинаций клавиш Ctrl + C. Остановить захват пакетов в Wireshark и сделать скриншоты.
13. Запустить на компьютере pc2 анализатор трафика Wireshark и задать фильтр по протоколу TCP.
14. Используя текстовый редактор *nano*, создать на pc2 фаи□л, содержащий название дисциплины и текущую дату. Выполнить команду:

```
# nano имяфайла.txt
```

15. В консоли pc2 запустить утилиту *netcat* в пассивном режиме, указав для прослушивания номер порта более 20000 и передать в нее созданныи фаи л при помощи утилиты *cat* и механизма конвеи ера:

```
# cat имя файла | nc -l -p номер порта
```

16. В консоли компьютера pc3 запустить утилиту *netcat*, установив соединение с pc2, и перенаправить ее вывод в текстовыи фаи л:

```
# nc IP-адрес номер порта > имя файла
```

17. Убедиться в том, что фаи л передан полностью. Завершить соединение с помощью комбинаций клавиш Ctrl + C. Остановить захват пакетов в Wireshark и сделать скриншоты.
18. Запустить на компьютере pc3 анализатор трафика Wireshark и задать фильтр по протоколу TCP.
19. В консоли pc3 запустить утилиту *netcat* в пассивном режиме, указав для прослушивания номер порта более 20000 и вызвав в качестве получателя/отправителя данных командныи процессор Bash:

```
# nc -l -p номер порта -e /bin/bash
```

20. В консоли pc1 запустить утилиту *netcat*, установив соединение с pc3:

```
# nc IP-адрес номер порта
```

21. После установления соединения, команды, набранные в консоли pc1 по нажатию клавиши Enter будут пересылаться на pc1 и передаваться в командныи процессор. Проверить это, последовательно используя команды:

```
# /sbin/ifconfig
```

```
# uname -a
```

22. Завершить соединение с помощью комбинаций клавиш Ctrl + C. Остановить захват пакетов в Wireshark. Сделать скриншоты консолей и программы Wireshark.

5 Содержание отчета

- титульный лист;
- задачи лабораторной работы;
- скриншот построенной топологии с подписью;
- команды установления соединения;
- скриншоты программы Wireshark;
- выводы о проделанной работе.

6 Контрольные вопросы

1) Назначение утилиты *netcat*.
2) Общий вид команды утилиты *netcat*.
3) Для чего служат опции *-l* и *-p* ?

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Компьютерные сети передачи данных [Электронный ресурс] // Электронное пособие СПбГУТ URL: http://opds.spbsut.ru/ecourse/2019-kspd/index.html (дата обращения 12.04.2020)
2. Сергеев А. Н. Основы локальных компьютерных сетей: Учебное пособие. / А. Н. Сергеев — СПб.: Издательство «Лань», 2016. — 184 с.: ил. — (Учебники для вузов. Специальная литература)
3. Олифер В. Г. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 5-е изд. / В. Г. Олифер, Н. А. Олифер — СПб.: Питер, 2016. — 992 с.: ил. — (Серия «Учебник для вузов»)
4. Сетевая модель OSI [Электронный ресурс] // Википедия: общедоступная многоязычная интернет-энциклопедия URL: https://ru.wikipedia.org/wiki/Сетевая_модель_OSI (дата обращения 19.04.2020)
5. Капустин Д. А. Информационно-вычислительные сети: Учебное пособие / Д. А. Капустин, В. Е. Дементьев — Ульяновск: УлГТУ, 2011. — 141 с.
6. Технология Ethernet (802.3) & метод доступа CSMA/CD [Электронный ресурс] // ColumbianX: информационный портал URL: https://columbianx.wordpress.com/2011/01/08/технология-ethernet-802-3-метод-доступа-csmacd/ (дата обращения 21.04.2020)
7. Чеппел Л. А. TCP/IP. Учебный курс: Пер. с англ. / Л. А. Чеппел, Э. Титтел — СПб.: БХВ-Петербург, 2003. — 976 с.: ил.
8. Имитационное моделирование компьютерных сетей [Электронный ресурс] // Educationspb: информационно-образовательный портал URL: http://www.educationspb.ru/komp/37520.html (дата обращения 29.04.2020)
9. Network simulation or emulation [Электронный ресурс] // Networkworld: информационный портал URL: https://www.networkworld.com/article/3227076/network-simulation-or-emulation.html (дата обращения 24.04.2020)

10. Open-Source Network Simulators [Электронный ресурс] // Open-Source Routing and Network Simulation: персональный блог инженера телекоммуникаций URL: http://www.brianlinkletter.com/open-source-network-simulators/ (дата обращения 05.05.2020)
11. Puljiz Z., Mikuc M. IMUNES Based Distributed Network Emulator [Электронный ресурс] // Hrvatska znanstvena bibliografija: хорватская научная библиография [сайт] URL: https://bib.irb.hr/datoteka/301604.PID283046.pdf (дата обращения 06.05.2020)
12. Salopek D., Vasić V., Mikuc M. Security research and learning environment based on scalable network emulation [Электронный ресурс] // Tehnic□ki vjesnik: технический журнал 2017, с. 535-544. URL: https://pdfs.semanticscholar.org/618b/6a5845169b557aadf392f05e7cafa3313396.pdf (дата обращения 06.05.2020)
13. IMUNES - an Integrated Multiprotocol Network Emulator/Simulator [Электронный ресурс] // GitHub: веб-сервис для хостинга IT-проектов URL: https://github.com/imunes (дата обращения 05.05.2020)
14. IMUNES Manual [Электронный ресурс] // Integrated Multiprotocol Network Emulator/Simulator: [сайт] URL: http://imunes.net/dl/imunes_user_guide.pdf (дата обращения 03.05.2020)
15. Traceroute [Электронный ресурс] // Википедия: общедоступная многоязычная интернет-энциклопедия URL: https://ru.wikipedia.org/wiki/Traceroute (дата обращения 28.05.2020)
16. Владимиров С.С. Компьютерные сети передачи данных : лабораторны□ практикум / С.С. Владимиров — СПб.: СПбГУТ, 2016.—24с.
17. Динамическая маршрутизация. Протокол RIP. Протокол OSPF [Электронный ресурс] // Tesenmir.net: персональный сайт инженера телекоммуникаций URL: https://pub.tesenmir.net/?p=524 (дата обращения 04.06.2020)

18. Инструментальные средства обеспечения безопасности. Лекция 1: NETCAT и CRYPTCAT [Электронный ресурс] // ИНТУИТ: национальный открытый университет [сайт] URL: https://www.intuit.ru/studies/courses/1053/150/lecture/4155?page=2 (дата обращения 04.06.2020)