

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Локтионова Оксана Геннадьевна  
Должность: проректор по учебной работе  
Дата подписания: 08.10.2026 18:40:32  
Уникальный программный ключ:  
0b817ca911e6668abb13a5d426d39a51c11eabbf73e947df4a4851fda56d089

## МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Юго-Западный государственный университет»  
(ЮЗГУ)

Кафедра информационной безопасности



## МАСКА ПОДСЕТИ

Методические указания по выполнению лабораторной и практической  
работы для студентов укрупненной группы специальностей и  
направлений подготовки 10.00.00.

Курск 2024

УДК 004

Составители: Е.М. Терещенко, В.В. Чуйкова

Рецензент

Доктор физико-математических наук, профессор кафедры  
Информационная безопасность *В.П. Добрица*

**Маска подсети:** методические указания к выполнению лабораторной и практической работ по дисциплинам / Юго-Зап. гос. Ун-т; сост. Е.М. Терещенко, В.В. Чуйкова. Курск, 2024, 14 с.: ил. 3.; Библиогр.: с. 14.

Содержат сведения по вопросам использования маски подсети для адресации сообщений. Указывается порядок выполнения лабораторных и практических работ, правила оформления, содержание отчета.

Методические указания по выполнению лабораторной и практической работ по дисциплинам «Безопасность систем и сетей передачи данных», «Сети и системы передачи информации», «Сети и системы передачи информации (специальные разделы)», «Администрирование вычислительных сетей», «Администрирование защищенных телекоммуникационных систем» для студентов укрупненной группы специальностей и направлений подготовки 10.00.00.

Текст печатается в авторской редакции

Подписано в печать 16.05.2024. Формат 60х84 1/16. Усл. печ. л. 0,81. Уч. –изд.л. 0,74. Тираж 30 экз. Заказ . Бесплатно. 375

Юго-Западный государственный университет.  
305040, г. Курск, ул. 50 лет Октября, 94.

## СОДЕРЖАНИЕ

|                                       |    |
|---------------------------------------|----|
| Введение .....                        | 4  |
| Цель работы .....                     | 4  |
| Порядок выполнения работы.....        | 4  |
| Содержание отчета .....               | 5  |
| Теоретическая часть .....             | 6  |
| Выполнение работы.....                | 12 |
| Задание.....                          | 15 |
| Контрольные вопросы .....             | 15 |
| Список информационных источников..... | 15 |

## **ВВЕДЕНИЕ**

IP-адрес представляет собой 32-разрядный номер, который уникально идентифицирует узел (компьютер или устройство, например, принтер или маршрутизатор) в сети TCP/IP.

IP-адреса обычно представлены в виде 4-х разрядов, разделенных точками, например 192.168.123.132. Чтобы глобальная сеть TCP/IP работала эффективно как совокупность сетей, маршрутизаторы, обеспечивающие обмен пакетами данных между сетями, не знают точного расположения узла, для которого предназначен пакет. Маршрутизаторы знают только, к какой сети принадлежит узел, и используют сведения, хранящиеся в таблицах маршрутизации, чтобы доставить пакет в сеть узла назначения. Для осуществления этого процесса IP-адрес состоит из двух частей. Первая часть IP-адреса обозначает адрес сети, последняя часть – адрес узла.

В терминологии сетей TCP/IP маской подсети или маской сети называется битовая маска, определяющая, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети.

## **ЦЕЛЬ РАБОТЫ**

Цель лабораторной работы – изучить назначение IP-адреса в TCP/IP узлах.

## **ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ**

1. Получить задание
2. Изучить теоретическую часть
3. Выполнить практическое задание
4. Написать вывод

## **СОДЕРЖАНИЕ ОТЧЕТА**

1. Титульный лист
2. Задание в соответствии с вариантом
3. Выполненное задание
4. Вывод

## ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Подсеть (subnet) – это физический сегмент ТСР/IP сети, в котором используются IP-адреса с общим идентификатором сети.

Для того чтобы разделить сеть на несколько подсетей, необходимо использовать различные идентификаторы сети для каждого сегмента. Уникальные идентификаторы подсетей создаются путем разбиения идентификатора узла на две группы бит.

Первая из них служит для идентификации сегмента объединенной сети, вторая – для идентификации конкретного узла. Такой механизм называется *деление на подсети* (subnet working). Деление на подсети не является необходимым в изолированной сети (т.е. не имеющей выход в Интернет).

### Классовая адресация

Классовая адресация IP сетей — архитектура сетевой адресации, которая использовалась в Интернете в период с 1981 по 1993 годы, до введения бесклассовой междоменной маршрутизации (CIDR - *Classless Inter-Domain Routing*).

Этот метод адресации делит адресное пространство протокола Интернета версии 4 (IPv4) на пять классов адресов: А, В, С, D и Е. Принадлежность адреса к конкретному классу задаётся первыми битами адреса. Каждый класс определяет либо соответствующий размер сети, то есть количество возможных адресов хостов внутри данной сети (классы А, В, С), либо сеть многоадресной передачи (класс D). Диапазон адресов пятого класса (Е) был зарезервирован для будущих или экспериментальных целей.

Определение класса IP-адреса осуществляется по его первому октету, т.е. первым числом, отображающим значения каждого из четырех байтов в десятичной форме. Вообще, любой IP-адрес представляет собой объединение двух логических частей:

- номер сети;
- номер узла в сети.

Именно первые биты IP-адреса и определяют, какая часть такого адреса отображает номер сети, а какая - номер узла в сети. Также по

умолчанию каждый класс адресов использует собственную маску подсети.

Сети класса А имеют адреса, начинающиеся с 0 до 126, и маску подсети 255.0.0.0. При этом, номер 127 предназначен для специальных целей, а номер 0 - не используется. Пример такого адреса - 10.52.36.11, где октетом выступает число 10.

Значение первого октета в диапазоне от 128 до 191 - признак принадлежности сети к классу В. Маской подсети таких сетей является 255.255.0.0. Примером такого адреса может служить 172.16.52.63, в котором первым октетом выступает число 172.

Если IP-адрес начинается с цифр в диапазоне от 192 до 223, то он относится к классу С. Такие адреса используют маску подсети 255.255.255.0. Примером адреса класса С может служить 192.168.123.132, первым октетом которого является номер 192.

Отдельный групповой адрес, или *multicast*, начинающийся с цифр 1110 относится к классу D. Назначение адреса класса D адресом пакета, подразумевает получение этого пакета всеми узлами с таким IP-адресом. Маска подсети адресов класса D - 239.255.255.255.

Еще один класс IP-адресов - это класс E. Первыми цифрами таких адресов является последовательность 11110, а маской подсети, используемой адресами класса E, выступает 247.255.255.255.

### **Определение маски подсети**

Задание маски подсети необходимо, если разбивается сеть на подсети. Для этого нужно выполнить следующие операции.

1. Определить количество физических сегментов в сети и перевести это значение в двоичный формат.
2. Подсчитать, сколько бит необходимо для записи полученного значения в двоичном формате. Например, если в сети шесть сегментов, двоичное значение равно 110, и для его записи в двоичном формате требуется 3 бита.
3. Записать эти биты единицами (количество требуемых бит равно количеству записываемых единиц), дополнив их справа нулями до одного байта. В рассматриваемом примере для идентификатора подсети потребовалось 3 бита. Переведя 11100000 в десятичное число, получим 224. Тогда маска подсети будет иметь вид 255.255.224.0 (для адресов класса В).

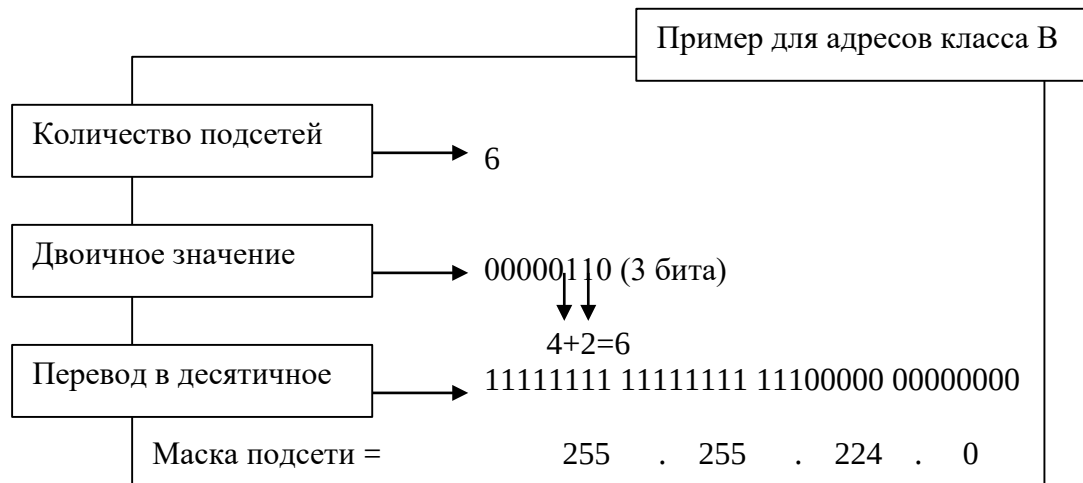


Рисунок 1 – Определение маски подсети

### Таблицы преобразования

Ниже перечислены маски подсетей для сетей класса А.

Таблица 1 – Маски подсетей для сетей класса А.

| 0   | 1 | Не используется | Не используется |
|-----|---|-----------------|-----------------|
| 2   | 2 | 255.192.0.0     | 4194302         |
| 6   | 3 | 255.224.0.0     | 2097150         |
| 14  | 4 | 255.240.0.0     | 1048574         |
| 30  | 5 | 255.248.0.0     | 524286          |
| 62  | 6 | 255.252.0.0     | 262142          |
| 126 | 7 | 255.254.0.0     | 131070          |
| 254 | 8 | 255.255.0.0     | 65534           |

В следующей таблице перечислены маски подсетей для сетей В.

Таблица 2 – Маски подсетей для сетей класса В.

| 0  | 1 | Не используется | Не используется |
|----|---|-----------------|-----------------|
| 2  | 2 | 255.255.192.0   | 16382           |
| 6  | 3 | 255.255.224.0   | 8190            |
| 14 | 4 | 255.255.240.0   | 4094            |
| 30 | 5 | 255.255.248.0   | 2046            |

|     |   |               |      |
|-----|---|---------------|------|
| 62  | 6 | 255.255.252.0 | 1022 |
| 126 | 7 | 255.255.254.0 | 510  |
| 254 | 8 | 255.255.255.0 | 254  |

В следующей таблице перечислены маски подсетей для сетей С.

Таблица 3 – Маски подсетей для сетей класса С.

|                 |   |                 |                 |
|-----------------|---|-----------------|-----------------|
| Не используется | 1 | Не используется | Не используется |
| 2               | 2 | 255.255.255.192 | 62              |
| 6               | 3 | 255.255.255.224 | 30              |
| 14              | 4 | 255.255.255.224 | 14              |
| 30              | 5 | 255.255.255.248 | 6               |
| 62              | 6 | 255.255.255.252 | 2               |
| Не используется | 7 | Не используется | Не используется |
| Не используется | 8 | Не используется | Не используется |

### Определение идентификаторов подсетей

Для задания идентификаторов подсетей используется тоже число бит, что и для соответствующей маски подсети. Определить диапазон идентификаторов подсетей, входящих в объединенную сеть, можно несколькими способами.

Возможные идентификаторы подсети комбинируются из тех бит в адресе узла, которые используются в маске подсети. Ниже приведена процедура, необходимая для определения диапазона идентификаторов подсетей последовательность действий:

1. Выписать все возможные комбинации бит, используемых для формирования маски подсети.
2. Вычеркнуть комбинации, где значения всех бит одновременно равны 0 и 1. Это нужно сделать потому, что они представляют недопустимые IP-адреса: комбинация все “0” означает всю локальную сеть, а “все 1” совпадает с маской подсети.
3. Перевести в десятичный формат значения комбинации бит для каждой подсети. Каждое такое значение представляет

одну сеть и используется для определения диапазона идентификаторов узлов в ней.

|   |          |          |          |                |
|---|----------|----------|----------|----------------|
|   | 255      | 255      | 224      | 0              |
| 1 | 11111111 | 11111111 | 11100000 | 00000000       |
|   |          |          |          | 00000000 = 0   |
|   |          |          |          | 00100000 = 32  |
|   |          |          |          | 01000000 = 64  |
|   |          |          |          | 01100000 = 96  |
|   |          |          |          | 10000000 = 128 |
|   |          |          |          | 10100000 = 160 |
|   |          |          |          | 11000000 = 192 |
|   |          | 2        |          | 3              |

Рисунок 2 – Определение идентификаторов подсетей

### Определение идентификаторов узлов в подсети

Когда определяются идентификаторы подсетей, тем самым задаются идентификаторы узлов в каждой из них.

Каждое очередное значение идентификатора подсети, увеличенное на единицу есть ничто иное, как начало диапазона идентификаторов узлов в подсети. Следующее по порядку возможное значение идентификаторов подсети, уменьшенное на единицу, дает конечное значение диапазона.

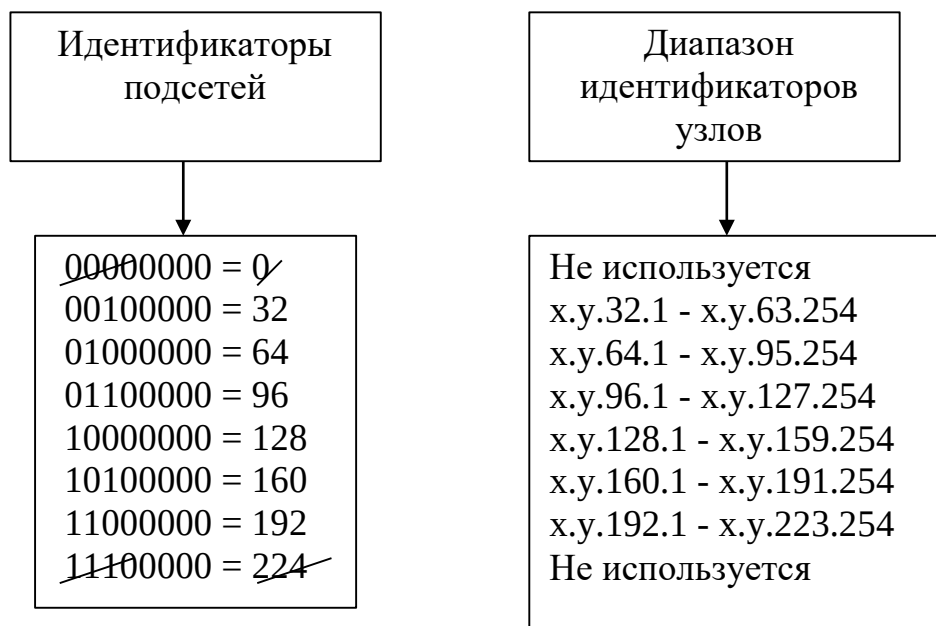


Рисунок 3 – Определение идентификаторов узлов в подсети

В таблице 4 указан диапазон идентификаторов узлов для сети класса В в случае, когда для маски подсети используется 3 бита.

Таблица 4 – Диапазон идентификаторов узлов для сети класса В.

|          |     |                 |                 |
|----------|-----|-----------------|-----------------|
| 00000000 | 0   | Не используется | Не используется |
| 00100000 | 32  | х.у.32.1        | х.у.63.254      |
| 01000000 | 64  | х.у.64.1        | х.у.95.254      |
| 01100000 | 96  | х.у.96.1        | х.у.127.254     |
| 10000000 | 128 | х.у.128.1       | х.у.159.254     |
| 10100000 | 160 | х.у.160.1       | х.у.191.254     |
| 11000000 | 192 | х.у.192.1       | х.у.223.254     |
| 11100000 | 224 | Не используется | Не используется |

## ВЫПОЛНЕНИЕ РАБОТЫ

### Вычисление масок подсети

Необходимо рассчитать количество бит, необходимое для идентификаторов подсети или узла, число бит, оставшееся на идентификатор узла или подсети, маску подсети в виде префикса сети и маску подсети в десятично-точечном виде.

Например, ваш адрес сети - 206.73.118.0/24. Количество бит, необходимое для идентификатора подсети – 3. Оставшееся на идентификатор узла количество бит – 5. Маска подсети (в виде префикса сети) - /27. Маска подсети (в десятично-точечном виде) – 255.255.255.224.

### Вычисление различных параметров подсети

Необходимо определить класс сети и маску подсети по умолчанию для каждого приведенного в таблице идентификатора сети. Затем с помощью калькулятора вычислите действительную маску подсети, назначенную адресу, доступное количество подсетей и количество узлов в каждой подсети.

Таблица 5 – Различные параметры подсети

| Идентификатор сети | Класс сети | Маска подсети по умолчанию | Заданная маска подсети (в десятично-точечном виде) | Доступное количество подсетей | Доступное количество узлов в подсети |
|--------------------|------------|----------------------------|----------------------------------------------------|-------------------------------|--------------------------------------|
|                    |            |                            |                                                    |                               |                                      |

### Вычисление диапазонов адресов подсети.

Необходимо вычислить диапазоны адресов подсети, определив диапазоны первых трех подсетей сети. Для каждого адреса сети и маски подсети (столбец А), вычтите из 256 значение соответствующего октета маски подсети. Запишите полученное значение в колонку В. Затем впишите в колонку С первые четыре кратные единицы (начните с 0) этого значения. С помощью этих значений заполните колонки D и E, как показано в примере.

Таблица 6 – Диапазоны адресов подсети

| (A)<br>Адрес сети и<br>маска подсети | (B)<br>Число групп | (C)<br>Первые четыре<br>кратные<br>единицы В<br>(включая 0) | (D)<br>Начальный<br>адрес<br>диапазонов<br>адресов<br>первых трех<br>подсетей | (E)<br>Конечный<br>адрес<br>диапазонов<br>адресов<br>первых трех<br>подсетей |
|--------------------------------------|--------------------|-------------------------------------------------------------|-------------------------------------------------------------------------------|------------------------------------------------------------------------------|
| 10.0.0.0<br>255.240.0.0              | $256 - 240 = 16$   | 0, 16, 32, 48                                               | 10.0.0.0,<br>10.16.0.0,<br>10.32.0.0                                          | 10.15.255.255,<br>10.31.255.255,<br>10.47.255.255                            |
|                                      |                    |                                                             |                                                                               |                                                                              |

### **Проверка двух адресов на принадлежность одной подсети**

С помощью логической функции «И» можно определить, принадлежат ли два адреса одной и той же логической подсети. Необходимо просто выполнить две операции «И » над соответствующими октетами маски подсети и октетами заданного IP-адреса. Если результаты совпадут, адреса принадлежат одной логической подсети.

### **ЗАДАНИЕ**

1. Вычислите маску подсети
2. Вычислите параметры подсети
3. Вычислите диапазоны адресов подсети
4. Проверьте два адреса на принадлежность одной подсети

### **КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Общие сведения о подсетях
2. Определение маски подсети
3. Определение идентификаторов узлов в подсети

### **СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ**

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов [Текст]/ В.Г. Олифер, Н.А. Олифер, 4-е изд.: СПб.: Питер, 2006 -958 с.
2. Хелби. С., Принципы маршрутизации в Internet: Учебник для вузов. [Текст]/ С. Хелби, 2-е изд.: Вильямс. 2001 – 402 с.
3. Танненбаум Э., Д. Уэзеролл Компьютерные сети [Текст]/ Э.Танненбаум, Д. Уэзеролл, 5-е Изд.: СПб.: Питер, 2012 - 903 с.

# **МИНОБРНАУКИ РОССИИ**

Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Юго-Западный государственный университет»  
(ЮЗГУ)

Кафедра информационной безопасности



## **СРЕДА МОДЕЛИРОВАНИЯ GNS3**

Методические указания по выполнению  
лабораторной и практической работ для студентов укрупненной  
группы специальностей и направлений подготовки 10.00.00.

Курск 2024

УДК 004

Составители: Терещенко Е.М., Чуйкова В.В.

Рецензент

Доктор физико-математических наук, профессор кафедры  
Информационная безопасность *В.П. Добрица*

**Среда моделирования GNS3:** методические указания по выполнению лабораторной и практической работ / Юго-Зап. гос. Ун- т; сост. Е.М.Терещенко, В.В. Чуйкова, 2024, 15 с.: ил. 9.; Библиогр.: с.15.

Содержат сведения по вопросам первичной установки, настройки и моделирования сетей в среде GNS3. Указывается порядок выполнения лабораторной и практической работ, правила оформления и содержание отчета.

Методические указания по выполнению лабораторной и практической работ по дисциплинам «Безопасность сетей ЭВМ», «Сети и системы передачи информации», «Сети и системы передачи информации (специальные разделы)», «Администрирование вычислительных сетей», «Администрирование защищенных телекоммуникационных систем» для студентов укрупненной группы специальностей и направлений подготовки 10.00.00.

Текст печатается в авторской редакции

Подписано в печать 16.05.2024. Формат 60x84 1/16.

Усл.печ.л.0,81. Уч.–изд.л.0,74. Тираж 30 экз. Заказ .Бесплатно.373

Юго-Западный государственный университет.

305040, г. Курск, ул. 50 лет Октября, 94.

## СОДЕРЖАНИЕ

|                                        |    |
|----------------------------------------|----|
| Введение.....                          | 4  |
| Цель работы .....                      | 5  |
| Порядок выполнения работы.....         | 5  |
| Содержание отчета .....                | 5  |
| Теоретическая часть .....              | 6  |
| Задание .....                          | 13 |
| Контрольные вопросы .....              | 13 |
| Список информационных источников ..... | 14 |

## ВВЕДЕНИЕ

GNS3 (Graphical Network Simulator) - среда моделирования компьютерных сетей, использующих сетевое оборудование, функционирующее на базе процессоров с архитектурой MIPS. К таким сетевым устройствам относятся, в том числе, большинство сетевых коммутаторов и маршрутизаторов, производимых компанией CISCO.

В отличие от других моделирующих сред, например, имитатора Cisco Packet Tracer, среда моделирования GNS3 является эмулятором, что позволяет использовать реальные образы сетевых устройств с ОС Cisco IOS.

Свою историю среда GNS3 начинает с 2007 года, в котором Джереми Гроссман (Jeremy Grossman) занимался выполнением выпускной квалификационной работы и ему было необходимо создать среду моделирования компьютерных сетей. В основу создаваемого программного продукта легла разработка эмулятора MIPS устройств Dynamips и его графического интерфейса Dynagen.

В дальнейшем среда GNS3 получила широкое распространение и теперь является одним из популярных сред для изучения компьютерных сетей и отработки различных промышленных решений.

Среда эмуляции состоит из четырех компонентов:

- Dynamips – среда моделирования сетевых устройств, реализованных на базе процессоров с MIPS архитектурой. Для своего функционирования требует наличие образов операционных систем iOS сетевых устройств CISCO. Допускает выполнение и иных операционных систем;
- VCPS, VirtualBox, QEMU – программы виртуализации хост-машин. Используются для эмулирования оконечных сетевых устройств или промежуточных устройств, реализованных на базе ЭВМ с архитектурой IBM/PC;
- Графического интерфейса (собственно GNS3), который предоставляет графически схемы построения сети и доступа к компонентам Dynagen, Dynamips, обеспечивая управление образами узлов и проектом в целом.

Для своего функционирования среда GNS3 использует также стороннее программное обеспечение:

- WinPCAP – системный драйвер и библиотеку функций, позволяющих получить доступ к сетевым интерфейсам физического компьютера и передаваемой/получаемой информации по ним.

Используется для анализа трафика, передаваемого по сети;

- Wireshark – графический анализатор сетевого трафика. Позволяет наглядно отобразить подробнейшую информацию о сетевом трафике. Используется как внутри среды GNS3, так и для анализа трафика реальной компьютерной сети (считывая его с физических интерфейсов с помощью драйвера WinPCAP);

- SolarWinds Response – среду для анализа сетевого трафика. Используется для графического отображения информации, подготовленной Wireshark;

- SuperPUTTY – систему виртуальных терминалов. Позволяет подключаться к сетевым устройствам для управления ими, используя командную строку;

- Cpulimit – средство ограничения объемов потребления процессорного времени.

## **ЦЕЛЬ РАБОТЫ**

Цель лабораторной работы – получить навыки использования среды GNS3 для моделирования компьютерных сетей на базе сетевых устройств, в том числе устройств Cisco.

## **ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ**

1. Получить задание
2. Изучить теоретическую часть
3. Выполнить практическое задание
4. Сформировать выводы

## **СОДЕРЖАНИЕ ОТЧЕТА**

1. Титульный лист, цель работы
2. Задание в соответствии с вариантом
3. Материалы выполненного задания
4. Выводы

## ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

### 1. Интерфейс среды GNS3 и её начальная настройка

После установки и запуска среды GNS3 пользователь видит главное рабочее окно (см. рисунок 1), содержащее традиционные элементы (строку заголовка, меню, строку состояния) и шесть рабочих областей:

- Панели инструментов (1 и 2), используются для доступа к основным инструментам работы с моделью компьютерной сети, включая сохранение проекта, его открытие и создание нового;
- Окно сетевых устройств (блоков, для построения модели сети, 3), используется для выбора элементов для размещения в создаваемой модели сети;
- Рабочая область, отражающая созданную модель сети (4);
- Перечень устройств, которые задействованы в модели сети (5);
- Командная панель, используется для управления моделью с использованием командной строки (6). Перечень команд и их описание доступны по команде help.

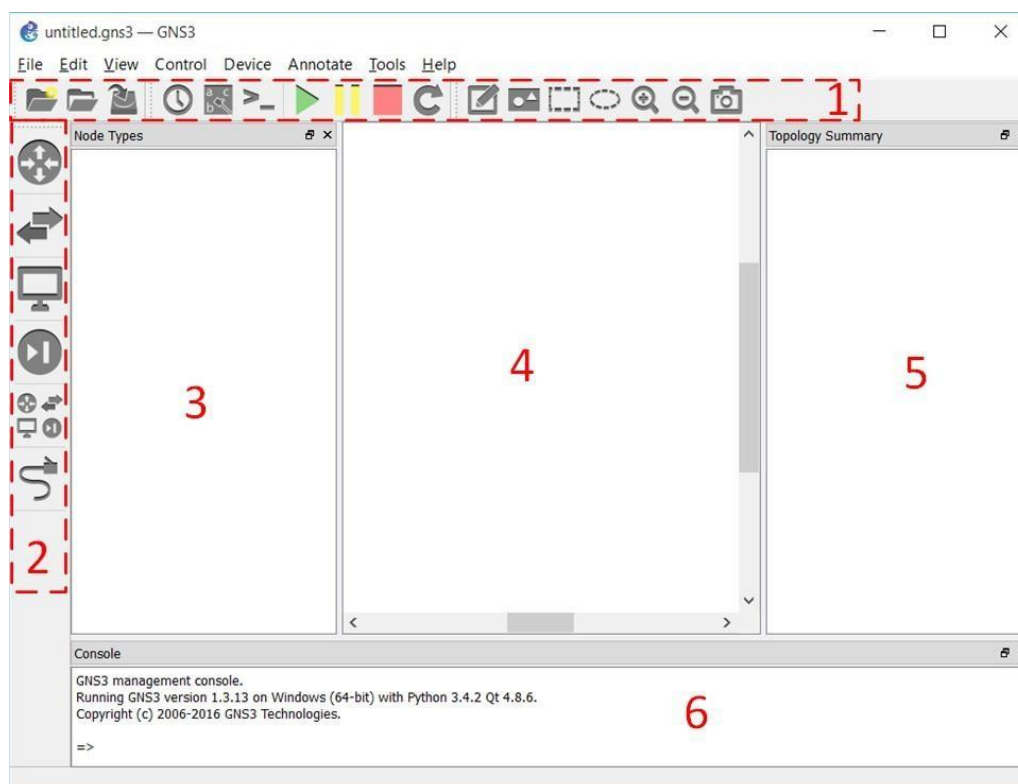


Рисунок 1 – Главное окно программы GNS3

Управление отдельным сетевым устройством осуществляется через контекстное меню, доступное при наведении на него курсора мышки и нажатии правой кнопки мышки (см. рисунок 2). В этом меню доступны также элементы включения (Start), выключения (Stop) и приостановки (Suspend), а также: конфигурирования устройства (Configure, см. рисунок 3), изменения имени устройства (Change hostname), изменения изображения устройства в модели (Change symbol), доступ к консоли управления (Console), доступ к порту управления (Auxiliary console), Сохранения и восстановления конфигурации устройства (import config, export config, save config), просмотр сетевого трафика (capture), настройка модельных параметров (idle-pc, auto idle-pc), изменения прикрепления к слоям отображения модели (raise one layer, lower one layer) и удаления устройства из модели (Delete).

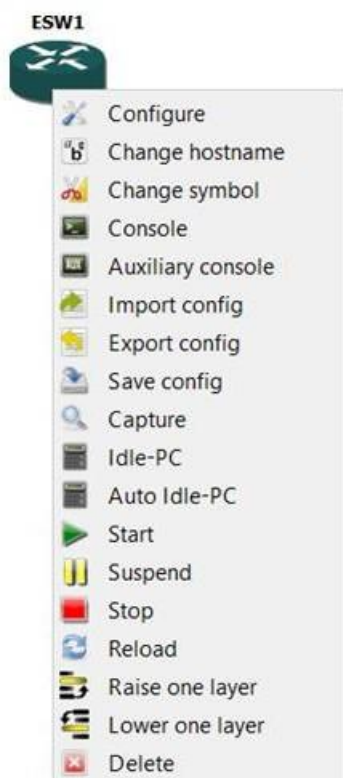


Рисунок 2 – Контекстное меню управления элементом модели (на примере маршрутизатора)

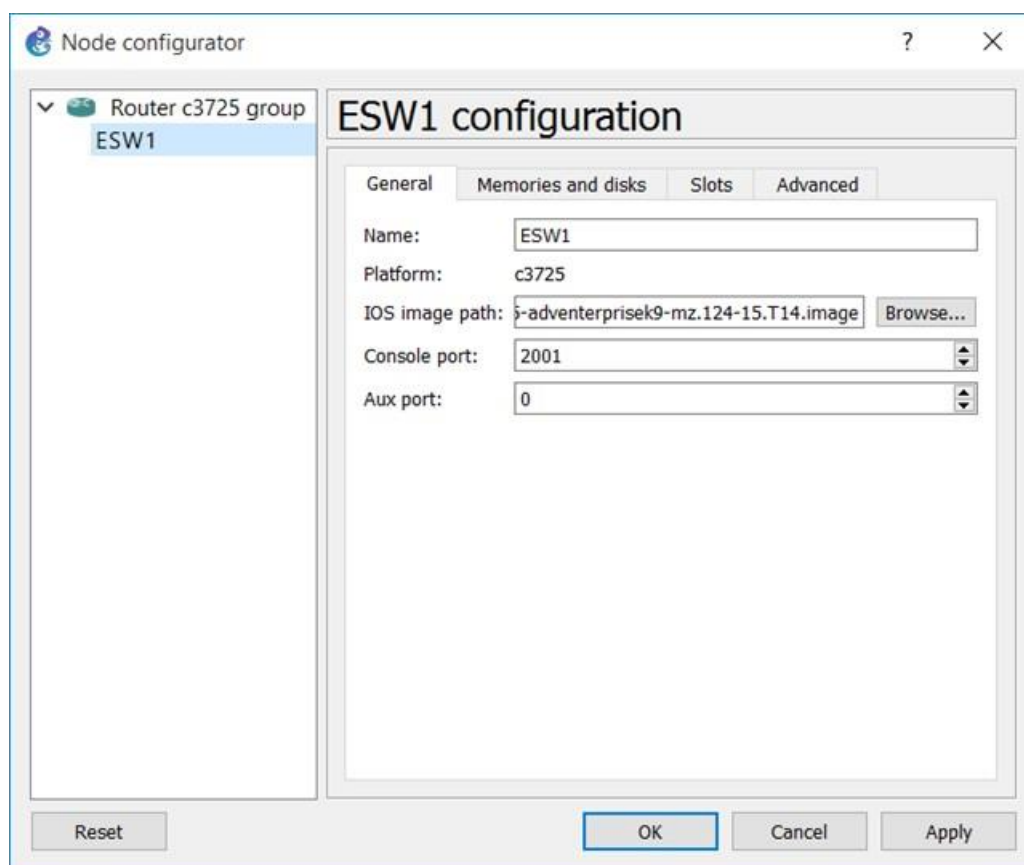


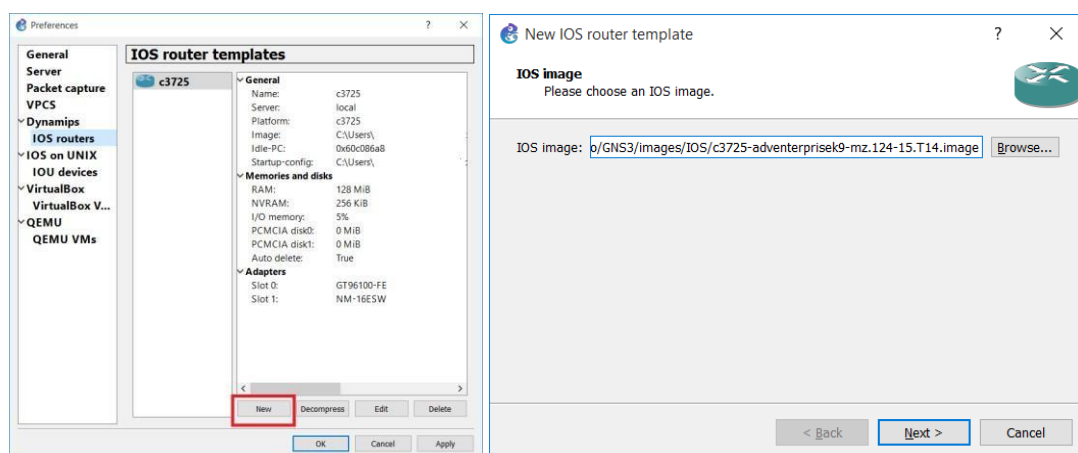
Рисунок 3 – Окно конфигурации маршрутизатора

Параметры конфигурирования зависят от типа устройств. Некоторые из параметров будут рассмотрены ниже.

При первом запуске системы GNS необходимо создать модели устройств, которые будут использоваться в создаваемом проекте компьютерной сети. В рамках лабораторной работы планируется использование трех типов устройств: коммутатор, маршрутизатор, персональный компьютер.

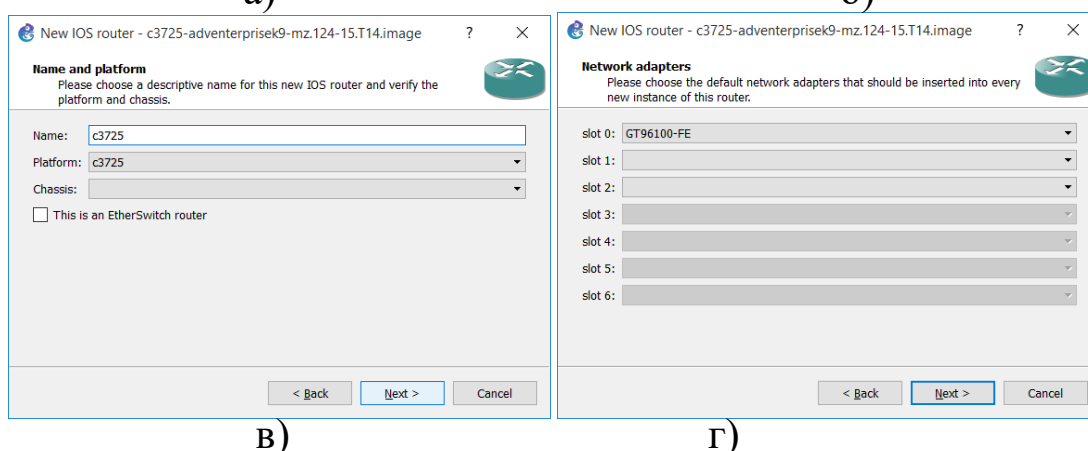
В качестве коммутатора будет использоваться встроенная модель коммутатора, имеющая несколько интерфейсов Ethernet. В режиме конфигурирования для этой модели доступны следующие параметры: количество интерфейсов, режим работы интерфейса (access или trunk) и номер VLAN, к которому этот интерфейс относится. В рамках лабораторной работы будет использован только параметр «количество интерфейсов».

Для создания маршрутизаторов необходимо загрузить в систему моделирования образ операционных системы и сформировать соответствующий модельный элемент. Сделать это можно в режиме настройки среды (пункт меню Edit->Preferences), далее выбрав в дереве параметров пункт Dynamips->iOS routers (см. рис. 4).



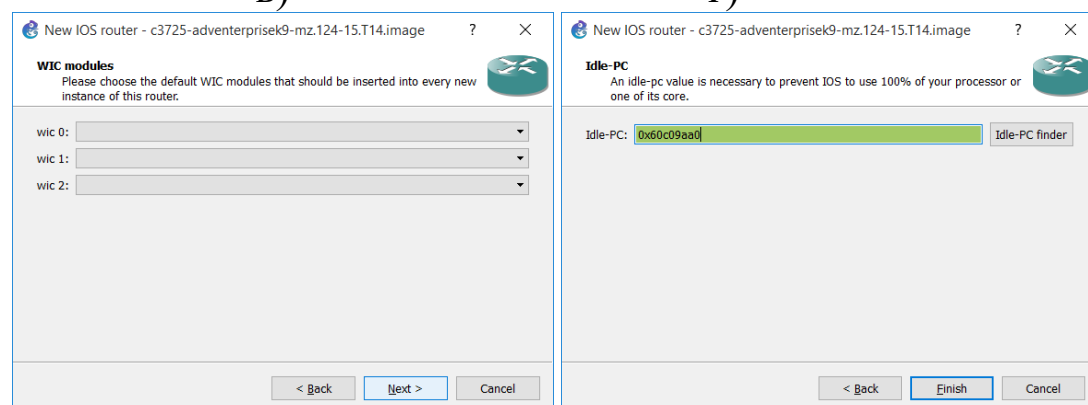
а)

б)



в)

г)



д)

е)

Рисунок 4 – Окно создания модельных элементов – маршрутизаторов (а – начальное окно, б – выбор файла с «образом» операционной системы, в) выбор типа модели создаваемого сетевого устройства и его корпуса, г) определение заполнения «больших» слотов расширения, д) определение заполнения «малых» слотов расширения, е) число Idle-PC)

Конфигурация слотов расширения для оборудования CISCO производится аналогично, как и в системе Cisco Packet Tracer.

Объем памяти, необходимый для функционирования операционной системы определяется в соответствии с техническим

описанием соответствующих устройств компании CISCO.

Особое внимание необходимо обратить на параметр Idle-PC, определяющий степень максимальной загрузки процесса, которую может достичь выполнение dynamips, реализующей этот маршрутизатор. Этот параметр предлагается системой автоматически при создании модельного объекта. Одна рекомендация после создания модели сети и первого запуска модели провести перерасчет этих параметров, используя контекстное меню (пункт меню IdlePC). При этом перерасчет этого значения производится сразу для всех объектов сети, созданных на базе одного модельного объекта.

Для создания моделей персональных компьютеров необходимо настроить одну из сред виртуализации: VirtualBox, QEMU или VirtualPC. В рамках лабораторной работы будет использована среда VirtualBox, в которой создаются пустые контейнеры (см. рис. 5).

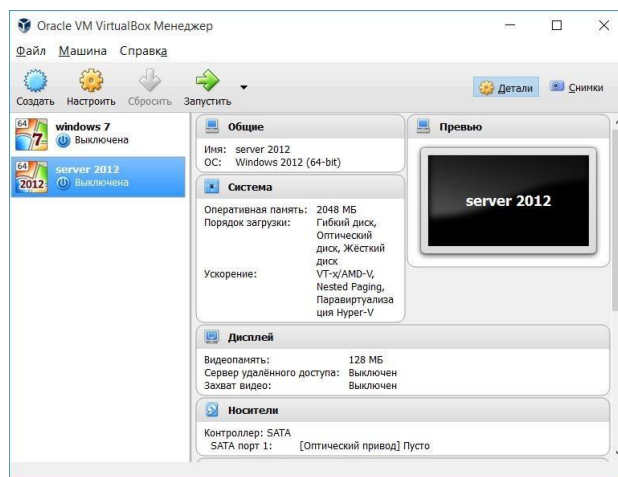
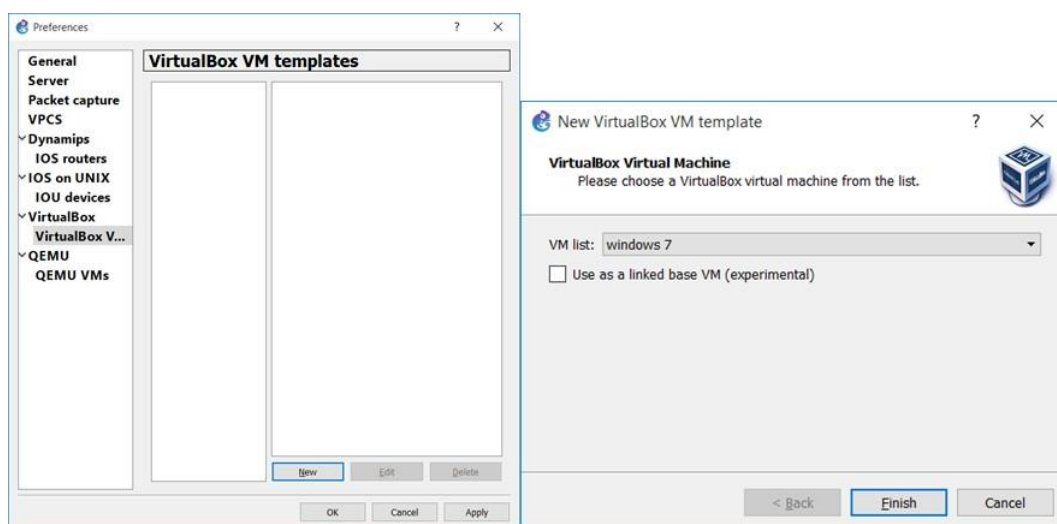


Рисунок 5 – Окно менеджера VirtualBox с двумя пустыми контейнерами

После создания виртуальных машин их необходимо подключить к среде моделирования GNS3. Сделать это можно также в режиме настройки среды (пункт меню Edit->Preferences), далее выбрав в дереве параметров пункт VirtualBox->VirtualBox VM templates (см. рис. 6).



а)

б)

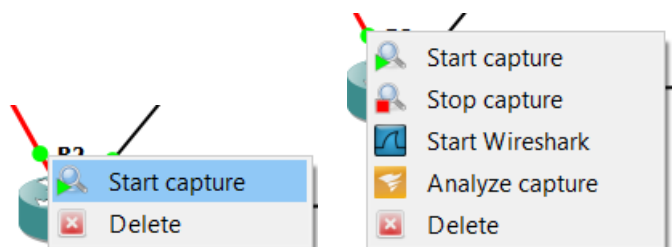
Рисунок 6 – Подключение виртуальных машин и их конфигурирование

Здесь следует отметить, что для связи с виртуальными машинами по эмулируемой компьютерной среде будет использовать протокол TCP/IP и потребуется разрешить доступ двух сред к соответствующим сетевым портам.

Для взаимодействия с физической сетевой средой в модели предусмотрено два объекта – Host и Cloud.

## 2. Анализ сетевого трафика с использованием Wireshark

Среда моделирования GNS3 позволяет проводить анализ трафика, генерируемого всеми сетевыми устройствами модели на всех интерфейсах. Для этого необходимо на интересующих интерфейсах включить сбор информации о сетевом трафике выбрав пункт контекстного меню Start Capture. (см. рис. 7).



а)

б)

Рисунок 7 – Настройка сбора информации о сетевом трафике (а) и запуска анализатора сетевого трафика(б).

Запустив Wireshark пользователь получает возможность наблюдать за всеми пакетами, проходящими через соответствующий интерфейс, а также смотреть их содержимое по уровням инкапсуляции (см. рис. 8). В приведенном примере в окне программы отображается (сверху вниз): список из 4 пакетов, содержимое одного из пакетов, содержащего данные по протоколу ARP (с расшифровкой уровня MAC и данных по протоколу ARP) и его представления в двоичном виде.

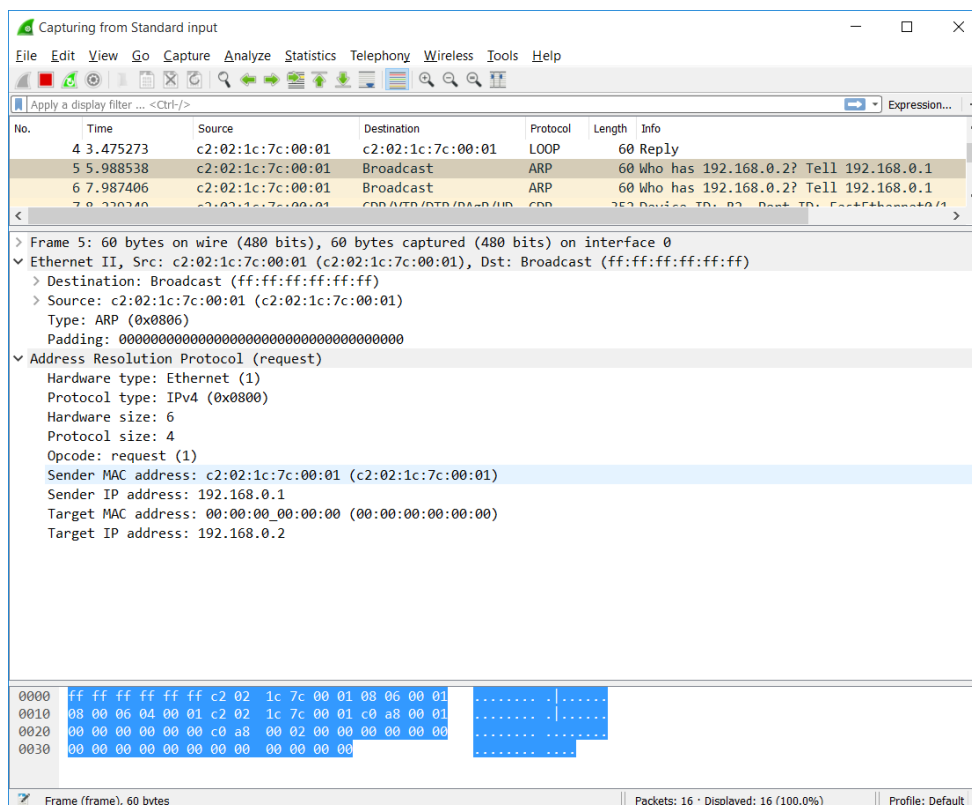


Рисунок 8 – Окно программы Wireshark

### 3. ОСНОВНЫЕ ПРИЕМЫ СОЗДАНИЯ СХЕМЫ СЕТИ

Для создания схемы моделируемой сети необходимо настроить среду, как описано в разделе 1: настроить конфигурацию среды, скачать и загрузить в область выбора устройств образы сетевых устройств. Для перемещения пиктограмм устройств из области выбора в рабочую область использовать левую кнопку мыши. Нажатие правой кнопки выводит контекстное меню, состав которого зависит от типа устройства.

### 3.1 Простейшая сеть

На этом этапе для освоения основных приемов построения, конфигурирования и исследования работы сети рекомендуется открыть в Ютубе тему «Основы GNS3» и просмотреть лекцию №5 «Простейшая сеть». Изучить материал построения сети и конфигурирования интерфейсов устройств с использованием командной строки терминала и основных команд.

Изучив материал и формат построения команд конфигурирования интерфейсов открыть GNS3 и повторить создание и конфигурирование простейшей сети, используя материал лекции. Полученные результаты в виде схемы сети и листинга настройки интерфейсов и проверки работоспособности сети сохранить для отчета.

### 3.2 Режимы командной строки

При работе в командной строке необходимо знать основные режимы ее работы.

В операционной системе Cisco IOS имеются два основных пользовательских режима для администрирования коммутатора и несколько других режимов, позволяющих контролировать конфигурацию устройства. В дополнение к различным режимам программное обеспечение Cisco IOS обеспечивает такие функции, как интерактивная справка и редактирование командной строки, которые позволяют взаимодействовать с коммутатором в административных целях.

1) Пользовательский EXEC-режим: **Switch>** в который входит терминал при его запуске пользователем. В этом режиме пользователю доступен ограниченный набор команд.

2) Привилегированный EXEC-режим: **Switch> enable Switch#**, который предоставляет полный доступ ко всем командам ОС. Перейти в этот режим можно из пользовательского режима, применив команду **enable**. Для выхода из привилегированного режима используют команду **disable** или **exit**.

3) Конфигурационный режим: **Switch#configure terminal Switch(config)#**, войти в который можно из привилегированного режима. В режиме конфигурации можно вводить любые команды

для настройки сетевых устройств, которые доступны в программном образе ОС IOS.

Конфигурационный режим организован иерархически. Режим глобальной конфигурации (global configuration mode) содержит команды, которые влияют на устройство в целом в целом. В режиме конфигурирования интерфейса (interface configuration mode) администратору предоставляются команды, позволяющие настраивать интерфейсы устройств в зависимости от настраиваемого ресурса.

Для перехода со специфического уровня конфигурирования на более общий вводится команда **exit**. Для того чтобы покинуть режим глобальной конфигурации и вернуться в привилегированный EXEC-режим необходимо также ввести команду **exit**. Для того чтобы покинуть любой конфигурационный режим и вернуться в привилегированный EXEC-режим, применяется команда **end**.

Вид приглашения командной строки в контекстах конфигурирования, которые будут встречаться наиболее часто:

**Switch(config)#** – глобальный;

**Switch(config-line)#** – терминальной линии,

**Switch(config-if)#** – интерфейса; например:

**Switch(config-if)#shutdown** – отключить интерфейс;

**Switch(config-if)#no shutdown** – включить интерфейс.

### 3.3 Моделирование стандартной компьютерной сети

1. Запустите среду моделирования GNS3 и проведите начальную конфигурацию, добавив маршрутизатор CISCO и две виртуальные машины от VirtualBox.
2. Соберите макет локальной сети, как показано на рис. 9.

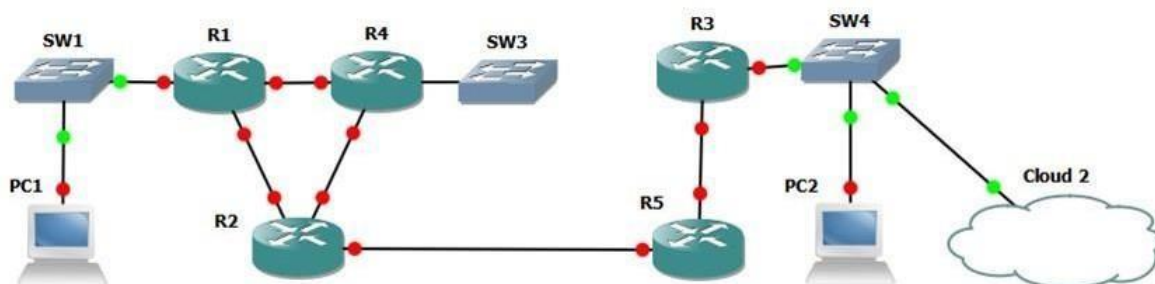


Рисунок 9 – Конфигурация моделируемой компьютерной сети

3. Исходя из того, что для функционирования создаваемой сети выделен диапазон адресов 10.255.0.0/16 определите сколько подсетей необходимо задать.
4. Настройте все интерфейсы всех маршрутизаторов и статическую маршрутизацию. Убедитесь, что имеется связь между всеми сетевыми интерфейсами всех маршрутизаторов.
5. Используя анализатор Wireshark, продемонстрируйте принцип работы ping между двумя маршрутизаторами, расположенными в разных подсетях (необходимо показать все генерируемые пакеты в прямом и в обратном пути при одном запросе ping).
6. Убедитесь, что Ваша среда имеет связь со средой другого студента используя реальную физическую сеть.

### **КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Для чего была разработана среда GNS3?
2. Какие устройства моделируются в GNS3?
3. Что такое Idle-PC?
4. Как получить доступ к консоли конфигурирования Маршрутизатора CISCO (показать).
5. Зачем используется Wireshark?
6. Что такое Dynamips?
7. Какие среды виртуализации использует GNS3?

### **СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ**

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов [Текст]/ В.Г. Олифер, Н.А. Олифер, 4-е изд.: СПб.: Питер, 2006 -958 с.
2. Software - Download - GNS3 [Электронный ресурс]: / Internet. - <http://www.gns3.com> (14.09.17).
3. Электронный каталог Documentation [Электронный ресурс]: / Internet. - <http://www.gns3.net/documentation/> (14.09.17).

# **МИНОБРНАУКИ РОССИИ**

Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Юго-Западный государственный университет»  
ЮЗГУ)

Кафедра информационной безопасности



## **ИЗУЧЕНИЕ СТАТИЧЕСКОЙ МАРШРУТИЗАЦИИ**

Методические указания по выполнению лабораторной и  
практической работы для студентов укрупненной группы  
специальностей и направлений подготовки 10.00.00.

Курск 2024

УДК 004

Составители: Е.М. Терещенко, В.В. Чуйкова.

Рецензент

Доктор физико-математических наук, профессор кафедры  
Информационная безопасность *В.П. Добрица*

**Изучение статической маршрутизации:** методические указания к выполнению лабораторной и практической работы/ Юго-Зап. гос. Ун-т; сост. Е.М. Терещенко, В.В. Чуйкова. Курск, 2024, 16 с.: ил. 6.; Библиогр.: с. 16.

Содержат сведения по вопросам настройки статической маршрутизации в среде GNS3. Указывается порядок выполнения лабораторной и практической работы, правила оформления, содержание отчета.

Методические указания по выполнению лабораторных и практических работ по дисциплинам «Безопасность систем и сетей передачи данных», «Сети и системы передачи информации», «Сети и системы передачи информации (специальные разделы)», «Администрирование вычислительных сетей», «Администрирование защищенных телекоммуникационных систем» для студентов укрупненной группы специальностей и направлений подготовки 10.00.00.

Текст печатается в авторской редакции

Подписано в печать 16.05.2024. Формат 60x84 1/16. Усл. печ. л. 0,93. Уч. –изд.л. 0,84. Тираж 30 экз. Заказ . Бесплатно. 376

Юго-Западный государственный университет.  
305040, г. Курск, ул. 50 лет Октября, 94.

## СОДЕРЖАНИЕ

|                                       |    |
|---------------------------------------|----|
| Введение .....                        | 4  |
| Цель работы .....                     | 4  |
| Порядок выполнения работы.....        | 4  |
| Содержание отчета .....               | 5  |
| Теоретическая часть .....             | 6  |
| Выполнение работы.....                | 8  |
| Задание.....                          | 16 |
| Контрольные вопросы.....              | 16 |
| Список информационных источников..... | 16 |

## **ВВЕДЕНИЕ**

Статическая маршрутизация - вид маршрутизации, при котором маршруты указываются в явном виде при конфигурации маршрутизатора. Вся маршрутизация при этом происходит без участия каких-либо протоколов маршрутизации.

При задании статического маршрута указывается:

- Адрес сети (на которую маршрутизируется трафик), маска сети
- Адрес шлюза (узла), который способствует дальнейшей маршрутизации (или подключен к маршрутизируемой сети напрямую)
- (опционально) метрика (иногда именуется также "ценой") маршрута. При наличии нескольких маршрутов на одну и ту же сеть некоторые маршрутизаторы выбирают маршрут с минимальной метрикой

## **ЦЕЛЬ РАБОТЫ**

Цель лабораторной работы – настроить маршрутизаторы 1, 2, 3 с IP адресами, затем добавить статические маршруты для всех маршрутизаторов.

## **ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ**

1. Получить задание
2. Изучить теоретическую часть
3. Выполнить практическое задание
4. Написать вывод

## **СОДЕРЖАНИЕ ОТЧЕТА**

1. Титульный лист
2. Задание в соответствии с вариантом
3. Выполненное задание
4. Вывод

## ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Статическая маршрутизация является одним из способов задания маршрута следования пакетов в сетях. При этом протоколы маршрутизации не используются, а необходимая информация заносится системным администратором вручную в соответствующие таблицы маршрутизации. Использование статической маршрутизации разумно в небольших локальных сетях, где поднятие сервера ДНСР для раздачи динамических IP-адресов оказывается более трудоемкой и ресурсозатратной задачей. Статические маршруты обладают преимуществом при передаче данных в каналах с узкой полосой пропускания, например, в аналоговых коммутируемых соединениях (через модем) или соединениях типа «точка-точка», поэтому они могут использоваться для создания резервного канала в случае отказа основного. Степень надежности маршрута определяется значением так называемой административной дистанции. Для статического маршрута оно равно 1, что ставит его на второе место по надежности после прямого соединения источника и получателя. Для того чтобы обеспечить резервный канал, достаточно изменить значение административной дистанции для определенного статического маршрута. Тогда при «падении» основного канала весь трафик заданной сети пойдет через резервный до восстановления основного. Этот вид маршрутизации называется плавающей статической маршрутизацией.

К статическим маршрутам также относится маршрут по умолчанию (default route). Он задается для тех пакетов сети, к которым не подходит ни одно из описанных правил в таблице маршрутизации, и позволяет осуществлять координацию трафика. Зачастую функцию шлюза по умолчанию выполняет центральный маршрутизатор, который связан со всеми остальными узлами и одной-двумя сетями.

Достоинства статической маршрутизации проявляются в полной мере в малых сетях. Она позволяет достаточно оперативно развернуть локальную сеть без затраты дополнительного времени на конфигурирование протоколов маршрутизации, а также снизить нагрузку на маршрутизатор за счет использования данных таблиц маршрутизации. Однако при масштабировании или изменении топологии такой сети могут возникнуть проблемы с ее администрированием, так как статические маршрутизаторы не обмениваются информацией друг с другом и с

динамическими маршрутизаторами и не сообщают о наличии сбоя в каком-либо из маршрутизаторов или в канале связи. Внесение изменений в конфигурацию таблиц будет занимать больше времени, что в разы снижает эффективность сети со статической маршрутизацией. Тем не менее, в средних и крупных сетях грамотное использование преимуществ статической маршрутизации в сочетании с динамической в качестве основного способа распределения трафика позволяет обеспечить достаточно эффективную передачу данных.

Использование статической маршрутизации дает преимущества небольшим локальным сетям или отдельным сегментам сети с медленным каналом, в которых дополнительная нагрузка в виде служебного трафика влияет на скорость передачи данных. В средних и больших сетях статическая маршрутизация применима только в сочетании с динамической, что повышает эффективность сети и облегчает ее администрирование.

## ВЫПОЛНЕНИЕ РАБОТЫ

В GNS3 соберите топологию, как показано на рис. 1

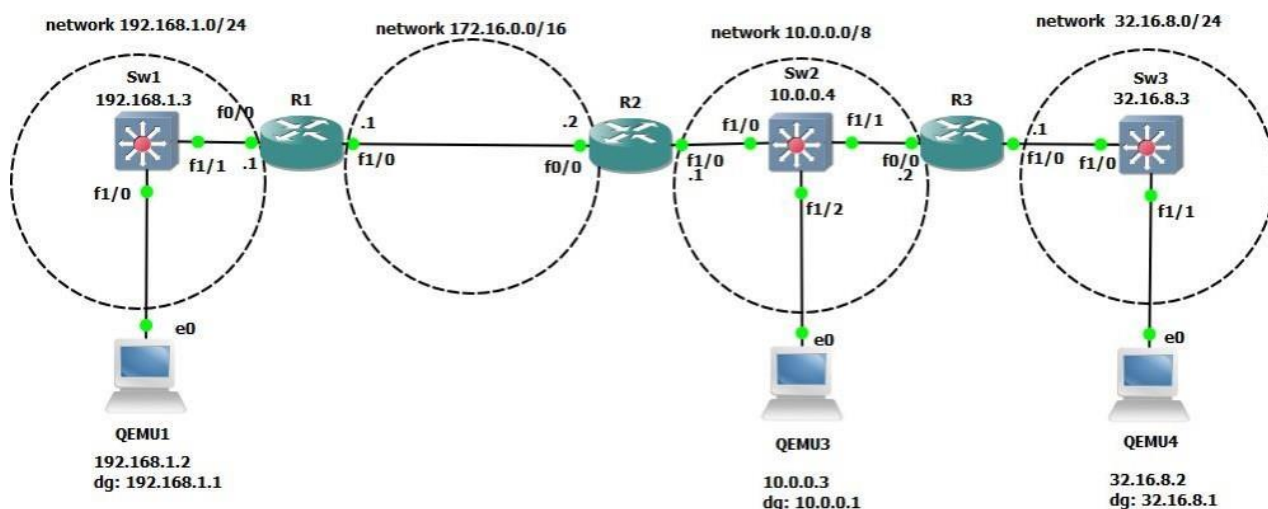


Рисунок 1 - Исследуемая топология

Настройте IP адреса для каждого маршрутизатора, как это сделано в предыдущих лабораторных работах.

### Настройка IP параметров для хостов

В данной работе в качестве операционной системы хостов используются ОС Linux, для настройки ip параметров каждой linux машины необходимо ввести следующие команды в окне терминал:

Qemu1:

`Sudo su` // Вход в режим суперпользователя (режим администратора)  
`ifconfig eth0 192.168.1.2 netmask 255.255.255.0` // Задание ip адреса и маски хоста

`route add -net 0/0 gw 192.168.1.1` // Задание ip-адреса шлюза по умолчанию (default gateway)

Qemu2:

`Sudo su` // Вход в режим суперпользователя (режим администратора)  
`ifconfig eth0 10.0.0.3 netmask 255.0.0.0` // Задание ip адреса и маски хоста  
`route add -net 0/0 gw 10.0.0.1` // Задание ip-адреса шлюза по умолчанию (default gateway)

Qemu3:

`Sudo su` // Вход в режим суперпользователя (режим администратора)  
`ifconfig eth0 32.16.8.2 netmask 255.255.255.0` // Задание ip адреса и маски хоста

route add -net 0/0 gw 32.16.8.1 // Задание ip-адреса шлюза по умолчанию (default gateway)

Эти же команды можно ввести и используя средства графического интерфейса.

### **Настройка IP параметров для коммутаторов**

Особенностью настройки коммутаторов является, то, что ip адрес присваивается не конкретному интерфейсу, а всему коммутатору, для этого необходимо зайти в специальный интерфейс Vlan 1, все остальные команды очевидны.

Настройка коммутатора Sw1:

```
Sw1>ena
Sw1>config t
Sw1(config)#interface vlan 1
Sw1(config-if)#ip address 192.168.1.3 255.255.255.0 Sw1(config-if)#no
shutd
Sw1(config-if)#exit
Sw1(config)# ip default-gateway
192.168.1.1 Sw1(config)#^Z
Sw1#ping 192.168.1.2
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/27/56 ms
```

Настройка коммутатора Sw2:

```
Sw2>ena
Sw2>config t
Sw2(config)#interface vlan 1
Sw2(config-if)#ip address 10.0.0.4
255.0.0.0 Sw2(config-if)#no shutd
Sw2(config-if)#exit
Sw2(config)# ip default-gateway
10.0.0.1 Sw2(config)#^Z
Sw2#ping 10.0.0.1
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/27/56 ms
```

Настройка коммутатора Sw3:

```
Sw3>ena
```

```

Sw3>config t
Sw3(config)#interface vlan 1
Sw3(config-if)#ip address 32.16.8.3
255.255.255.0 Sw3(config-if)#no shutd
Sw3(config-if)#exit
Sw3(config)# ip default-gateway
32.16.8.1 Sw3(config)#^Z
Sw3#ping 10.0.0.1
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/27/56 ms

```

После задания IP адресов всем устройствам, используя утилиту ping, и проверьте доступность устройств. На данном этапе настройки связи на сетевом уровне будет возможно только в рамках одной IP подсети, это означает, что связь устройств находящихся в разных подсетях невозможна, так как еще не настроены таблицы маршрутизации.

### Настройка удаленного доступа

На каждом маршрутизаторе и коммутаторе настройте удаленный доступ по протоколу telnet. Для настройки удаленного доступа по telnet необходимо задать пароль для входа в привилегированный режим, а также пароль для авторизации по протоколу telnet

```

Sw1(config)#enable secret qwerty // установка зашифрованного пароля
"qwerty" для привилегированного режима
Sw1(config)#line vty 0 4 //установка количества одновременных сессий
по протоколу telnet в данном случае, с устройством
можно установить одновременно пять сеансов telnet
Sw1(config-line)#login // данная команда инструктирует коммутатор о
том, что нужно запрашивать пароль без имени Sw1(config-line)#password
asdfg // пароль для доступа по протоколу telnet
Sw1(config-line)#exit

```

После настроек зайдите на хост и попробуйте ввести команду  
**telnet** [*ip адрес устройства*]

Пример для хоста qemu1:

```
root@box:~# telnet 192.168.1.3
```

```
Entering character mode
Escape character is '^I'.
```

```
User Access Verification
```

```
Password: _
```

Рисунок 2 – Ввод команды telnet для хоста Qemu1

Далее необходимо ввести пароль, указанный для доступа по протоколу telnet, после чего вы попадете в режим настройки маршрутизатора R1, для выхода из терминала наберите команду **quit**.

### Установка статических маршрутов

Теперь вам нужно установить статические маршруты на каждом маршрутизаторе для сетей, которые не присоединены непосредственно. Для сетей, которые не подключены непосредственно к маршрутизатору, необходимо сконфигурировать статические маршруты. Например, к маршрутизатору R1 подключены сети 192.168.1.0/24 и 172.16.0.0/16, остальные сети: 10.0.0.0/8 и 32.16.8.0/24, - не являются непосредственно подключенными к маршрутизатору, поэтому для этих сетей необходимо сконфигурировать статические маршруты, используя команду `ip route`, в режиме конфигурации терминала:

Создание статических маршрутов для R1:

```
R1(config)#ip route 10.0.0.0 255.0.0.0 172.16.0.2
```

```
R1(config)#ip route 32.16.8.0 255.255.255.0 172.16.0.2
```

Команду `ip route 10.0.0.0 255.0.0.0 172.16.0.2` необходимо понимать следующим образом: если в `ip` пакете указан адрес хоста, находящегося в сети 10.0.0.0/8, то такой пакет необходимо переслать следующему устройству, чей `ip` адрес равен 172.16.0.2

Теперь попробуйте «пропинговать» с хоста Qemu1 устройства, которые находятся в сетях 10.0.0.0 и 32.16.8.0, хотя в маршрутизаторе R1 и прописаны статические маршруты к этим сетям, но ответа от узлов получено не будет, так как остальные

маршрутизаторы не имеют в своих таблицах информации о сети 192.168.1.0/24, куда надо отправить ответный пакет.

Сконфигурируйте остальные маршруты на других маршрутизаторах и попробуйте "пропинговать" с Qemu1 все остальные устройства.

```
R2(config)#ip route 192.168.1.0 255.255.255.0 172.16.0.1
```

```
R2(config)#ip route 32.16.8.0 255.255.255.0 10.0.0.2
```

```
R3(config)#ip route 192.168.1.0 255.255.0.0 10.0.0.1
```

```
R3(config)#ip route 172.16.0.0 255.255.0.0 10.0.0.1
```

Для просмотра текущей таблицы маршрутизации на любом маршрутизаторе наберите команду:

```
R1#show ip route
```

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA -

OSPF inter area N1 - OSPF NSSA external type 1,

N2 - OSPF NSSA external type 2 E1 - OSPF

external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1,

L2 - IS-IS level-2 ia - IS-IS inter area, \* -

candidate default, U - per-user static route o -

ODR, P - periodic downloaded static route

Gateway of last resort is not

set 32.0.0.0/24 is

subnetted, 1 subnets

```
S 32.16.8.0 [1/0] via 172.16.0.2
```

```
C 172.16.0.0/16 is directly connected,
```

```
FastEthernet1/0 S 10.0.0.0/8 [1/0] via
172.16.0.2
```

```
C 192.0.0.0/8 is directly connected, FastEthernet0/0
```

Рисунок 3 – Просмотр текущей таблицы маршрутизации

S - статический маршрут, ведущий к соответствующей сети через (via) определенный интерфейс, цифры в квадратных скобках указывают приоритет/стоимость данного маршрута.

После указания статических маршрутов, любые устройства из любой сети могут взаимодействовать между собой, для проверки

этого утверждения зайдите на любую Vbox машину и «пропингуйте» все устройства, а также проверьте работу удаленного доступа по протоколу telnet.

### Создание таблицы хостов

Таблица хостов может быть использована для установления имён для обычно используемых IP адресов, которые помогают при поиске и устранении неисправностей. Для создания таблицы хостов используется следующая команда:

**ip host** *[имя хоста]* *[ip адрес хоста]*

Данная команда связывает имя хоста с его ip адресом, после этой команды в место ip адресов можно использовать символьные имена.

Пример, создания таблицы хостов на маршрутизаторе R1

```
R1(config)#ip host R2
172.16.0.2 R1(config)#ip
host R3 10.0.0.2
R1(config)#ip host Sw1
192.168.1.3 R1(config)#ip
host Sw2 10.0.0.4
R1(config)#ip host Sw3
32.16.8.1
```

Теперь попробуйте проверить соединение R3 из R1, просто командой ping R3.

```
R1#ping R3
```

Или попробуйте соединиться по протоколу telnet с коммутатором Sw2, используя команду telnet Sw2.

```
R1#telnet Sw2
```

Используйте команду show hosts для проверки, что введенное сохранено таблице хостов маршрутизатора.

```
R1#show hosts
```

## Работа с таблицами ARP

После разрешения IP адреса в MAC с помощью протокола ARP, данное пара IP-MAC сохраняется в кэш памяти устройства, которое использовало протокол ARP, для просмотра таблицы ARP на маршрутизаторах и коммутаторах необходимо ввести команду в привилегированном режиме:

```
R1# show arp
```

Для отчистки кэша в привилегированном режиме наберите команду:

```
R1# clear arp-cache
```

Для создания статической записи в кэше необходимо зайти в режим конфигурации терминала и ввести команду:

```
arp [ip адрес] [mac адрес] arpa
```

Пример, для создания статической записи, что IP адрес 1.0.0.10 соответствует MAC адресу 1111.2222.3333 необходимо ввести следующую команду:

```
R1(config)#arp 1.0.0.10 1111.2222.3333 arpa
```

## Сохранение настроек

Во время настройки маршрутизаторы хранят вводимые команды в оперативной памяти в файле running-config, поэтому после перезапуска, все данные о настройках будут стерты, поэтому с целью предотвращения потери конфигурации необходимо сохранить введенные команды. Сохранение идет в энергонезависимую память маршрутизатора/коммутатора в файл startup-config, с которого маршрутизатор берет первоначальную конфигурацию при загрузке.

Для сохранения зайдите в меню файл и выберите пункт «Сохранить проект» (см. рис. 4).

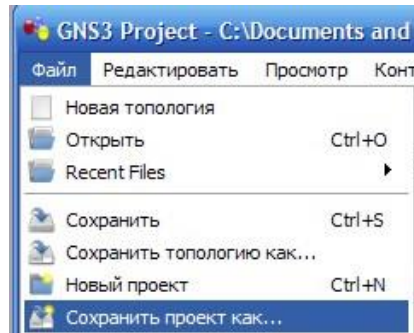


Рисунок 4 – Сохранение проекта, этап 1

Перед вами появится сообщение, нажмите кнопку «ДА» (см. рис. 5).

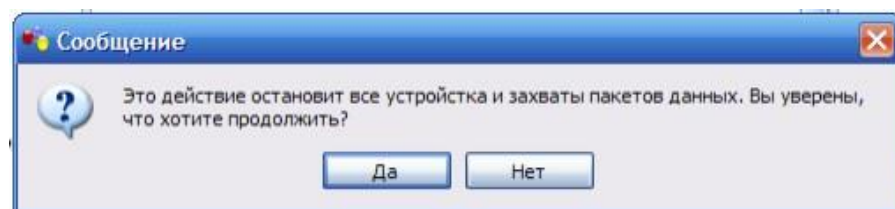


Рисунок 5 – Сохранение проекта, этап 2

Далее появится окно «Save Project As...», где вам необходимо указать имя проекта и путь к папке будут храниться файлы проекта, также галочкой должно быть отмечено «Сохранить базовые файлы конфигурации», нажмите ОК (см. рис. 6).

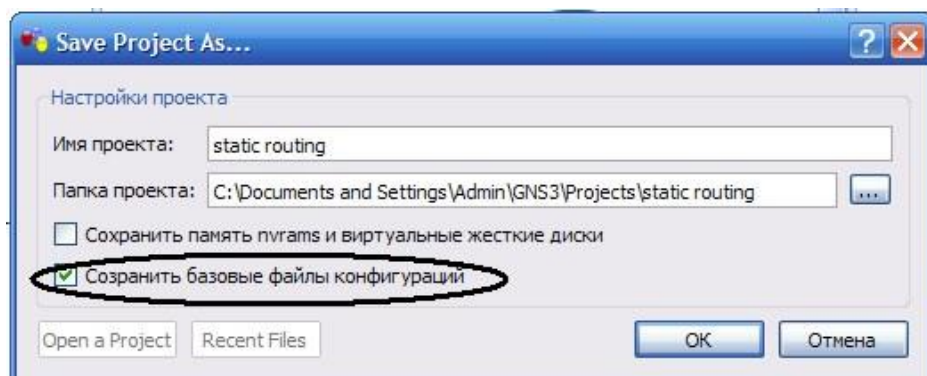


Рисунок 6 - Сохранение проекта, этап 3

Затем в консолях всех маршрутизаторов и коммутаторов введите следующие команды:

```
R1#copy running-config startup-config
```

```
Destination filename [startup-config]? // нажмите Enter
```

В результате данные из файла рабочей конфигурации перезапишутся в файл стартовой конфигурации. Потом нажмите кнопку «Сохранить» на панели инструментов  если вы внесли какие-либо изменения в конфигурацию устройства, не забывайте набирать команду «copy running-config startup-config» и нажимать кнопку «Сохранить», автосохранение не предусмотрено! Файлы конфигурации хранятся в папке проекта в папке «configs», сами файлы имеют расширения \*.cfg и могут быть открыты любым текстовым редактором.

## **ЗАДАНИЕ**

1. Составьте исходную топологию;
2. Получите листинг команд конфигурации, либо конфигурационные файлы;
3. Создайте таблицу хостов;
4. Сделайте захваты следующих пакетов с помощью программы Wireshark: Arp request, Arp reply, ICMP echo request, ICMP echo reply, поясните значение всех полей данных пакетов.

## **КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Структура IP пакета;
2. Процедура фрагментации и сборки IP пакета;
3. Структура ICMP пакетов echo request, echo reply;
4. Структура ARP пакетов request и reply;
5. Объясните структуру таблицы маршрутизации;
6. Каково назначение сетевой маски;
7. Что такое суммирование маршрутов;
8. Поясните процедуру разбиения сети на подсети;

## **СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ**

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов [Текст]/ В.Г. Олифер, Н.А. Олифер, 4-е изд.: СПб.: Питер, 2006 -958 с.
  2. Хелби. С., Принципы маршрутизации в Internet: Учебник для вузов. [Текст]/ С. Хелби, 2-е изд.: Вильямс. 2001 – 402 с.
  3. Электронный каталог Documentation [Электронный ресурс]: / Internet. - <http://www.gns3.net/documentation/> (15.10.17).
- .

# **МИНОБРНАУКИ РОССИИ**

Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Юго-Западный государственный университет»  
(ЮЗГУ)

Кафедра информационной безопасности



## **ИЗУЧЕНИЕ РАБОТЫ КОММУТАТОРОВ ETHERNET**

Методические указания по выполнению лабораторной и практической  
работ для студентов укрупненной группы специальностей и  
направлений подготовки 10.00.00.

Курск 2024

УДК 004

Составители: Е.М. Терещенко, В.В. Чуйкова

Рецензент

Доктор физико-математических наук, профессор кафедры  
Информационная безопасность *В.П. Добрица*

**Изучение работы коммутаторов Ethernet:** методические указания к выполнению лабораторной и практической работ/ Юго-Зап. гос. Ун-т; сост. Е.М. Терещенко, В.В.Чуйкова. Курск, 2024, 15 с.: ил. 4.; Библиогр.: с. 15.

Содержат сведения по настройке и работе коммутаторов Ethernet в среде GNS3. Указывается порядок выполнения лабораторных и практических работ, правила оформления, содержание отчета.

Методические указания по выполнению лабораторной и практической работ по дисциплинам «Безопасность систем и сетей передачи данных», «Сети и системы передачи информации», «Сети и системы передачи информации (специальные разделы)», «Администрирование вычислительных сетей», «Администрирование защищенных телекоммуникационных систем» для студентов укрупненной группы специальностей и направлений подготовки 10.00.00.

Текст печатается в авторской редакции

Подписано в печать 16.05.2024. Формат 60х84 1/16. Усл. печ.  
л. 0,87. Уч. –изд.л. 0,79. Тираж 30 экз. Заказ . Бесплатно. 377

Юго-Западный государственный университет.  
305040, г. Курск, ул. 50 лет Октября, 94.

## СОДЕРЖАНИЕ

|                                       |    |
|---------------------------------------|----|
| Введение .....                        | 4  |
| Цель работы .....                     | 4  |
| Порядок выполнения работы.....        | 4  |
| Содержание отчета .....               | 5  |
| Теоретическая часть .....             | 6  |
| Выполнение работы.....                | 9  |
| Задание.....                          | 14 |
| Контрольные вопросы .....             | 14 |
| Список информационных источников..... | 15 |

## ВВЕДЕНИЕ

**Сетевой коммутатор** (от англ. *switch* — переключатель) — устройство, предназначенное для соединения нескольких узлов компьютерной сети в пределах одного или нескольких сегментов сети. Коммутатор работает на канальном (втором) уровне модели OSI. Коммутаторы были разработаны с использованием мостовых технологий и часто рассматриваются как многопортовые мосты. Для соединения нескольких сетей на основе сетевого уровня служат маршрутизаторы (3 уровень OSI).

В отличие от концентратора (1 уровень OSI), который распространяет трафик от одного подключённого устройства ко всем остальным, коммутатор передаёт данные только непосредственно получателю (исключение составляет широковещательный трафик всем узлам сети и трафик для устройств, для которых неизвестен исходящий порт коммутатора). Это повышает производительность и безопасность сети, избавляя остальные сегменты сети от необходимости (и возможности) обрабатывать данные, которые им не предназначались.

## ЦЕЛЬ РАБОТЫ

Цель лабораторной работы – освоить правило формирования таблиц коммутации в коммутаторах Ethernet и формирование логических топологий с помощью протокола покрывающего дерева STP.

## ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Получить задание
2. Изучить теоретическую часть
3. Выполнить практическое задание
4. Написать вывод

## СОДЕРЖАНИЕ ОТЧЕТА

1. Титульный лист
2. Задание в соответствии с вариантом
3. Выполненное задание
4. Вывод

## ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

### Принцип работы коммутатора

Коммутатор хранит в памяти (т.н. ассоциативной памяти) таблицу коммутации, в которой указывается соответствие MAC-адреса узла порту коммутатора. При включении коммутатора эта таблица пуста, и он работает в режиме обучения. В этом режиме поступающие на какой-либо порт данные передаются на все остальные порты коммутатора. При этом коммутатор анализирует фреймы (кадры) и, определив MAC-адрес хоста-отправителя, заносит его в таблицу на некоторое время. Впоследствии, если на один из портов коммутатора поступит кадр, предназначенный для хоста, MAC-адрес которого уже есть в таблице, то этот кадр будет передан только через порт, указанный в таблице. Если MAC-адрес хоста-получателя не ассоциирован с каким-либо портом коммутатора, то кадр будет отправлен на все порты, за исключением того порта, с которого он был получен. Со временем коммутатор строит таблицу для всех активных MAC-адресов, в результате трафик локализуется.

Стоит отметить малую латентность (задержку) и высокую скорость пересылки на каждом порту интерфейса.

### Режимы коммутации

Существует три способа коммутации. Каждый из них — это комбинация таких параметров, как время ожидания и надёжность передачи.

1. С промежуточным хранением (Store and Forward). Коммутатор читает всю информацию в кадре, проверяет его на отсутствие ошибок, выбирает порт коммутации и после этого посылает в него кадр.
2. Сквозной (cut-through). Коммутатор считывает в кадре только адрес назначения и после выполняет коммутацию. Этот режим уменьшает задержки при передаче, но в нём нет метода обнаружения ошибок.
3. Бесфрагментный (fragment-free) или *гибридный*. Этот режим является модификацией сквозного режима. Передача осуществляется после фильтрации фрагментов коллизий (первые 64 байта кадра анализируются на наличие ошибки и при её отсутствии кадр обрабатывается в сквозном режиме).

Задержка, связанная с «принятием коммутатором решения», добавляется к времени, которое требуется кадру для входа на порт коммутатора и выхода с него, и вместе с ним определяет общую задержку коммутатора.

### **Симметричная и асимметричная коммутация**

Свойство симметрии при коммутации позволяет дать характеристику коммутатора с точки зрения ширины полосы пропускания для каждого его порта. Симметричный коммутатор обеспечивает коммутируемые соединения между портами с одинаковой шириной полосы пропускания, например, когда все порты имеют ширину пропускания 10 Мб/с или 100 Мб/с.

Асимметричный коммутатор обеспечивает коммутируемые соединения между портами с различной шириной полосы пропускания, например, в случаях комбинации портов с шириной полосы пропускания 10 Мб/с или 100 Мб/с и 1000 Мб/с.

Асимметричная коммутация используется в случае наличия больших сетевых потоков типа клиент-сервер, когда многочисленные пользователи обмениваются информацией с сервером одновременно, что требует большей ширины пропускания для того порта коммутатора, к которому подсоединён сервер, с целью предотвращения переполнения на этом порте. Для того, чтобы направить поток данных с порта 100 Мб/с на порт 10 Мб/с без опасности переполнения на последнем, асимметричный коммутатор должен иметь буфер памяти.

Асимметричный коммутатор также необходим для обеспечения большей ширины полосы пропускания каналов между коммутаторами, осуществляемых через вертикальные кросс-соединения, или каналов между сегментами магистрали.

### **Буфер памяти**

Для временного хранения фреймов и последующей их отправки по нужному адресу коммутатор может использовать буферизацию. Буферизация может быть также использована в том случае, когда порт пункта назначения занят. Буфером называется область памяти, в которой коммутатор хранит передаваемые данные.

Буфер памяти может использовать два метода хранения и отправки фреймов: буферизация по портам и буферизация с общей памятью. При буферизации по портам пакеты хранятся в очередях (queue), которые связаны с отдельными входными портами. Пакет передаётся на выходной порт только тогда, когда все фреймы, находившиеся впереди него в очереди, были успешно переданы. При этом возможна ситуация, когда один фрейм задерживает всю очередь из-за занятости порта его пункта назначения. Эта задержка может происходить даже в том случае, когда остальные фреймы могут быть переданы на открытые порты их пунктов назначения.

При буферизации в общей памяти все фреймы хранятся в общем буфере памяти, который используется всеми портами коммутатора. Количество памяти, отводимой порту, определяется требуемым ему количеством. Такой метод называется динамическим распределением буферной памяти. После этого фреймы, находившиеся в буфере, динамически распределяются по выходным портам. Это позволяет получить фрейм на одном порте и отправить его с другого порта, не устанавливая его в очередь.

Коммутатор поддерживает карту портов, в которые требуется отправить фреймы. Очистка этой карты происходит только после того, как фрейм успешно отправлен.

Поскольку память буфера является общей, размер фрейма ограничивается всем размером буфера, а не долей, предназначенной для конкретного порта. Это означает, что крупные фреймы могут быть переданы с меньшими потерями, что особенно важно при асимметричной коммутации, то есть, когда порт с шириной полосы пропускания 100 Мб/с должен отправлять пакеты на порт 10 Мб/с.

### **Возможности и разновидности коммутаторов**

Коммутаторы подразделяются на управляемые и неуправляемые (наиболее простые).

Более сложные коммутаторы позволяют управлять коммутацией на сетевом (третьем) уровне модели OSI. Обычно их именуют соответственно, например «Layer 3 Switch» или сокращенно «L3 Switch». Управление коммутатором может осуществляться посредством Web-

интерфейса, интерфейса командной строки (CLI), протокола SNMP, RMON и т. п.

Многие управляемые коммутаторы позволяют настраивать дополнительные функции: VLAN, QoS, агрегирование, зеркалирование. Многие коммутаторы уровня доступа обладают такими расширенными возможностями, как сегментация трафика между портами, контроль трафика на предмет штормов, обнаружение петель, ограничение количества изучаемых mac-адресов, ограничение входящей/исходящей скорости на портах, функции списков доступа и т.п.

Сложные коммутаторы можно объединять в одно логическое устройство — стек — с целью увеличения числа портов. Например, можно объединить 4 коммутатора с 24 портами и получить логический коммутатор с 90  $((4*24)-6=90)$  портами либо с 96 портами (если для стекирования используются специальные порты).

## ВЫПОЛНЕНИЕ РАБОТЫ

В программе GNS3 составьте следующую топологию (рис. 1).

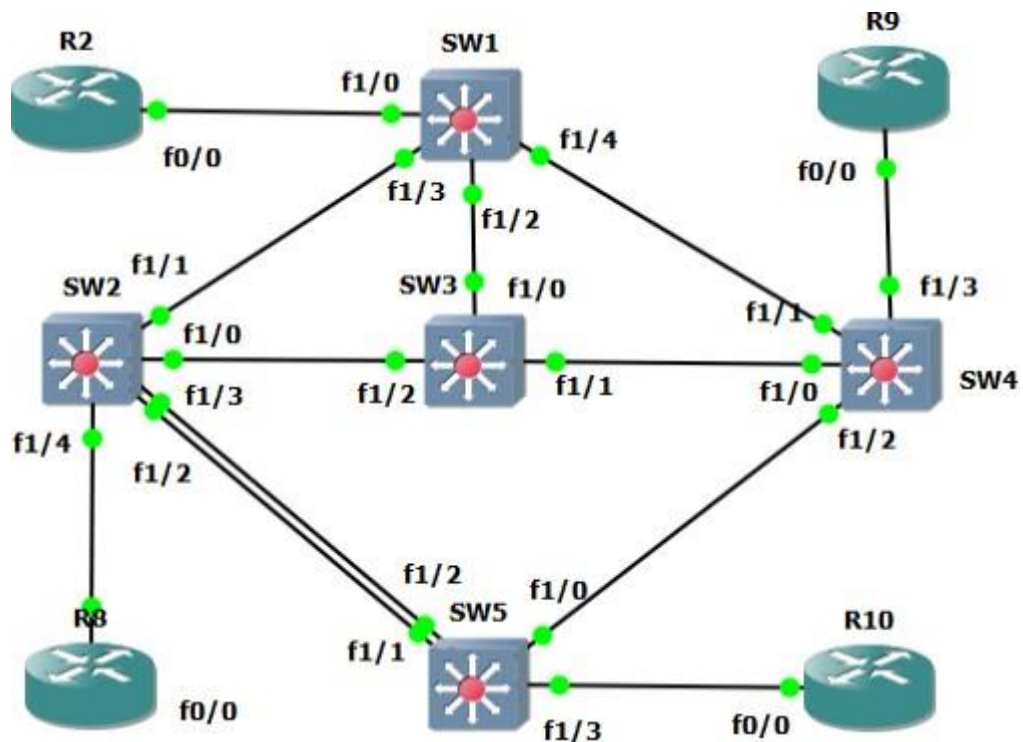


Рисунок 1 - Топология сети

Для моделирования коммутаторов воспользуйтесь

значком  в панели "Типы узлов", данный узел оснащен 2 портами маршрутизатора f0/0 и f0/1, а также 16 портами коммутатора f1/0 - f1/15, коммутаторы соединяйте между собой только через интерфейсы f1/x. Для работы коммутатора подключите образ маршрутизатора Cisco IOS c3745. В качестве хостов выбраны маршрутизаторы 3640, для которых в настройках необходимо добавить 1 слот NM-1FE-TX.

С помощью параметра IdlePC отрегулируйте загрузку центрального процессора вашего персонального компьютера для повышения быстродействия.

Настройте IP адреса для маршрутизаторов на примере R2:

1. На маршрутизаторе войдите в режим глобальной конфигурации  
R2>enable  
R2#conf t

R2(config)#

2. Введите команду, чтобы войти в режим конфигурации интерфейса FastEthernet 0/0.

R2(config)#interface f0/0

R2(config-if)#

3. Выведите на экран все доступные команды режима конфигурации интерфейса командой «?».

R2(config-if)#?

4. Команда **no shutdown** включает выбранный интерфейс, команда **shutdown** выключает интерфейс. Выполните команду на FastEthernet 0/0, маршрутизатора 1 для запуска интерфейса.

5. Введите команду, которая установит IP адрес 192.168.1.1 255.255.255.0 на интерфейс FastEthernet 0/0 и включите интерфейс.

R2(config-if)#ip address 192.168.1.1 255.255.255.0

6. Включите интерфейс, введя следующую команду:

R2(config-if)#no shutdown

7. Добавьте описание этого интерфейса.

R2(config-if)#description Ethernet interface on R2

8. Для просмотра описания интерфейса, вернитесь обратно в привилегированный режим и запустите команду **show interface**. Вы должны увидеть описание FastEthernet 0/0.

R2(config-if)#end

R2#show interface f 0/0

**FastEthernet0/0 is up, line protocol is up**

**Hardware is AmdFE, address is cc0a.194c.0000 (bia  
cc0a.194c.0000) Internet address is 192.168.1.102/24**

**---- more ----**

Рисунок 2 – Просмотр описания интерфейса

Обратите внимание на MAC адрес интерфейсов. Повторите пункты с первого по восьмой для хостов R8, R9, R10, только в качестве IP адресов возьмите соответственно 192.168.1.100, 192.168.1.101 и 192.168.1.102.

Проверьте связь между всеми маршрутизаторами с помощью

утилиты ping, для этого зайдите на любой из маршрутизаторов и в привилегированном режиме введите команду **ping [ip адрес]**, как показано ниже:

```
R2# ping 192.168.1.100
```

Появляющиеся восклицательные знаки говорят о доступности соответствующих устройств, если вместо восклицательных знаком появляется точка ".", это говорит о недоступности устройства, в этом случае проверьте правильность введенных вами IP адресов и номера интерфейсов, которыми соединены между собой все устройства.

### Просмотр таблицы коммутации

После того как вы «пропинговали» все маршрутизаторы, коммутаторы составили таблицу коммутации, которую можно посмотреть введя в привилегированном режиме команду:

```
Sw1#show mac-address-table
```

После чего будет выведена следующая таблица:

| Destination Address | Address Type | VLAN | Destination Port |
|---------------------|--------------|------|------------------|
| c40c.0b28.0000      | Self         | 1    | Vlan1            |
| cc02.194c.0000      | Dynamic      | 1    | FastEthernet1/0  |
| c40e.19cc.f100      | Dynamic      | 1    | FastEthernet1/2  |
| c40e.19cc.f102      | Dynamic      | 1    | FastEthernet1/3  |
| c40e.19cc.0000      | Dynamic      | 1    | FastEthernet1/2  |
| cc0b.06a4.0000      | Dynamic      | 1    | FastEthernet1/3  |

Рисунок 3 – Просмотр таблицы коммутации

Как видно в таблице коммутации в столбце Address Type указаны два типа записей: Self и Dynamic. Self - указывает на свой MAC адрес, Dynamic указывает на то, что данная запись была получена при обучении коммутатора. Помимо этого в таблицу коммутации можно заносить и статические записи с использованием следующей команды:

**mac-address-table static [H.H.H {мас адрес}] interface [номер интерфейса] vlan [номер VLAN]**

**Пример:** необходимо указать коммутаторы, что устройство с MAC адресом 1111.2222.3333 закреплено за интерфейсом fastEthernet 1/1, и оно находится в VLAN 1, такая команда будет выглядеть следующим образом:

```
SW2(config)#mac-address-table static 1111.2222.3333 interface
fastethernet 1/1 vlan 1
```

```
SW1#show mac-address-table
```

| Destination Address | Address Type | VLAN | Destination Port |
|---------------------|--------------|------|------------------|
| c40c.0b28.0000      | Self         | 1    | Vlan1            |
| cc02.194c.0000      | Dynamic      | 1    | FastEthernet1/0  |
| 1111.2222.3333      | Static       | 1    | FastEthernet1/1  |

Рисунок 4 – Таблица MAC-адресов

Обратите внимание на новый тип записи – Static

### Изменение идентификаторов протокола STP

В качестве изменяемых идентификаторов в протоколе STP могут выступать приоритеты мостов и значение стоимости канала связи.

#### Изменение приоритетов мостов

Данные для изменения работы протокола STP необходимо вводить в режиме конфигурации терминала.

**spanning-tree vlan [номер Vlan] priority [значение приоритета]**

**Пример.** Необходимо назначить значение приоритета равное 10000 коммутатору SW1 для VLAN 1:

```
SW1(config)#spanning-tree vlan 1 priority 10000
```

Введите данную команду на коммутаторе SW1, предварительно запустив анализатор протокола между SW1 и SW2, обратите внимание на то, что сразу произойдет отправка сообщения об изменении топологии TCN (topology change notification), а если коммутатор SW1 был корневым коммутатором, то будет отправлено конфигурационное сообщение с флагом TC=1, который указывает на факт изменения топологии.

Еще одной командой для изменения приоритетов коммутаторов является команда:

**spanning-tree vlan [номер Vlan] root primary diameter [диаметр сети] hello-time [значение таймера]**

Данная команда заставит работать коммутатор как корневой, путем установки собственного приоритета меньшего чем у текущего

корневого. диаметр сети указывает максимальное количество коммутаторов между двумя самыми удаленными коммутаторами в сети.

В качестве команды `primary` можно указать `secondary`, заставив коммутатор, установить приоритет больший, чем у корневого, но меньший чем у всех остальных.

Сконфигурируйте SW3 как **secondary**, а SW4 как **primary**, и проанализируйте вновь полученную топологию протокола STP.

## ЗАДАНИЕ

1. Составьте исходную топологию.
2. Составьте следующую таблицу, как это показано ниже, для каждого коммутатора.

Таблица 1 - Идентификаторы коммутаторов

| Имя коммутатора | MAC адрес      | Приоритет | Идентификатор порта |        |
|-----------------|----------------|-----------|---------------------|--------|
| SW2             | c40c.0b28.0000 | 32768     | FastEthernet1/0     | 128.41 |
|                 |                |           | FastEthernet1/1     | 128.42 |
|                 |                |           | FastEthernet1/2     | 128.43 |
|                 |                |           | FastEthernet1/3     | 128.44 |
|                 |                |           | FastEthernet1/4     | 128.45 |
| ....            | ....           | ....      | ...                 |        |

3. По данным из таблицы 1 определите логическую топологию, которую составил протокол STP и состояние каждого порта.

4. Приведите три захваченных сообщения с помощью Wireshark: конфигурационное сообщение, конфигурационное сообщение с установленными флагами TC и TCA, сообщение об уведомлении изменения топологии TCN;

## КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Процедура составления таблицы коммутации;
2. Методы коммутации;
3. Структура MAC адреса;
4. Структура кадра Ethernet;
5. В чем разница между повторителем и коммутатором;
6. Объясните алгоритм работы метода доступа к среде CSMA/CD;
7. Какой минимальный размер кадра определен стандартом для технологии Ethernet;
8. Перечислите скорости, на которых может работать технология Ethernet;

## СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов [Текст]/ В.Г. Олифер, Н.А. Олифер, 4-е изд.: СПб.: Питер, 2006 -958 с.
2. Сергеев А.Н. Основы локальных компьютерных сетей [Текст]/ Сергеев А.Н.: Изд.: «Лань», 2016. 184 с.
3. Танненбаум Э., Д. Уэзеролл Компьютерные сети [Текст]/ Э.Танненбаум, Д. Уэзеролл, 5-е Изд.: СПб.: Питер, 2012 - 903 с.
4. Электронный каталог Documentation [Электронный ресурс]: / Internet. - <http://www.gns3.net/documentation/> (20.10.17).

## **МИНОБРНАУКИ РОССИИ**

Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Юго-Западный государственный университет»  
(ЮЗГУ)

Кафедра информационной безопасности



## **АНАЛИЗ РАБОТЫ ТЕХНИКИ VLAN**

Методические указания по выполнению лабораторной и практической  
работы для студентов укрупненной группы специальностей и направлений  
подготовки 10.00.00.

Курск 2024

УДК 004

Составители: Е.М. Терещенко, В.В. Чуйкова.

Рецензент

Доктор физико-математических наук, профессор кафедры  
Информационная безопасность *В.П. Добрица*

**Анализ работы техники VLAN:** методические указания к выполнению лабораторной и практической работ / Юго-Зап. гос. Ун-т; сост. Е.М. Терещенко, В.В. Чуйкова. Курск, 2024, 19 с.: ил. 7.; Библиогр.: с. 19.

Содержат сведения по настройке VLAN в среде GNS3. Указывается порядок выполнения лабораторной и практической работ, правила оформления и содержание отчета.

Методические указания по выполнению лабораторной и практической работ по дисциплинам «Безопасность систем и сетей передачи данных», «Сети и системы передачи информации», «Сети и системы передачи информации (специальные разделы)», «Администрирование вычислительных сетей», «Администрирование защищенных телекоммуникационных систем» для студентов укрупненной группы специальностей и направлений подготовки 10.00.00.

Текст печатается в авторской редакции

Подписано в печать 16.05.2024. Формат 60х84 1/16. Усл. печ. л. 1,10. Уч. –изд.л. 1,00. Тираж 30 экз. Заказ . Бесплатно. 378

Юго-Западный государственный университет.  
305040, г. Курск, ул. 50 лет Октября, 94.

## СОДЕРЖАНИЕ

|                                       |    |
|---------------------------------------|----|
| Введение .....                        | 4  |
| Цель работы .....                     | 4  |
| Порядок выполнения работы.....        | 5  |
| Содержание отчета .....               | 5  |
| Теоретическая часть .....             | 6  |
| Выполнение работы.....                | 8  |
| Задание.....                          | 19 |
| Контрольные вопросы.....              | 19 |
| Список информационных источников..... | 19 |

## **ВВЕДЕНИЕ**

VLAN (аббр. от англ. *Virtual Local Area Network*) — логическая («виртуальная») локальная компьютерная сеть, представляет собой группу хостов с общим набором требований, которые взаимодействуют так, как если бы они были подключены к широковещательному домену, независимо от их физического местонахождения. VLAN имеет те же свойства, что и физическая локальная сеть, но позволяет конечным станциям группироваться вместе, даже если они не находятся в одной физической сети. Такая реорганизация может быть сделана на основе программного обеспечения вместо физического перемещения устройств.

## **ЦЕЛЬ РАБОТЫ**

Цель лабораторной работы – исследовать принцип работы виртуальных локальных сетей VLAN (Virtual Local Area Network) на основе портов и стандарта IEEE 802.1q.

.

## **ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ**

1. Получить задание
2. Изучить теоретическую часть
3. Выполнить практическое задание
4. Написать вывод

## **СОДЕРЖАНИЕ ОТЧЕТА**

1. Титульный лист
2. Задание в соответствии с вариантом
3. Выполненное задание
4. Вывод

## ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Виртуальной локальной сетью называется логическая группа узлов сети, трафик которой, в том числе и широковещательный, на канальном уровне полностью изолирован от других узлов сети. Это означает, что передача кадров между разными виртуальными сетями на основании MAC-адреса невозможна независимо от типа адреса — уникального, группового или широковещательного. В то же время внутри виртуальной сети кадры передаются по технологии коммутации, то есть только на тот порт, который связан с адресом назначения кадра. Таким образом с помощью виртуальных сетей решается проблема распространения широковещательных кадров и вызываемых ими последствий, которые могут развиваться в широковещательные штормы и существенно снизить производительность сети.

VLAN обладают следующими преимуществами:

- гибкость внедрения. VLAN являются эффективным способом группировки сетевых пользователей в виртуальные рабочие группы, несмотря на их физическое размещение в сети;
- VLAN обеспечивают возможность контроля широковещательных сообщений, что увеличивает полосу пропускания, доступную для пользователя;
- VLAN позволяют повысить безопасность сети, определив с помощью фильтров, настроенных на коммутаторе или маршрутизаторе, политику взаимодействия пользователей из разных виртуальных сетей.

Типы VLAN:

В коммутаторах могут быть реализованы следующие типы VLAN:

- на основе портов;
- на основе стандарта IEEE 802.1Q;
- на основе стандарта IEEE 802.1ad (Q-in-Q VLAN);
- на основе портов и протоколов IEEE 802.1v;
- на основе MAC-адресов;
- асимметричные.

Также для сегментирования сети на канальном уровне модели OSI в коммутаторах могут использоваться другие функции, например функция *Traffic Segmentation*.

Построение VLAN на основе портов основано только на добавлении дополнительной информации к адресным таблицам коммутатора и не использует возможности встраивания информации о принадлежности к виртуальной сети в передаваемый кадр. Виртуальные локальные сети, построенные на основе стандарта IEEE 802.1Q, используют дополнительные поля кадра для хранения информации о принадлежности к VLAN при его перемещении по сети. С точки зрения удобства и гибкости настроек, VLAN стандарта IEEE 802.1Q является лучшим решением по сравнению с VLAN на основе портов. Его основные преимущества:

1. Гибкость и удобство в настройке и изменении — можно создавать необходимые комбинации VLAN как в пределах одного коммутатора, так и во всей сети, построенной на коммутаторах с поддержкой стандарта IEEE 802.1Q. Способность добавления тегов позволяет информации о VLAN распространяться через множество 802.1Q-совместимых коммутаторов по одному физическому соединению (*магистральному каналу, Trunk Link*);
2. Позволяет активизировать алгоритм связующего дерева (Spanning Tree) на всех портах и работать в обычном режиме. Протокол Spanning Tree оказывается весьма полезным для применения в крупных сетях, построенных на нескольких коммутаторах, и позволяет коммутаторам автоматически определять древовидную конфигурацию связей в сети при произвольном соединении портов между собой. Для нормальной работы коммутатора требуется отсутствие замкнутых маршрутов в сети. Эти маршруты могут создаваться администратором специально для образования резервных связей или же возникать случайным образом, что вполне возможно, если сеть имеет многочисленные связи, а кабельная система плохо структурирована или документирована. С помощью протокола Spanning Tree коммутаторы после построения схемы сети блокируют избыточные маршруты. Таким образом, автоматически предотвращается возникновение петель в сети;
3. Способность VLAN IEEE 802.1Q добавлять и извлекать теги из заголовков кадров позволяет использовать в сети коммутаторы и

сетевые устройства, которые не поддерживают стандарт IEEE 802.1Q;

4. Устройства разных производителей, поддерживающие стандарт, могут работать вместе, независимо от какого-либо фирменного решения;
5. Чтобы связать подсети на сетевом уровне, необходим маршрутизатор или коммутатор L3. Однако для более простых случаев, например, для организации доступа к серверу из различных VLAN, маршрутизатор не потребуется. Нужно включить порт коммутатора, к которому подключен сервер, во все подсети, а сетевой адаптер сервера должен поддерживать стандарт IEEE 802.1Q.

## ВЫПОЛНЕНИЕ РАБОТЫ

В программе GNS3 постройте следующую топологию:

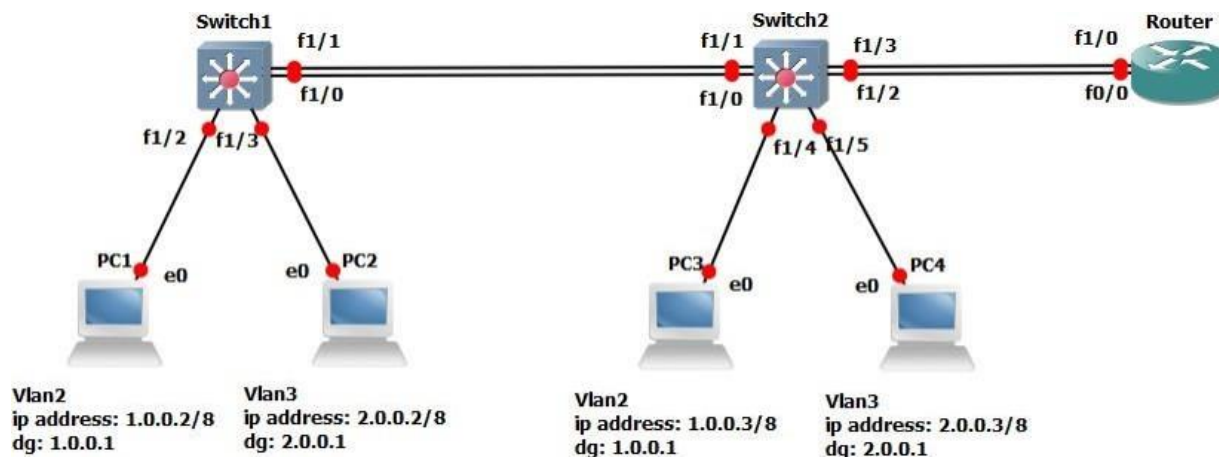


Рисунок 1 – Исследуемая сеть

В данном примере в качестве маршрутизатора используется Cisco 3640 с двумя сетевыми модулями NM-1FE-TX. для моделирования персональных компьютеров используются QEMU или VirtualBox машины с образом операционной системы linux-microcore 2.10 или tinyCore.

### Настройка персональных компьютеров

Настройте персональные компьютеры на работу с соответствующими IP адресами. используя следующие команды:

1. Вход в режим суперпользователя:

```
sudo su
```

2. Выставить интерфейсу eth0 ip-адрес и маску подсети: ipconfig eth[X] [IP\_adress] netmask [MASK]

3. Задать ip-адрес шлюза по умолчанию(default gateway): route add – net 0/0 gw[IP\_Gateway]

Пример для компьютера PC1:

```
tc@box: "$ sudo su
```

```
root@box: "# ifconfig eth0 1.0.0.2 netmask 255.0.0.0
```

```
root@box: "#route add -net 0/0 gw 1.0.0.1
```

С помощью следующих команд убедитесь в правильности введенной вами конфигурации:

```
root@box: "# ifconfig eth0 // для просмотра информации об
```

```
интерфейсе root@box: "# route // выводит таблицу
```

```
маршрутизации
```

После ввода команды route, в таблице должна существовать следующая запись:

| Destination | Gateway | Genmask |
|-------------|---------|---------|
| Default     | 1.0.0.1 | 0.0.0.0 |

Таким же образом настраиваются оставшиеся персональные компьютеры

## Настройка коммутаторов

Перед настройкой коммутаторов составьте таблицу, в которой указано какие номера виртуальных локальных сетей сопоставлены интерфейсам коммутаторов.

Таблица 1 – номера VLAN и связанные с ними интерфейсы

| Коммутатор | VLAN2            | VLAN3            |
|------------|------------------|------------------|
| Switch1    | f1/0, f1/2       | f1/1, f1/3       |
| Switch2    | f1/0, f1/4, f1/2 | f1/1, f1/3, f1/5 |

## Настройка коммутатора Switch1

```
Router>ena
```

```
Router#vlan database // команда для входа в режим создания VLAN
```

```
Router(vlan)#vlan 2 // создание VLAN 2
```

```
VLAN 2 added:
```

```

Name:
VLAN0002
Router(vlan)#vlan 3 name QWERTY      // создание VLAN 3 с именем
VLAN 3 added:                        // QWERTY
Name: QWERTY
Router(vlan)#apply // команда применяющая введенные настройки
APPLY completed.
Router(vlan)#exit // команда выхода из режима конфигурации VLAN
APPLY completed.Exiting.... Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname SW1
SW1(config)#int f1/0
SW1(config-if)#switchport access vlan 2 // сопоставление порта f1/0 с
VLAN 2
SW1(config-if)#int f1/1
SW1(config-if)#switchport access vlan 3 // сопоставление порта f1/1 с
VLAN 3
SW1(config-if)#int f1/2
SW1(config-if)#switchport access vlan 2 // сопоставление порта f1/2 с
VLAN 2
SW1(config-if)#int f1/3
SW1(config-if)#switchport access vlan 3 // сопоставление порта
f1/3 с VLAN 3

```

Для просмотра информации о VLAN введите следующую команду SW1#show vlan-switch

| VLAN     | Name                    | Status        | Ports                                                                                           |
|----------|-------------------------|---------------|-------------------------------------------------------------------------------------------------|
| ----     | -----                   | -----         | -----                                                                                           |
| 1        | default                 | active        | Fa1/4, Fa1/5, Fa1/6, Fa1/7<br>Fa1/8, Fa1/9, Fa1/10,<br>Fa1/11 Fa1/12, Fa1/13,<br>Fa1/14, Fa1/15 |
| <b>2</b> | <b>VLAN0002</b>         | <b>active</b> | <b>Fa1/0, Fa1/2</b>                                                                             |
| <b>3</b> | <b>QWERTY</b>           | <b>active</b> | <b>Fa1/1, Fa1/3</b>                                                                             |
| 1002     | <u>fddi</u> -default    | active        |                                                                                                 |
| 1003     | token-ring-default      | active        |                                                                                                 |
| 1004     | <u>fddinet</u> -default | active        |                                                                                                 |
| 1005     | <u>trnet</u> -default   | active        |                                                                                                 |

Рисунок 2 – Информация о VLAN 1

Обратите внимание на выделенные строки.

Для сохранения конфигурации введите следующую

команду:

```
SW1#copy run start
```

```
Destination filename [startup-config]? Y Building  
configuration...
```

```
[OK]
```

```
SW1#
```

## Настройка коммутатора Switch2

```
Router>ena
```

```
Router#vlan
```

```
database
```

```
Router(vlan)#vlan
```

```
2
```

```
VLAN 2 added:
```

```
  Name:
```

```
  VLAN0002
```

```
Router(vlan)#vlan 3
```

```
VLAN 3 added:Name: VLAN0003
```

```
Router(vlan)#app
```

```
ly APPLY
```

```
completed.
```

```
Router(vlan)#exit
```

```
APPLY
```

```
completed.
```

```
Exiting....
```

```
Router#config t
```

```
Enter configuration commands, one per line. End  
with CNTL/Z. Router(config)#hostname SW2
```

```
SW2(config)#int f1/1
```

```
SW2(config-if)#switchport
```

```
access vlan 3 SW2(config-if)#int  
f1/0
```

```
SW2(config-if)#switchport
```

```
access vlan 2 SW2(config-if)#int  
f1/2
```

```
SW2(config-if)#switchport
```

```

access vlan 2 SW2(config-if)#int
f1/4
SW2(config-if)#switchport
access vlan 2 SW2(config-if)#int
f1/3
SW2(config-if)#switchport
access vlan 3 SW2(config-if)#int
f1/5
SW2(config-if)#switchport
access vlan 3 SW2(config-if)#
*Mar 1 00:03:04.775: %LINEPROTO-5-UPDOWN: Line protocol on
Interface Vlan1, changed state to down
SW2#show vlan-switch brief

```

| VLAN | Name                      | Status | Ports                                                                          |
|------|---------------------------|--------|--------------------------------------------------------------------------------|
| ---- | -----                     | -----  | -----                                                                          |
| 1    | default                   | active | Fa1/6, Fa1/7, Fa1/8, Fa1/9<br>Fa1/10, Fa1/11, Fa1/12, Fa1/13<br>Fa1/14, Fa1/15 |
| 2    | VLAN0002                  | active | Fa1/0, Fa1/2, Fa1/4                                                            |
| 3    | VLAN0003                  | active | Fa1/1, Fa1/3, Fa1/5                                                            |
| 1002 | <u>fddi-default</u>       | active |                                                                                |
| 1003 | <u>token-ring-default</u> | active |                                                                                |
| 1004 | <u>fddinet-default</u>    | active |                                                                                |
| 1005 | <u>trnet-default</u>      | active |                                                                                |

Рисунок 3 – Информация о VLAN 2

После настройки коммутаторов с помощью утилиты Ping проверьте доступность хостов. Компьютеры одной VLAN должны взаимодействовать друг с другом, когда компьютеры из разных VLAN взаимодействовать не будут. Для остановки послыки пакетов нажмите Ctrl+C

### Настройка Маршрутизатора

Настройка маршрутизатора тривиальна и состоит в задании соответствующим интерфейсам правильных IP адресов.

```

Router>ena
Router#config t
Router(config)#int
f0/0
Router(config-if)#ip address 1.0.0.1
255.0.0.0 Router(config-if)#no shutd
Router(config-if)#int f1/0
Router(config-if)#ip address 2.0.0.1
255.0.0.0 Router(config-if)#no shutd

```

После настройки маршрутизатора снова проверьте доступность хостов с помощью утилиты Ping, если все сделано правильно, то сообщения ICMP эхо запросы должны доходить до любого интерфейса в любой сети.

### Составьте следующую топологию

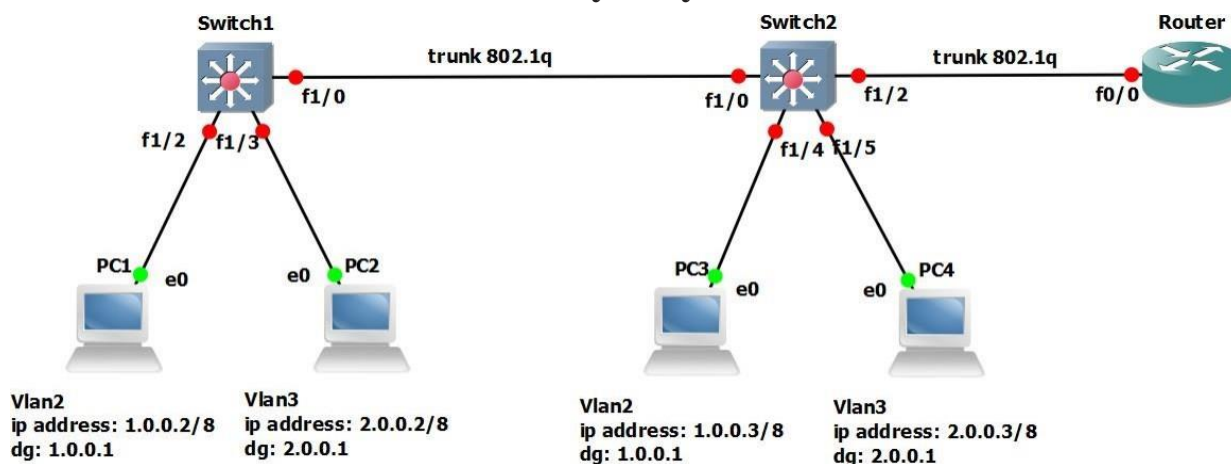


Рисунок 4 – Исследуемая сеть

Настройка компьютеров такая же, как и в предыдущей топологии

### Настройка коммутаторов

Настройка коммутаторов полностью идентична, за исключением настройки портов которые стали транкинговыми. Для коммутатора Switch1 это порт f1/0, а для коммутатора Switch2 это порты f1/0 и f1/2. В режиме конфигурации этих портов необходимо ввести команду: **switchport mode trunk**, то есть перевести порты в магистральный

режим.

## Настройка коммутатора Switch1

```
Router>ena
Router#vlan
database
Router(vlan)#vlan
2
VLAN 2 added:
  Name:
  VLAN0002
Router(vlan)#vlan 3 name
QWERTY
Router(vlan)#apply
Router(vlan)#exit
Router#config t
Router(config)#hostname
SW1 SW1(config)#int
f1/0
SW1(config-if)#switchport mode trunk
SW1(config-if)#int f1/2
SW1(config-if)#switchport
access vlan 2 SW1(config-if)#int
f1/3
SW1(config-if)#switchport
access vlan 3 SW1#show vlan-
switch
```

После ввода этой команды обратите внимание что порт который стал транкинговым не принадлежит ни одной VLAN

```
SW1#show int f1/0 switchport
Name: Fa1/0
Switchport: Enabled
Administrative Mode:
trunk Operational Mode:
trunk
Administrative Trunking Encapsulation:
dot1q Operational Trunking Encapsulation:
```

**dot1q** Negotiation of Trunking: Disabled  
 Access Mode VLAN: 0 ((Inactive))  
 Trunking Native Mode VLAN: 1  
 (default) Trunking VLANs Enabled:  
**ALL Trunking VLANs Active: 1-3**  
 Priority for untagged frames: 0  
 Override vlan tag priority:  
 FALSE Voice VLAN: none  
 Appliance trust: none

Внимательно посмотрите на выведенную информацию, из которой видно режим работы порта, тип инкапсуляции, и номера VLAN, которые разрешены на транкинговом порте. Сохраните конфигурацию.

```
SW1#copy run start
Destination filename [startup-config]?
y Building configuration...
[OK]
SW1#
```

## Настройка коммутатора Switch2

```
Router>ena
Router#vlan
database
Router(vlan)#vlan
2
Router(vlan)#vlan 3
Router(vlan)#apply
Router(vlan)#exit
Router#config t
Router(config)#hostname
SW2 SW2(config-if)#int
f1/0
SW2(config-if)# switchport mode trunk
SW2(config-if)#int f1/2
SW2(config-if)# switchport mode trunk
SW2(config-if)#int f1/4
SW2(config-if)#switchport
access vlan 2 SW2(config-if)#int
f1/5
```

```
SW2(config-if)#switchport
access vlan 3 SW2(config-if)#
SW2#show vlan-switch
brief SW1#show int f1/0
switchport
```

SW1#show mac-address-table //просмотр таблицы коммутации

| Destination Address | Address Type | VLAN | Destination Port |
|---------------------|--------------|------|------------------|
| -----               | -----        | ---- | -----            |
| c403.0f04.0000      | Self         | 1    | Vlan1            |
| cc05.1674.0000      | Dynamic      | 2    | FastEthernet1/0  |
| cc05.1674.0000      | Dynamic      | 3    | FastEthernet1/0  |
| 00aa.0078.2600      | Dynamic      | 3    | FastEthernet1/3  |
| 00aa.0018.4f00      | Dynamic      | 2    | FastEthernet1/2  |
| 00aa.0026.f900      | Dynamic      | 3    | FastEthernet1/0  |
| 00aa.009a.9400      | Dynamic      | 2    | FastEthernet1/0  |

Рисунок 5 – Просмотр таблицы коммутации

Обратите внимание на столбец VLAN.

После настройки коммутаторов с помощью утилиты Ping проверьте доступность хостов. Компьютеры одной VLAN должны взаимодействовать друг с другом, когда компьютеры из разных VLAN взаимодействовать не будут.

### Настройка маршрутизатора

Для работы с VLAN на маршрутизаторе необходимо сконфигурировать столько подинтерфейсов сколько VLAN существует, затем указать тип инкапсуляции для каждого подинтерфейса и задать ip адрес каждому подинтерфейсу маршрутизатора. Перед созданием подинтерфейса необходимо командой no shutdown включить основной интерфейс.

Например, для создания подинтерфейса 43 на интерфейсе f0/0, необходимо в режиме конфигурации ввести interface f0/0.43. Номер интерфейса ни как ни связан с номером VLAN, но для удобства, как правило, номер подинтерфейса устанавливают равным номеру VLAN. то есть интерфейс f0/0.43 может принадлежать VLAN 50 и на оборот, интерфейс f0/0.50 может принадлежать VLAN 43, но одновременно

интерфейс f0/0.43 не может принадлежать сразу двум VLAN

Указание типа инкапсуляции: `encapsulation dot1Q [номер VLAN]`

### Пример конфигурации маршрутизатора

```
Router>ena
Router#config t
Router(config)#int f0/0    // заходим на основной интерфейс
Router(config-if)#no shutdown // включаем его
Router(config-if)#int f0/0.2    // создаем подинтерфейс номер 2
Router(config-subif)#encapsulation dot1Q 2    // задаем тип инкапсуляции
dot1q, а 2 указывает номер VLAN, к которой принадлежит данный
подинтерфейс
Router(config-subif)#ip address 1.0.0.1 255.0.0.0    //
устанавливаем ip адрес
Router(config-subif)#exit    // выход из режима
конфигурации подинтерфейса
Router(config)#int f0/0.3
Router(config-subif)#encapsulation
dot1Q 3 Router(config-subif)#ip
address 2.0.0.1 255.0.0.0
Router(config-subif)#^Z
Router#show
Router#show ip
route
C 1.0.0.0/8 is directly connected, FastEthernet0/0.2
C 2.0.0.0/8 is directly connected, FastEthernet0/0.3
```

Обратите внимание на номера интерфейсов, к которым подключены сети 1.0.0.0 и 2.0.0.0. После настройки маршрутизатора с помощью утилиты `ping` проверьте связь между компьютерами, находящимися в разных VLAN

### Анализ форматов кадров

Во время или до запуска утилиты `Ping` между хостами PC1 и PC2 запустите сетевой анализатор протокола `Wireshark` и посмотрите перехваченные пакеты между коммутаторами и между хостом и коммутатором, обратите внимание на разные форматы кадров. Точки захвата указаны стрелками на рисунке 6. Для остановки посылки

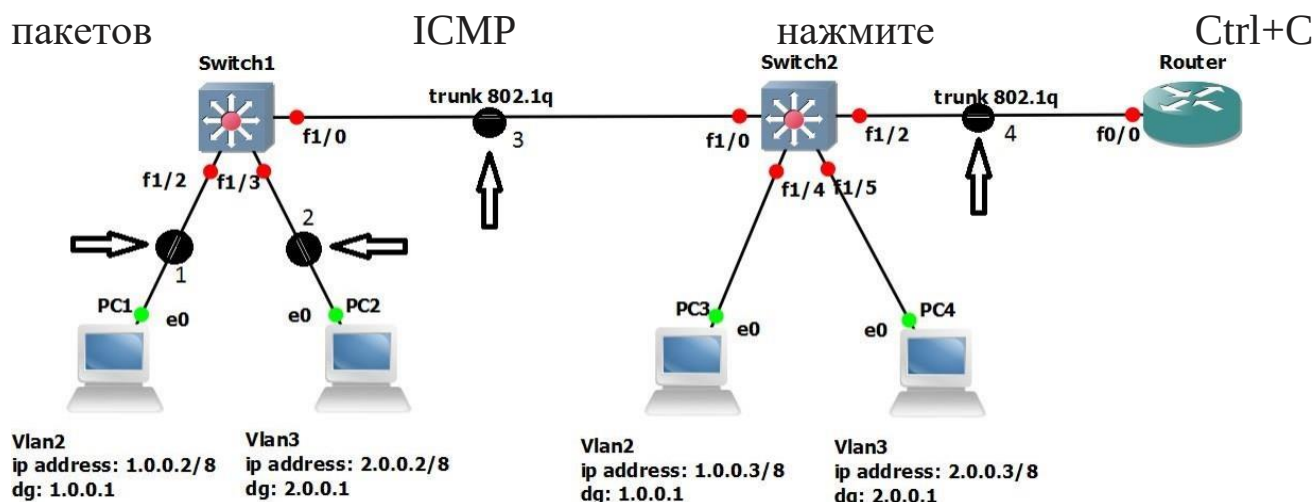


Рисунок 6 – Места захвата пакетов

На рисунке 7 изображено окно сетевого анализатора пакетов, в котором отображен формат пакета ICMP (столбец Protocol, цифра 3) echo request (столбец info, цифра 4) отправленного с адреса 1.0.0.2 (столбец source цифра 1), на адрес 2.0.0.1 (столбец destination, цифра 2). Детализация захваченного пакета показана в окне ниже, а заголовок Ethernet (цифра 5) и заголовок IP (цифра 6) выделены.

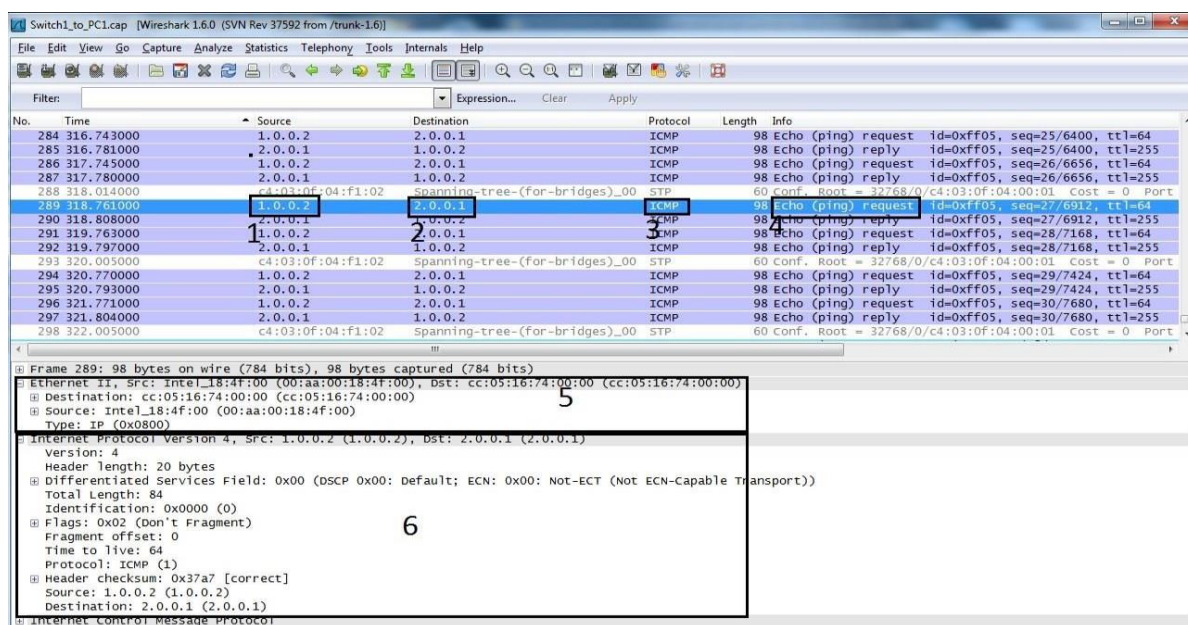


Рисунок 7 – Пример захвата пакета ICMP echo request

Пакет отправленный с компьютера PC1 до PC2, пройдет следующий маршрут:

PC1(int e0) =1=> (int f1/2)Switch1(int f1/0) =2=> (int f1/0)Switch2 (int f1/2)=3=> (int f0/0.2) Router (int f0/0.3) =4=> (int f1/2)Switch2 (int f1/0) =5=> (int f1/0)Switch1(int f1/2) =6=> (int e0)PC2

## ЗАДАНИЕ

1. Составьте исходную топологию;
2. Получите листинг команд конфигурации, либо конфигурационные файлы;
3. Сделайте захваты следующих пакетов с помощью программы Wireshark: Arp request, Arp reply, ICMP echo request, ICMP echo reply, поясните значение всех полей данных пакетов.

## КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Формат метки VLAN согласно стандарту IEEE802.1 q;
2. Объясните принцип коммутации пакетов с учетом техники VLAN;
3. Дайте определение виртуальной локальной сети;
4. В чем разница между транкинговым портом и портом магистральным;
5. Какое максимальное количество VLAN можно организовать согласно стандарту IEEE 802.1q;
6. Поясните процедуру передачи пакета ICMP echo request от PC1 к PC2.

## СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов [Текст]/ В.Г. Олифер, Н.А. Олифер, 4-е изд.: СПб.: Питер, 2006 -958 с.
2. Сергеев А.Н. Основы локальных компьютерных сетей [Текст]/ Сергеев А.Н.: Изд.: «Лань», 2016. 184 с.
3. Электронный каталог Documentation [Электронный ресурс]: / Internet. - <http://www.gns3.net/documentation/> (10.10.17).

**МИНОБРНАУКИ РОССИИ**  
Федеральное государственное бюджетное  
образовательное учреждение высшего  
образования «Юго-Западный государственный  
университет»  
(ЮЗГУ)  
Кафедра информационной безопасности



**ПОНЯТИЕ СЕТЕВЫХ ПРОТОКОЛОВ. ПРОТОКОЛЫ TCP, UDP**

Методические указания по выполнению лабораторной и практической  
работы для студентов укрупненной группы специальностей и направлений  
подготовки 10.00.00.

Курск 2024

УДК 004

Составители: Е.М. Терещенко, В.В. Чуйкова.

Рецензент

Доктор физико-математических наук, профессор кафедры  
Информационная безопасность *В.П. Добрица*

**Понятие сетевых протоколов. Протоколы TCP, UDP:**  
методические указания к выполнению лабораторной и практической работ / Юго-Зап. гос. Ун-т; сост.Е .М.Терещенко, В .В.Чуйкова. Курск, 2024, 23 с.: ил. 11.; Библиогр.: с. 23.

Содержат сведения по сетевым и транспортным протоколам уровней модели OSI. Указывается порядок выполнения лабораторной и практической работ, правила оформления, содержание отчета.

Методические указания по выполнению лабораторной и практической работ по дисциплинам «Безопасность систем и сетей передачи данных», «Сети и системы передачи информации», «Сети и системы передачи информации (специальные разделы)», «Администрирование вычислительных сетей», «Администрирование защищенных телекоммуникационных систем» для студентов укрупненной группы специальностей и направлений подготовки 10.00.00.

Текст печатается в авторской редакции

Подписано в печать 16.05.2024. Формат 60х84 1/16. Усл. печ.  
л. 1,34. Уч. –изд.л. 1,21. Тираж 30 экз. Заказ . Бесплатно. 374

Юго-Западный государственный университет.  
305040, г. Курск, ул. 50 лет Октября, 94.

## СОДЕРЖАНИЕ

|                                       |    |
|---------------------------------------|----|
| Введение .....                        | 4  |
| Цель работы.....                      | 5  |
| Порядок выполнения работы.....        | 5  |
| Содержание отчета .....               | 5  |
| Теоретическая часть .....             | 6  |
| Выполнение работы.....                | 19 |
| Задание.....                          | 23 |
| Контрольные вопросы.....              | 23 |
| Список информационных источников..... | 23 |

## ВВЕДЕНИЕ

Обмен данными, ориентированный на соединения, может обеспечить надежную связь, для обеспечения которой протокол уровня 4 посылает подтверждения о получении данных и запрашивает повторную передачу, если данные не получены или искажены. Транспортный протокол TCP использует именно такую надежную связь. TCP используется в таких прикладных протоколах, как HTTP, FTP, SMTP и Telnet.

Протокол TCP требует, чтобы перед отправкой сообщения было установлено соединение. Серверное приложение должно выполнить так называемое пассивное открытие (passive open), чтобы создать соединение с известным номером порта, и, вместо того чтобы отправлять вызов в сеть, сервер переходит в ожидание поступления входящих запросов. Клиентское приложение должно выполнить активное открытие (active open), отправив серверному приложению синхронизирующий порядковый номер (SYN), идентифицирующий соединение. Клиентское приложение может использовать динамический номер порта в качестве локального порта.

В отличие от TCP, другой транспортный протокол UDP — очень быстрый протокол, поскольку в нем определен самый минимальный механизм, необходимый для передачи данных. Конечно, он имеет некоторые недостатки. Сообщения поступают в любом порядке, и то, которое отправлено первым, может быть получено последним. Доставка сообщений UDP вовсе не гарантируется, сообщение может потеряться, или могут быть получены две копии одного и того же сообщения. Последний случай возникает, если для отправки сообщений в один адрес использовались два разных маршрута.

UDP не требует открывать соединение, и данные могут быть отправлены сразу же, как только они подготовлены. UDP не отправляет подтверждающие сообщения, поэтому данные могут быть получены или потеряны. Если при использовании UDP требуется надежная передача данных, ее следует реализовать в протоколе более высокого уровня.

## **ЦЕЛЬ РАБОТЫ**

Цель лабораторной работы – изучить понятие и назначение сетевых протоколов, изучить протоколы сетевого и транспортного уровней: IP, TCP, UDP.

## **ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ**

1. Получить задание
2. Изучить теоретическую часть
3. Выполнить практическое задание
4. Написать вывод

## **СОДЕРЖАНИЕ ОТЧЕТА**

1. Титульный лист
2. Задание в соответствии с вариантом
3. Выполненное задание
4. Вывод

## ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

### Понятие протоколов

Сетевой протокол в компьютерных сетях – набор правил для специфического типа связи.

Разные протоколы зачастую описывают лишь разные стороны одного типа связи; взятые вместе, они образуют стек протоколов. Названия «протокол» и «стек протоколов» также указывают на программное обеспечение, которым реализуется протокол.

Новые протоколы для Интернета определяются IETF (Internet Engineering Task Force – проблемная группа проектирования Internet), а прочие протоколы – IEEE или ISO. ITU-T (International Telecommunication Union, ITU) занимается телекоммуникационными протоколами и форматами.

### Эталонная модель OSI

Чтобы помочь поставщикам в стандартизации и интеграции их сетевого программного обеспечения, международная организация по стандартизации (ISO) определила программную модель пересылки сообщений между компьютерами. Эта модель получила название *эталонной модели OSI* (Open Systems Interconnection). В ней определено семь уровней программного обеспечения (рис. 1).

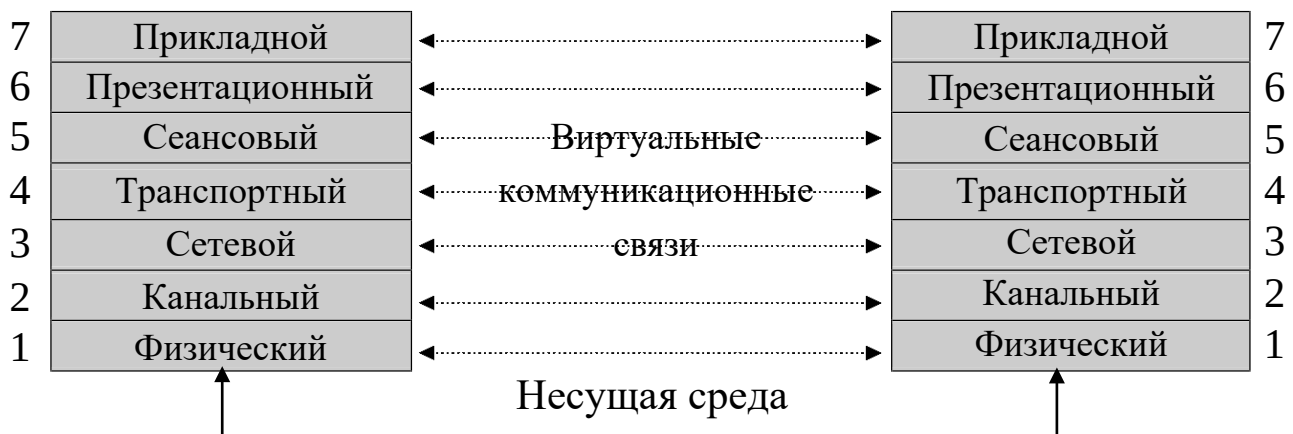


Рисунок 1 - Эталонная модель OSI.

Эталонная модель OSI – идеал, точно реализованный лишь в очень немногих системах, но часто используемый при объяснении основных принципов работы сети. Каждый уровень на одной из машин считает, что

он взаимодействует с тем же уровнем на другой машине. На данном уровне обе машины «разговаривают» на одном языке, или протоколе. Но в действительности сетевой запрос должен сначала пройти до самого нижнего уровня на первой машине, затем он передается по несущей среде и уже на второй машине вновь поднимается до уровня, который его поймет и обработает.

Задача каждого уровня в том, чтобы предоставить сервисы более высоким уровням и скрывать от них конкретную реализацию этих сервисов.

**Прикладной уровень (Application layer).** Верхний (7-й) уровень модели, обеспечивает взаимодействие сети и пользователя. Уровень разрешает доступ к сетевым службам приложениям пользователя, таким как обработчик запросов к базам данных, доступ к файлам, пересылке электронной почты. Также отвечает за передачу служебной информации, предоставляет приложениям информацию об ошибках и формирует запросы к уровню представления.

**Уровень представления (Presentation layer).** Этот уровень отвечает за преобразование протоколов и кодирование/декодирование данных. Запросы приложений, полученные с уровня приложений, он преобразует в формат для передачи по сети, а полученные из сети данные преобразует в формат, понятный приложениям. На этом уровне может осуществляться сжатие/распаковка или кодирование/раскодирование данных, а также перенаправление запросов другому сетевому ресурсу, если они не могут быть обработаны локально.

**Сеансовый уровень (Session layer).** Отвечает за поддержание сеанса связи, позволяя приложениям взаимодействовать между собой длительное время. Уровень управляет созданием/завершением сеанса, обменом информацией, синхронизации задач, определением права на передачу данных и поддержание сеанса в периоды неактивности приложений. Синхронизация передачи обеспечивается помещением в поток данных контрольных точек, начиная с которых возобновляется процесс при нарушении взаимодействия.

**Транспортный уровень (Transport layer).** 4-й уровень модели, предназначен для доставки данных без ошибок, потерь и дублирования в той последовательности, как они были переданы. При этом неважно какие данные передаются, откуда и куда, то есть он предоставляет сам механизм передачи. Блоки данных он разделяет на фрагменты, размер которых зависит от протокола, короткие объединяет в один, длинные

разбивает. Протоколы этого уровня предназначены для взаимодействия типа точка-точка.

**Сетевой уровень (Network layer).** 3-й уровень сетевой модели OSI, предназначен для определения пути передачи данных. Отвечает за трансляцию логических адресов и имён в физические, определение кратчайших маршрутов, коммутация и маршрутизация пакетов, отслеживание неполадок и заторов в сети. На этом уровне работает такое сетевое устройство, как маршрутизатор.

**Канальный уровень (Data Link layer).** Этот уровень предназначен для обеспечения взаимодействия сетей на физическом уровне и контроле за ошибками, которые могут возникнуть. Полученные данные от физического уровня он упаковывает в кадры данных, проверяет на целостность, если нужно исправляет ошибки и отправляет на сетевой уровень. Канальный уровень может взаимодействовать с одним или несколькими физическими уровнями, контролируя и управляя этим взаимодействием. Спецификация IEEE 802 разделяет этот уровень на 2 подуровня – MAC (Media Access Control) регулирует доступ к разделяемой физической среде, и LLC (Logical Link Control) обеспечивает обслуживание сетевого уровня. На этом уровне работают коммутаторы, мосты и сетевые адаптеры.

В программировании этот уровень представляет драйвер сетевой платы, в операционных системах имеется программный интерфейс взаимодействия канального и сетевого уровня между собой, это не новый уровень, а просто реализация модели для конкретной ОС. Примеры таких интерфейсов: ODI, NDIS.

**Физический уровень (Physical layer).** Самый нижний уровень модели, предназначен непосредственно для передачи потока данных. Осуществляет передачу электрических или оптических сигналов в кабель и соответственно их приём и преобразование в биты данных в соответствии с методами кодирования цифровых сигналов. Другими словами осуществляет интерфейс между сетевым носителем и сетевым устройством. На этом уровне работают концентраторы и повторители (ретрансляторы) сигнала.

### **Инкапсуляция и обработка пакетов**

При продвижении пакета данных по уровням сверху вниз каждый новый уровень добавляет к пакету свою служебную информацию в виде заголовка и, возможно, трейлера (информации, помещаемой в конец сообщения). Эта операция называется инкапсуляцией данных верхнего

уровня в пакете нижнего уровня. Служебная информация предназначена для объекта того же уровня на удаленном компьютере, ее формат и интерпретация определяются протоколом данного уровня.

Разумеется, данные, приходящие с верхнего уровня, могут на самом деле представлять собой пакеты с уже инкапсулированными данными еще более верхнего уровня.

С другой стороны, при получении пакета от нижнего уровня он разделяется на заголовок (трейлер) и данные. Служебная информация из заголовка (трейлера) анализируется и в соответствии с ней данные, возможно, направляются одному из объектов верхнего уровня. Тот в свою очередь рассматривает эти данные как пакет со своей служебной информацией и данными для еще более верхнего уровня, и процедура повторяется, пока пользовательские данные, очищенные от всей служебной информации, не достигнут прикладного процесса.

Возможно, что пакет данных не будет доведен до самого верхнего уровня, например, в случае, если данный компьютер представляет собой промежуточную станцию на пути между отправителем и получателем. В этом случае объект соответствующего уровня при анализе служебной информации заметит, что пакет на этом уровне адресован не ему (хотя с точки зрения нижележащих уровней он был адресован именно этому компьютеру). Тогда объект выполнит необходимые действия для перенаправления пакета к месту назначения или возврата отправителю с сообщением об ошибке, но в любом случае не будет продвигать данные на верхний уровень.

### **Применимость модели OSI**

Хотя модель OSI полезна как основа для обсуждения сетевых архитектур и реализаций, ее нельзя рассматривать как готовый чертеж для создания любой сетевой архитектуры. Не следует также думать, что размещение некоторой функции на уровне N в этой модели означает, что только здесь наилучшее для нее место.

Модель OSI имеет множество недостатков. Хотя, в конечном итоге, были созданы работающие реализации, протоколы OSI на сегодняшний день утратили актуальность. Основные проблемы этой модели в том, что, во-первых, распределение функций между уровнями произвольно и не всегда очевидно, во-вторых, она была спроектирована (комитетом) без готовой реализации.

Другая проблема модели OSI – это сложность и неэффективность. Некоторые функции выполняются сразу на нескольких уровнях. Так,

обнаружение и исправление ошибок происходит на большинстве уровней.

Наконец, выбор именно семи уровней продиктован, скорее, политическими, а не техническими причинами. В действительности, сеансовый уровень и уровень представления редко встречаются в реально работающих сетях.

### Стек протоколов TCP/IP

TCP/IP – собирательное название для набора (стека) сетевых протоколов разных уровней, используемых в Интернет. Особенности TCP/IP:

- открытые стандарты протоколов, разрабатываемые независимо от программного и аппаратного обеспечения;
- независимость от физической среды передачи;
- система уникальной адресации;
- стандартизированные протоколы высокого уровня для распространенных пользовательских сервисов

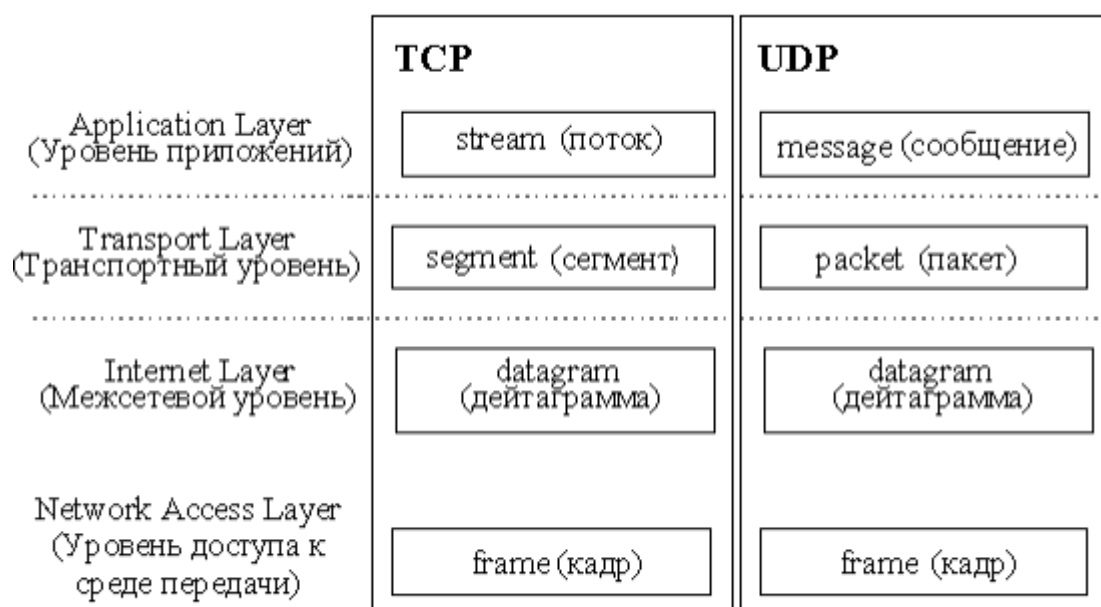


Рисунок 2 - Уровни стека протоколов TCP/IP.

Стек протоколов TCP/IP делится на 4 уровня: прикладной (application), транспортный (transport), межсетевой (internet) и уровень доступа к среде передачи (network access). Термины, применяемые для обозначения блока передаваемых данных, различны при использовании разных протоколов транспортного уровня – TCP и UDP, поэтому на

рисунке 2 изображено два стека. Как и в модели OSI, данные более верхних уровней инкапсулируются в пакеты нижних уровней (см. рис. 3).

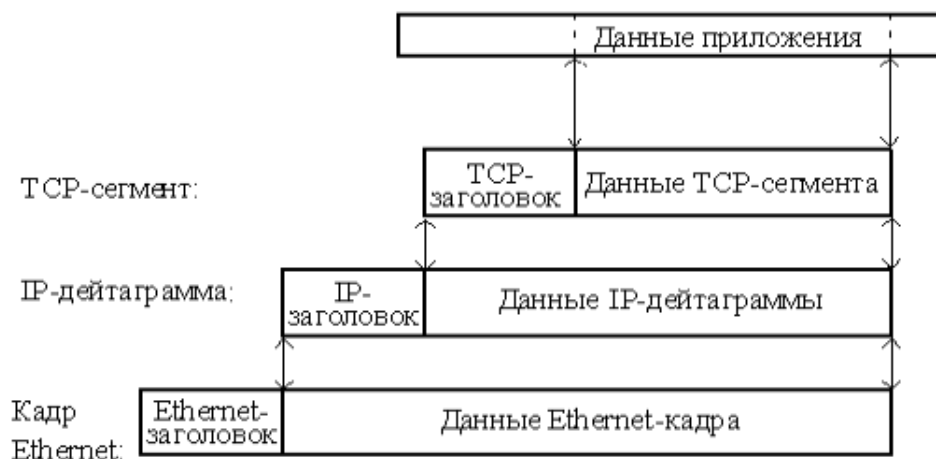


Рисунок 3 - Пример инкапсуляции пакетов в стеке



Рисунок 4 - Соотношение уровней стеков OSI и TCP/IP

**Уровень приложений.** Приложения, работающие со стеком TCP/IP, могут также выполнять функции уровней представления и частично сеансового модели OSI; например, преобразование данных к внешнему представлению, группировка данных для передачи и т.п.

Распространенными примерами приложений являются программы telnet, ftp, HTTP-серверы и клиенты, программы работы с электронной почтой и др.

Для пересылки данных другому приложению, приложение обращается к тому или иному модулю транспортного уровня.

**Транспортный уровень.** Протоколы транспортного уровня обеспечивают прозрачную (сквозную) доставку данных (end-to-end delivery service) между двумя прикладными процессами. Процесс,

получающий или отправляющий данные с помощью транспортного уровня, идентифицируется на этом уровне номером, который называется номером порта. Таким образом, роль адреса отправителя и получателя на транспортном уровне выполняет номер *порта* (см. далее).

Анализируя заголовок своего пакета, полученного от межсетевого уровня, транспортный модуль определяет по номеру порта получателя, какому из прикладных процессов направлены данные, и передает эти данные соответствующему прикладному процессу (возможно, после проверки их на наличие ошибок и т.п.). Номера портов получателя и отправителя записываются в заголовок транспортным модулем, отправляющим данные; заголовок транспортного уровня содержит также и другую служебную информацию; формат заголовка зависит от используемого транспортного протокола.

На транспортном уровне работают два основных протокола: UDP и TCP.

**TCP** (Transmission Control Protocol – протокол контроля передачи, *RFC 793*) – это транспортный механизм, предоставляющий поток данных, с предварительной установкой соединения, за счёт этого дающий уверенность в безошибочности получаемых данных, осуществляет повторный запрос данных в случае потери пакетов и устраняет дублирование при получении двух копий одного пакета. Естественно, что в общем случае данные не могут быть гарантировано доставлены до адресата; в таком случае клиентский процесс получает об этом уведомление.

Данными для TCP является не интерпретируемая протоколом последовательность пользовательских октетов, разбиваемая для передачи по частям. Каждая часть передается в отдельном TCP-сегменте. Для продвижения сегмента по сети между компьютером-отправителем и компьютером-получателем модуль TCP пользуется сервисом межсетевого уровня (вызывает модуль IP). Протокол TCP гарантирует, что приложение получит данные точно в такой же последовательности, в какой они были отправлены, и без потерь.

**UDP** (User Datagram Protocol, протокол пользовательских дейтаграмм, *RFC 768*) фактически не выполняет каких-либо особых функций дополнительно к функциям межсетевого уровня (протокола IP см. далее). Протокол UDP используется либо при пересылке коротких сообщений, когда накладные расходы на установление сеанса и проверку успешной доставки данных оказываются выше расходов на повторную (в случае неудачи) пересылку сообщения, либо в том случае, когда сама

организация процесса-приложения обеспечивает установление соединения и проверку доставки пакетов.

Пользовательские данные, поступившие от прикладного уровня, предваряются UDP-заголовком, и сформированный таким образом UDP-пакет отправляется на межсетевой уровень.

**Межсетевой уровень и протокол IP.** Основным протоколом этого уровня является протокол IP (Internet Protocol, *RFC 791*).

Протокол IP доставляет блоки данных, называемых дейтаграммами, от одного сетевого узла к другому.

В современной сети Интернет используется IP четвёртой версии, также известный как IPv4. В протоколе IP этой версии каждому узлу сети ставится в соответствие IP-адрес длиной 4 октета (иногда говорят «байта», подразумевая распространённый восьмибитовый минимальный адресуемый фрагмент памяти ЭВМ). Более подробно об IP-адресах протокола 4-й версии можно прочитать в предыдущей лабораторной работе.

В настоящее время вводится в эксплуатацию шестая версия протокола — IPv6, которая позволяет адресовать значительно большее количество узлов, чем IPv4. Эта версия отличается повышенной разрядностью адреса, встроенной возможностью шифрования и некоторыми другими особенностями. Переход с IPv4 на IPv6 связан с трудоёмкой работой операторов связи и производителей программного обеспечения и не может быть выполнен одномоментно. На начало 2007 года в Интернете присутствовало около 760 сетей, работающих по протоколу IPv6. Для сравнения, на то же время в адресном пространстве IPv4 присутствовало более 203 тысяч сетей, но в IPv6 сети гораздо более крупные, нежели в IPv4.

Данные для IP дейтаграммы передаются IP-модулю транспортным уровнем. IP-модуль предваряет эти данные заголовком, содержащим IP-адреса отправителя и получателя и другую служебную информацию, и сформированная таким образом дейтаграмма передается на уровень доступа к среде передачи (например, одному из физических интерфейсов) для отправки по каналу передачи данных.

Не все сетевые узлы могут непосредственно связаться друг с другом; часто для того, чтобы передать дейтаграмму по назначению, требуется направить ее через один или несколько промежуточных узлов по тому или иному маршруту. Задача определения маршрута для каждой дейтаграммы решается протоколом IP.

Когда модуль IP получает дейтаграмму с нижнего уровня, он проверяет IP-адрес назначения. Если дейтаграмма адресована данному компьютеру, то данные из нее передаются на обработку модулю вышестоящего уровня (какому конкретно – указано в заголовке дейтаграммы). Если же адрес назначения дейтаграммы – чужой, то модуль IP может принять два решения: первое – уничтожить дейтаграмму, второе – отправить ее дальше к месту назначения, определив маршрут следования – так поступают промежуточные станции – маршрутизаторы.

Также может потребоваться, на границе сетей с различными характеристиками, разбить дейтаграмму на фрагменты, а потом собрать в единое целое на компьютере-получателе. Это тоже задача протокола IP.

Если модуль IP по какой-либо причине не может доставить дейтаграмму, она уничтожается. При этом модуль IP может отправить компьютеру-источнику этой дейтаграммы уведомление об ошибке; такие уведомления отправляются с помощью протокола ICMP, являющегося неотъемлемой частью модуля IP. Более никаких средств контроля корректности данных, подтверждения их доставки, обеспечения правильного порядка следования дейтаграмм, предварительного установления соединения между компьютерами протокол IP не имеет. Эта задача возложена на транспортный уровень.

Более подробную информацию об IP-адресах можно найти в предыдущей лабораторной работе.

**Уровень доступа к среде передачи.** Функции этого уровня:

- отображение IP-адресов в физические адреса сети (MAC-адреса, например, Ethernet-адрес в случае сети Ethernet). Эту функцию выполняет протокол ARP;
- инкапсуляция IP-дейтаграмм в кадры для передачи по физическому каналу и извлечение дейтаграмм из кадров. При этом не требуется какого-либо контроля безошибочности передачи (хотя он может и присутствовать), поскольку в стеке TCP/IP такой контроль возложен на транспортный уровень или на само приложение. В заголовке кадров указывается точка доступа к сервису (SAP, Service Access Point) - поле, содержащее код протокола межсетевого уровня, которому следует передать содержимое кадра (в нашем случае это протокол IP);
- определение метода доступа к среде передачи - то есть способа, с помощью которого компьютер устанавливает свое право на произведение передачи данных (передача токена, опрос

компьютеров, множественный доступ с детектированием коллизий и т.п.).

- определение представления данных в физической среде;
- пересылка и прием кадра.

Стек ТСР/ІР не подразумевает использования каких-либо определенных протоколов уровня доступа к среде передачи и физических сред передачи данных. От уровня доступа к среде передачи требуется наличие интерфейса с модулем ІР, обеспечивающего передачу дейтаграммы между уровнями. Также требуется обеспечить преобразование ІР-адреса узла сети, на который передается дейтаграмма, в МАС-адрес. Часто в качестве уровня доступа к среде передачи могут выступать целые протокольные стеки, тогда говорят об ІР поверх АТМ, ІР поверх ІРХ, ІР поверх Х.25 и т.п.

Обобщенная модель взаимодействия узлов на базе протоколов ТСР/ІР представлена на рис 5.

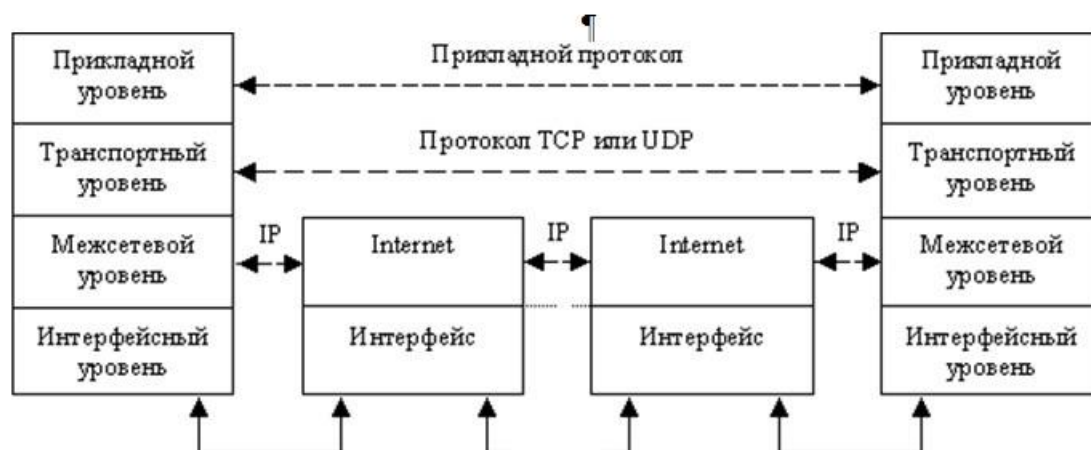


Рисунок 5 - Модель взаимодействия стеков ТСР/ІР

## Понятие сетевых портов и сокетов

Как уже было сказано, основные прикладные сетевые сервисы используют средства транспортного уровня для взаимодействия.

Любые 2 сетевых процесса могут идентифицировать друг друга при помощи 3-х компонент: ір-адрес, протокол(ТСР/UDP), порт. Часто данные компоненты носят название **сокетами**. Сокеты – это название программного интерфейса для обеспечения информационного обмена между процессами. Т.е. для прикладных сетевых процессов взаимодействие осуществляется через сокеты. Более детально сокеты мы

будем изучать на последующих занятиях. Сейчас же рассмотрим понятие портов более подробно.

**Порт** – параметр протоколов TCP и UDP, определяющий пункт назначения для данных, принимаемых по сети. Порту сопоставляется номер от 1 до 65535, позволяющие различным программам, выполняемым на одном хосте, получать данные независимо друг от друга. В этом случае каждая из них обрабатывает данные, поступающие на определённый порт (иногда говорят, что программа «слушает» на том или ином порту).

Согласно IP, в каждом пакете присутствуют IP адрес узла-источника и IP адрес узла-назначения. В TCP/UDP пакетах дополнительно указываются порт источника и порт назначения. Узел назначения, получив пакет, смотрит на порт назначения и передает пакет соответствующему у себя приложению. Использование портов позволяет независимо использовать TCP/UDP протокол сразу многим приложениям на одном и том же компьютере.

Для сетевых приложений нотация указания порта следующая: «ip:port». Например, <http://web-service.org:8888>

Пояснение понятия портов представлено на рис. 6. На самом деле сетевой порт – это всего лишь числовой параметр в сетевом пакете протоколов TCP и UDP. Такие понятия как «открыть порт» означают что пакеты адресованные на данный порт будут приниматься на обработку.

Порты из диапазона 1-1024 являются привилегированными. Называются они так, потому что для их открытия (и, соответственно, запуска соответствующих сетевых сервисов) на большинстве ОС требуются права системного администратора. Большая часть привилегированных портов распределена для общеупотребительных сетевых протоколов. В табл. 1 перечислены некоторые протоколы и порты, за которыми они закреплены. Данные порты являются портами по умолчанию для соответствующих служб и чаще всего не перенастраиваются.

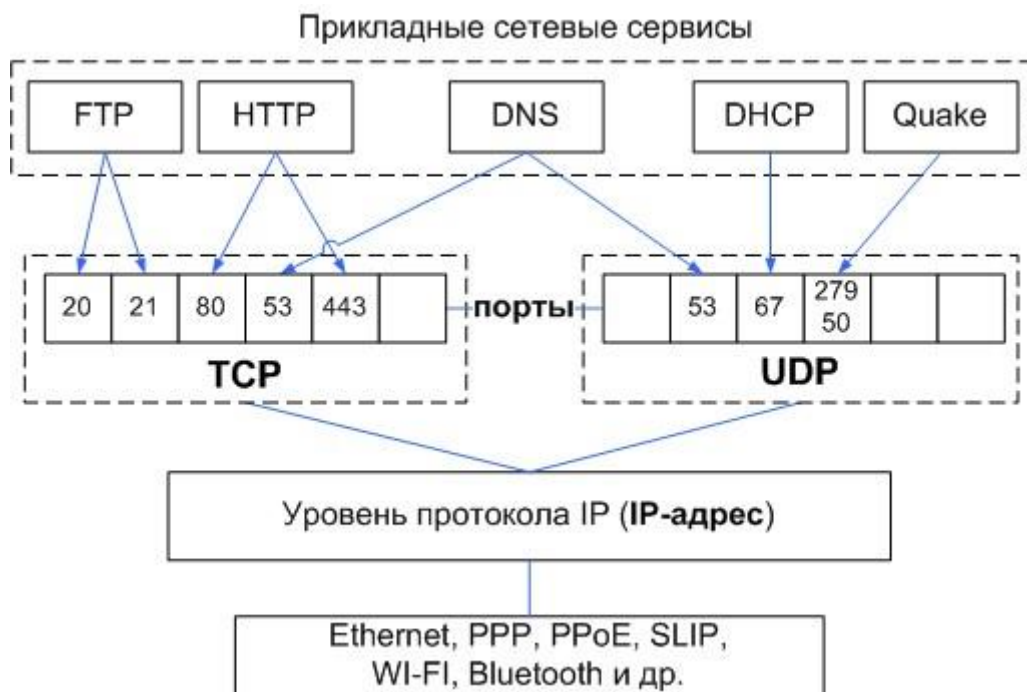


Рисунок 6. - Компоненты сокетов

Таблица 1 - Примеры некоторых стандартных сетевых портов

| Порт/протокол | Сервис   | Описание                                                                                                                                  |
|---------------|----------|-------------------------------------------------------------------------------------------------------------------------------------------|
| 20/TCP        | ftp-data | Порт данных FTP                                                                                                                           |
| 21/TCP        | ftp      | Порт протокола передачи файлов (File Transfer Protocol, FTP); иногда используется протоколом файловой службы (File Service Protocol, FSP) |
| 22/TCP        | ssh      | Служба Безопасной ОБолочки (Secure SHell, SSH)                                                                                            |
| 23/TCP        | telnet   | Служба Telnet                                                                                                                             |
| 25/TCP        | smtp     | Протокол простой передачи почты (Simple Mail Transfer Protocol, SMTP)                                                                     |
| 53/(UDP,TCP)  | domain   | Службы доменных имён (такие как BIND)                                                                                                     |

|         |        |                                                                                                                     |
|---------|--------|---------------------------------------------------------------------------------------------------------------------|
| 80/TCP  | http   | Протокол передачи гипертекста (HyperText Transfer Protocol, HTTP) для служб всемирной паутины (World Wide Web, WWW) |
| 110/TCP | pop3   | Протокол почтового отделения (Post Office Protocol) версии 3                                                        |
| 443/TCP | https  | Протокол HTTP поверх SSL                                                                                            |
| 992/TCP | telnet | Telnet поверх SSL (TelnetS)                                                                                         |
| 993/TCP | imap   | IMAP поверх SSL (IMAPS)                                                                                             |
| 994/TCP | irc    | IRC поверх SSL (IRCS)                                                                                               |
| 995/TCP | pop3s  | POP 3 поверх SSL (POP3S)                                                                                            |

## ВЫПОЛНЕНИЕ РАБОТЫ

Для наглядной демонстрации на практике будет рассмотрен ряд полезных сетевых утилит, которые позволяют вести мониторинг состояния стека TCP/IP.

Первой утилитой будет netstat. Данная утилита позволяет получать информацию об активных сетевых соединениях на уровне TCP и UDP протоколов, получать базовую статистику по количеству переданных и полученных пакетов уровня IP и т.д. Синтаксис команды можно почерпнуть из встроенной справки в саму утилиту ( «netstat /?» )

Рассмотрим несколько примеров.

«netstat -a» – получение информации обо всех установленных соединениях и открытых на прослушивание портах (см. рис. 7)

«netstat -n -b» – получение информации обо всех активных соединениях и процессах инициировавших их (см. рис. 8)

«netstat -e -s» – получение основной статистики по всем протоколам (ethernet, IPv4, IPv6, TCP, UDP) (см. рис. 9)

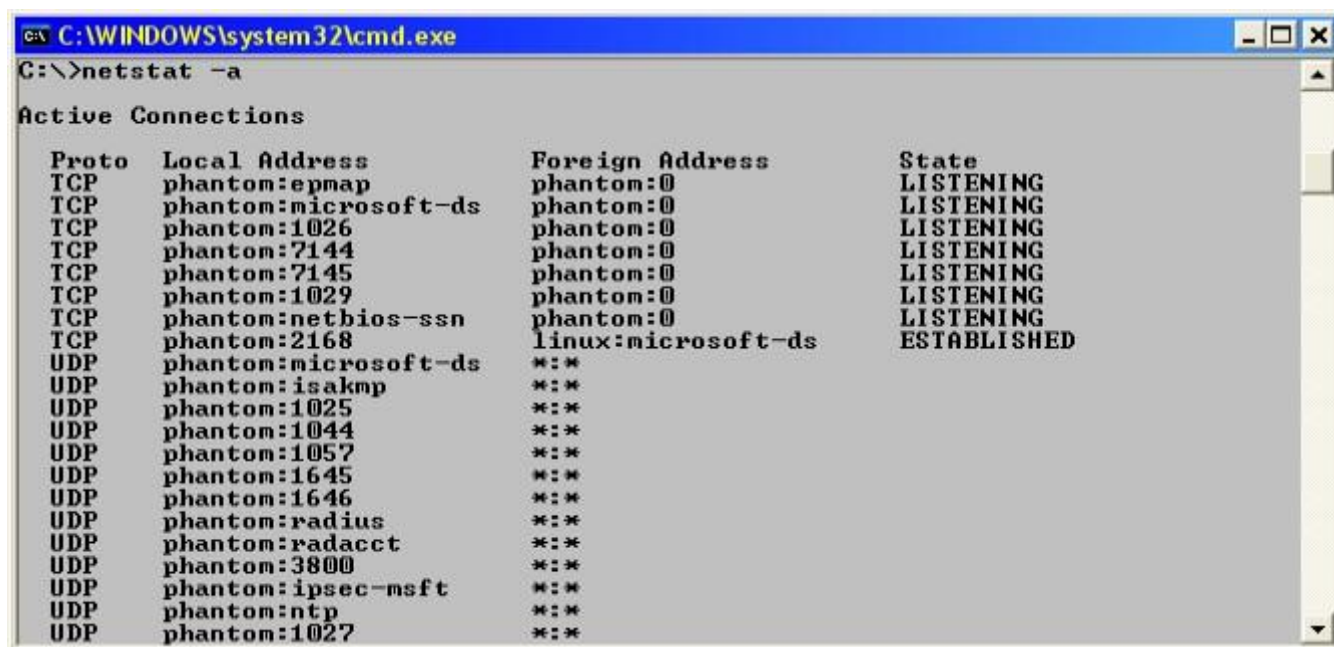


Рисунок 7 - Список всех установленных TCP/UDP соединений и открытых на прослушивание портов.

```

C:\>netstat -n -b

Active Connections

Proto Local Address          Foreign Address        State       PID
TCP   192.168.1.2:2168        192.168.1.6:445       ESTABLISHED 4
[System]

C:\>

```

Рисунок 8 - Список активных соединений с указанием процесса

```

Z:\>netstat -e -s

Interface Statistics

Received Sent
Bytes 30095310 142750144
Unicast packets 81452 39921
Non-unicast packets 11700 136
Discards 0 0
Errors 0 0
Unknown protocols 784 0

IPv4 Statistics

Packets Received = 85277
Received Header Errors = 0
Received Address Errors = 0
Datagrams Forwarded = 0
Unknown Protocols Received = 0
Received Packets Discarded = 0

```

Рисунок 9 - Статистика по протоколам

Очень удобным аналогом утилиты netstat для отслеживания активных соединений является утилита TCPView (см. рис. 10). Она позволяет в интерактивном режиме отслеживать сетевые соединения, а также позволяет получать информацию о процессах, установивших соединение и завершать их в случае необходимости

The screenshot shows the TCPView application window from Sysinternals. The title bar reads 'TCPView - Sysinternals: www.sysinternals.com'. The menu bar includes 'File', 'Options', 'Process', 'View', and 'Help'. Below the menu is a toolbar with icons for saving, refreshing, and adding filters. The main area is a table with the following columns: 'Process', 'Protocol', 'Local Address', 'Remote Address', and 'State'. The table lists various processes and their network connections, including 'alg.exe:1044', 'explorer.exe:1776', 'Far.exe:3592', 'lsass.exe:1108', 'Maxthon.exe:1728', 'Maxthon.exe:2156', and several instances of 'svchost.exe'.

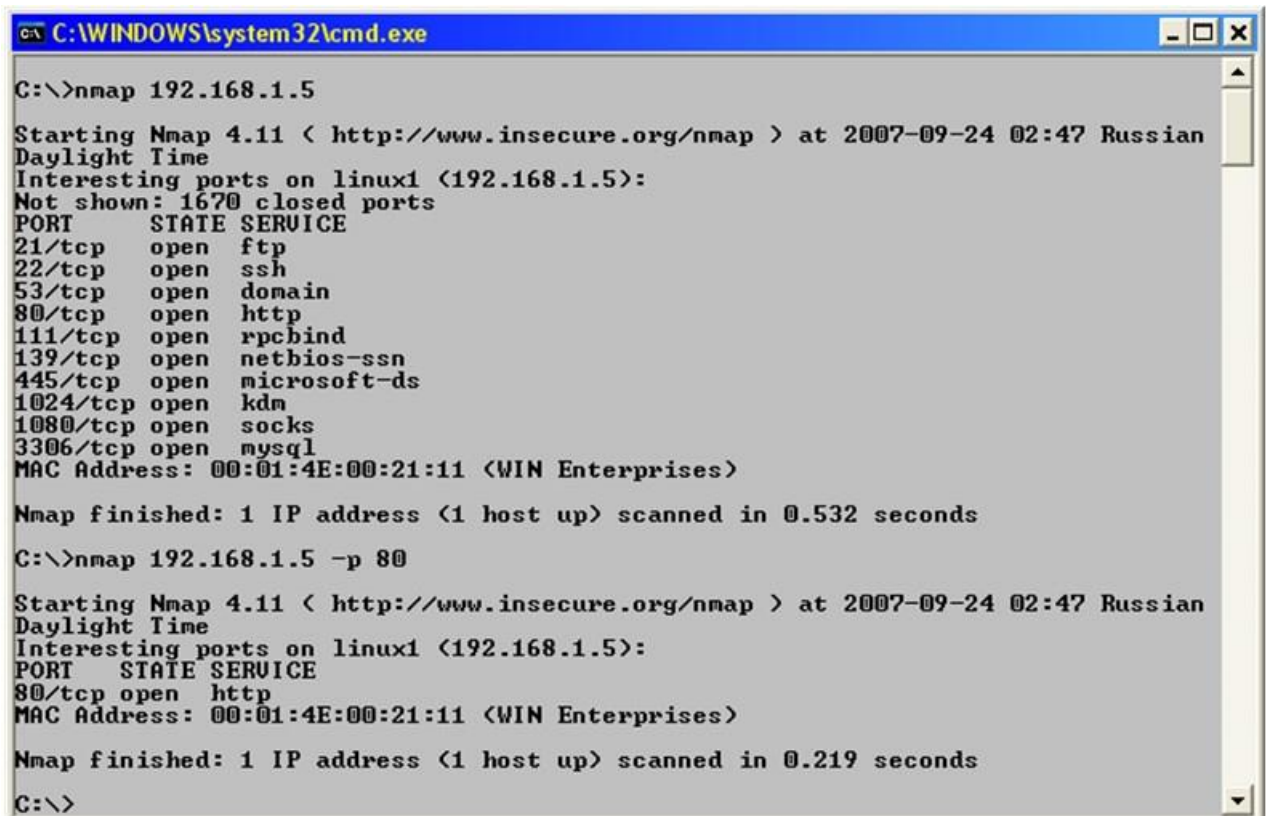
| Process           | Protocol | Local Address      | Remote Address  | State       |
|-------------------|----------|--------------------|-----------------|-------------|
| alg.exe:1044      | TCP      | phantom:1029       | phantom:0       | LISTENING   |
| explorer.exe:1776 | UDP      | phantom:1243       | N.N.            |             |
| Far.exe:3592      | TCP      | win2003:4226       | 10.197.0.11:ftp | ESTABLISHED |
| lsass.exe:1108    | TCP      | phantom:1026       | phantom:0       | LISTENING   |
| lsass.exe:1108    | UDP      | phantom:isakmp     | N.N.            |             |
| lsass.exe:1108    | UDP      | phantom:ipsec-msft | N.N.            |             |
| Maxthon.exe:1728  | UDP      | phantom:1041       | N.N.            |             |
| Maxthon.exe:2156  | UDP      | phantom:1108       | N.N.            |             |
| svchost.exe:1388  | TCP      | phantom:epmap      | phantom:0       | LISTENING   |
| svchost.exe:1452  | UDP      | phantom:radacct    | N.N.            |             |
| svchost.exe:1452  | UDP      | phantom:radius     | N.N.            |             |
| svchost.exe:1452  | UDP      | phantom:1646       | N.N.            |             |
| svchost.exe:1452  | UDP      | phantom:1645       | N.N.            |             |
| svchost.exe:1452  | UDP      | phantom:1027       | N.N.            |             |
| svchost.exe:1452  | UDP      | phantom:1028       | N.N.            |             |
| svchost.exe:1480  | UDP      | phantom:1044       | N.N.            |             |
| svchost.exe:1480  | UDP      | phantom:1025       | N.N.            |             |
| svchost.exe:1480  | UDP      | phantom:1057       | N.N.            |             |
| svchost.exe:1480  | UDP      | phantom:3800       | N.N.            |             |

Рисунок 10 - Утилита TCPView

Часто возникает необходимость узнать, какие сетевые сервисы запущены на удаленной машине. Для решения данной задачи служат т.н. сканеры портов. Данная группа утилит позволяет с некоторой точностью узнать, какие порты открыты на удаленной машине и некоторые другие параметры. Зная номера портов зачастую можно с достаточно большой уверенностью предположить, какие сервисы запущены на удаленной машине.

Наиболее достоверная информация может быть получена для протокола TCP. Задача детектирования UDP-сервисов не всегда может быть решена. Поэтому высокая точность сканирования UDP-сервисов не гарантируется.

Одним из самых распространенных профессиональных сканеров портов является сканер «NMAP» (см. рис. 11)



```
C:\WINDOWS\system32\cmd.exe

C:\>nmap 192.168.1.5

Starting Nmap 4.11 ( http://www.insecure.org/nmap ) at 2007-09-24 02:47 Russian Daylight Time
Interesting ports on linux1 (192.168.1.5):
Not shown: 1670 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
1024/tcp  open  kdm
1080/tcp  open  socks
3306/tcp  open  mysql
MAC Address: 00:01:4E:00:21:11 (WIN Enterprises)

Nmap finished: 1 IP address (1 host up) scanned in 0.532 seconds

C:\>nmap 192.168.1.5 -p 80

Starting Nmap 4.11 ( http://www.insecure.org/nmap ) at 2007-09-24 02:47 Russian Daylight Time
Interesting ports on linux1 (192.168.1.5):
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 00:01:4E:00:21:11 (WIN Enterprises)

Nmap finished: 1 IP address (1 host up) scanned in 0.219 seconds

C:\>
```

Рисунок 11 – Сканер портов NMAP

## ЗАДАНИЕ

1. Изучить стек протоколов TCP/IP.
2. Найти описание протоколов IP, TCP и UDP в соответствующих RFC.
3. Изучить утилиты netstat и tcpview: проанализировать текущие сетевые соединения на сетевой машине, получить статистику по протоколам (только netstat).
4. Изучить основные команды сканера портов nmap

## КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое сетевой протокол?
2. Зачем необходима стандартизация протоколов?
3. Понятие стека протоколов
4. Зачем введена модель OSI/ISO
5. Перечислите уровни стека протоколов TCP/IP и кратко охарактеризуйте их назначение.
6. Что такое IP-адрес?
7. В чем принципиальное отличие протоколов TCP и UDP.
8. Что такое сокет?
9. Зачем введен механизм сетевых портов?
10. Есть ли различие в протоколах реализованных, например, для ОС Windows и Linux?

## СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Танненбаум Э. Компьютерные сети. [Текст]/ Э. Танненбаум: СПб.: Питер, 2002.
2. Куроуз Дж. Ф., Росс К. В. Компьютерные сети. [Текст]/ Дж. Ф. Куроуз, К.В. Росс: СПб.: Питер, 2004.
3. TCPView for Windows [Электронный ресурс]: / Internet. - <https://technet.microsoft.com/ru-ru/sysinternals/tcpview.aspx> (28.09.17).