

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Ульяновский государственный университет»

**О. Л. Курилова**

## **Инфокоммуникационные системы и сети**

**Лабораторный практикум**



Ульяновск 2016

УДК 621.39:004.7(075.8)  
ББК 32.883я73+32.971.35я73

Печатается по решению Ученого совета  
Факультета математики, информатики и авиационных технологий  
Ульяновского государственного университета  
(протокол № \_\_\_\_\_ г.)

Рецензенты:  
профессор, доктор технических наук,  
зав. каф. «Телекоммуникации» УлГТУ **Васильев К.К.**  
профессор, доктор технических наук,  
зав. каф. «Телекоммуникационные технологии и сети» УлГУ **Смагин А.А.**

Курилова О. Л.

**К93 Инфокоммуникационные системы и сети:** лабораторный практикум / О. Л. Курилова.- Ульяновск: УлГУ, 2016.- 115 с.

Настоящее издание является руководством к лабораторному практикуму по дисциплине «Инфокоммуникационные системы и сети», входящей в образовательную программу по направлению подготовки 09.03.02 Информационные системы и технологии (уровень бакалавриата). Также лабораторный практикум будет интересен и полезен для студентов вузов, обучающихся по специальностям в области информационных технологий и сетей, преподавателям и специалистам в области построения компьютерных сетей.

© Курилова О. Л., 2016

© Ульяновский государственный университет, 2016

## Содержание

ВВЕДЕНИЕ .....	4
ЛАБОРАТОРНАЯ РАБОТА №1 ДИАГНОСТИКА IP-ПРОТОКОЛА. ....	5
ЛАБОРАТОРНАЯ РАБОТА №2 АНАЛИЗ ПРОТОКОЛОВ СЕТЕВОГО И ТРАНСПОРТНОГО УРОВНЕЙ.....	11
ЛАБОРАТОРНАЯ РАБОТА №3 АНАЛИЗ ПРОТОКОЛОВ ПРИКЛАДНОГО УРОВНЯ.....	14
ЛАБОРАТОРНАЯ РАБОТА №4 АНАЛИЗАТОР ПРОТОКОЛОВ COMMVIEW.....	24
ЛАБОРАТОРНАЯ РАБОТА №5 WWW-СЕРВИС, ПОИСК ИНФОРМАЦИИ В ИНТЕРНЕТ.....	30
ЛАБОРАТОРНАЯ РАБОТА №6 РАБОТА В РЕЖИМАХ FTP . ....	37
ЛАБОРАТОРНАЯ РАБОТА №7 ТЕЛЕКОНФЕРЕНЦИИ. ....	40
ЛАБОРАТОРНАЯ РАБОТА №8 ЭЛЕКТРОННАЯ ПОЧТА. ....	43
ЛАБОРАТОРНАЯ РАБОТА №9 RSS-ТЕХНОЛОГИИ.....	47
ЛАБОРАТОРНАЯ РАБОТА №10 СОЗДАНИЕ HTML-ДОКУМЕНТА.....	49
ЛАБОРАТОРНАЯ РАБОТА №11 РАЗМЕЩЕНИЕ ГРАФИКИ НА WEB-СТРАНИЦЕ, СПИСКИ. ....	58
ЛАБОРАТОРНАЯ РАБОТА №12 СОЗДАНИЕ ТАБЛИЦ В HTML-ДОКУМЕНТЕ.....	63
ЛАБОРАТОРНАЯ РАБОТА №13 СОЗДАНИЕ ГИПЕРССЫЛОК В HTML-ДОКУМЕНТЕ.....	68
ЛАБОРАТОРНАЯ РАБОТА №14 СОЗДАНИЕ ФРЕЙМОВ В HTML-ДОКУМЕНТЕ. ....	70
ЛАБОРАТОРНАЯ РАБОТА №15 ЗНАКОМСТВО СО СРЕДОЙ BOSON NETWORK DESIGNER.....	74
ЛАБОРАТОРНАЯ РАБОТА №16 ВВЕДЕНИЕ В МЕЖСЕТЕВУЮ ОПЕРАЦИОННУЮ СИСТЕМУ IOS КОМПАНИИ CISCO. ....	82
ЛАБОРАТОРНАЯ РАБОТА №17 СТАТИЧЕСКАЯ МАРШРУТИЗАЦИЯ. ....	96
ЛАБОРАТОРНАЯ РАБОТА №18 ДИНАМИЧЕСКАЯ МАРШРУТИЗАЦИЯ. ....	104
СПИСОК ЛИТЕРАТУРЫ И ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ. ....	116

## Введение

В настоящее время компьютерные сети являются ключевой составляющей современных информационно-телекоммуникационных систем. Лабораторный практикум позволит наиболее глубоко и подробно изучить возможности современных сетевых технологий, протоколов и маршрутизаторов.

Целью лабораторного практикума является формирование у студентов системы знаний по общей теории инфокоммуникационных систем и сетей с учетом тенденций современного развития.

Изучение лабораторного практикума базируется на успешно усвоенных базовых понятиях дисциплин математического и естественно-научного цикла и дисциплин профессионального цикла: «Информатика», «Операционные системы и среды», «Архитектура информационных систем», «Web-технологии».

Настоящее издание является руководством к лабораторному практикуму по дисциплине «Инфокоммуникационные системы и сети», входящей в образовательную программу по направлению подготовки 09.03.02 «Информационные системы и технологии» (уровень бакалавриата). Описание каждой лабораторной работы включает: название, цель, краткие теоретические сведения, постановку задачи, последовательность действий исполнителя, а также вопросы и задания для самостоятельных исследований.

Лабораторные работы можно разделить на несколько частей:

- в лабораторной работе №1 исследуются методы контроля и мониторинга с помощью стандартных утилит сетей, построенных на операционной системе *Windows*;
- в лабораторных работах №2-3 изучаются протоколы сетевого и транспортного уровней и прикладные протоколы (*HTTP* и *FTP*) стека *TCP/IP* с помощью программы *Network Monitor*;
- в лабораторной работе №4 анализируются различные протоколы и фильтры, позволяющие контролировать сетевой трафик, на примере программы *CommView*;
- в лабораторных работах №5-8 исследуются различные сервисы Интернета - *WWW*-сервис, поиск информации в Интернет с использованием различных поисковых систем и средств поиска; работа в режимах *FTP*; создание учетной записи и умение работать в *NNTP*-телеконференциях; работа с электронной почтой - *e-mail* и *web-mail*; знакомство с *RSS*-технологиями;
- в лабораторных работах №9-14 исследуются различные возможности создания *HTML* – страниц с помощью языка разметки гипертекста *HTML*, содержащие элементы форматирования, графику, списки, таблицы, гиперссылки, фреймы;
- в лабораторных работах №15-18 исследуется работа в среде *Boson Network Designer* - создание топологии компьютерной сети, назначение компьютерам адресов, пингование компьютеров, конфигурация интерфейсов, настройка IP адресов интерфейсов, применение команды *Telnet*; статическая маршрутизация; динамическая маршрутизация; конфигурирование динамической маршрутизации, использование протоколов *RIP*, *IGRP*, *OSPF*. Программное обеспечение «*Boson Network*» позволяет эмулировать все основные процессы функционирования реальных компьютерных сетей и предоставляет в распоряжение обучающегося виртуальную компьютерную сеть, в которой и проводятся все настройки сетевой инфраструктуры и изучаются сетевые технологии

Лабораторный практикум подготовлен на кафедре «Телекоммуникационных технологий и сетей» и предназначен для студентов, обучающихся по направлению подготовки «Информационные системы и технологии». Также лабораторный практикум будет интересен и полезен для студентов вузов, обучающихся по специальностям в области информационных технологий и сетей, преподавателям и специалистам в области построения компьютерных сетей.

## Лабораторная работа №1 Диагностика IP-протокола.

**Тема работы:** диагностика IP-протокола.

**Цель работы:** научиться проверять работоспособность сетевого подключения, ознакомиться с утилитами TCP/IP.

**Теоретическая часть:**

IP-адрес, состоит из 4 байт, например, 109.26.17.100. Этот адрес используется на сетевом уровне. IP-адрес состоит из двух частей: номера сети и номера узла. IP-адрес характеризует не отдельный компьютер или маршрутизатор, а одно сетевое соединение. IP-адрес имеет длину 4 байта и обычно записывается в виде четырех чисел, представляющих значения каждого байта в десятичной форме, и разделенных точками, например: 128.10.2.30 - традиционная десятичная форма представления адреса, 10000000 00001010 00000010 00011110 - двоичная форма представления этого же адреса.

Символьный идентификатор-имя, например, SERV1.IBM.COM. Этот адрес состоит из нескольких частей, например, имени машины, имени организации, имени домена. Такой адрес, называемый также DNS-именем, используется на прикладном уровне, например, в протоколах FTP или telnet.

DNS (Domain Name System) - это распределенная база данных, поддерживающая иерархическую систему имен для идентификации узлов в сети Internet. Служба DNS предназначена для автоматического поиска IP-адреса по известному символьному имени узла.

Существуют различные утилиты, позволяющие быстро продиагностировать IP-подключение. Однако большинство операций легко может быть выполнено с использованием команд самой операционной системы.

Пользователи Windows XP для диагностики сетевого подключения могут воспользоваться специальным мастером. Эта программа вызывается из меню задачи *Сведения о системе* (Пуск > Все программы > Стандартные > Служебные > Сведения о системе > меню Сервис > Диагностика сети, затем *Собрать информацию*) (рис.1).

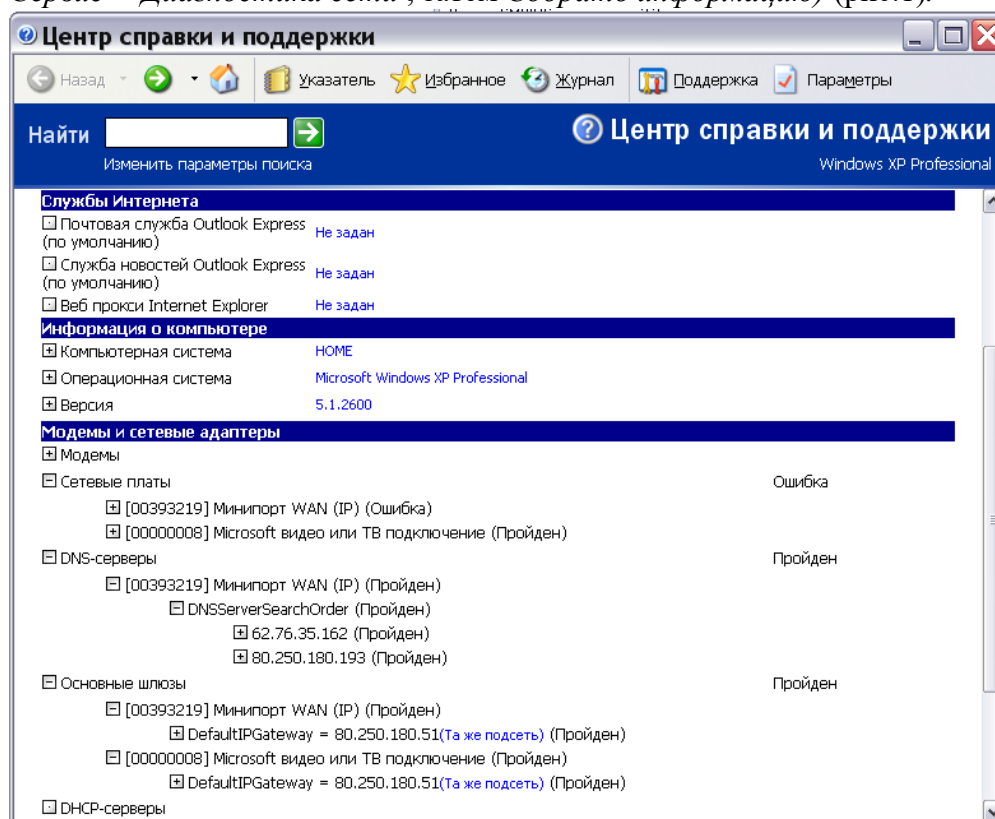


Рисунок 1. Сведения о системе.

В ходе выполнения лабораторной работы Вы познакомитесь с утилитами, запускаемыми из командной строки, позволяющими детально продиагностировать работоспособность подключения Вашего компьютера к сети.

#### Ход работы:

#### **Ipconfig**

Для отображения параметров IP-протокола используются утилиты *ipconfig* (Windows NT/2000/XP) и *winipcfg* (Windows 9x). Эта утилита выводит на экран основные параметры настройки протокола TCP/IP: значения адреса, маски, шлюза.

1. Нажмите кнопку *Пуск*, выберите строку меню *Выполнить*, наберите символы *cmd* и нажмите клавишу *Enter* на клавиатуре.
2. В открывшемся окне наберите *ipconfig /all*. При нормальной работе компьютера на экран должен вывестись примерно такой листинг:

Адаптер беспроводной локальной сети Беспроводная сеть:

```
DNS-суффикс подключения . . . . . :
Описание . . . . . : Intel(R) Centrino(R) Advanced-N 6235
Физический адрес . . . . . : C4-85-08-C3-5B-D6
DHCP включен . . . . . : Да
Автонастройка включена . . . . . : Да
Локальный IPv6-адрес канала . . . . : fe80::305b:9b95:9c4a:df14%2(Основной)
IPv4-адрес . . . . . : 192.168.1.38(Основной)
Маска подсети . . . . . : 255.255.255.0
Аренда получена . . . . . : 15 мая 2016 г. 12:08:09
Срок аренды истекает . . . . . : 15 мая 2016 г. 19:08:09
Основной шлюз . . . . . : 192.168.1.1
DHCP-сервер . . . . . : 192.168.1.1
IAID DHCPv6 . . . . . : 331646216
DUID клиента DHCPv6 . . . . . : 00-01-00-01-19-1D-CD-B8-50-B7-C3-69-61-38

DNS-серверы . . . . . : 192.168.1.1
NetBios через TCP/IP . . . . . : Включен
```

3. Отключите сетевое подключение, повторите команду. При отсутствующем соединении изучите полученный листинг. Рассмотрите отличия по сравнению с п. 2.

#### **Ping**

Команда используется для проверки протокола TCP/IP и достижимости удаленного компьютера. Она выводит на экран время, за которое пакеты данных достигают заданного в ее параметрах компьютера.

Проверка правильности установки протокола TCP/IP. Нажмите кнопку *Пуск*, выберите строку меню *Выполнить*, наберите символы *cmd* и нажмите клавишу *Enter* на клавиатуре.

*ping 127.0.0.1*

Адрес 127.0.0.1 — это личный адрес любого компьютера. Таким образом, эта команда проверяет прохождение сигнала "на самого себя". Она может быть выполнена без наличия какого-либо сетевого подключения. Вы должны увидеть приблизительно следующие строки:

```
C:\WINDOWS\system32>ping 127.0.0.1
```

```
Обмен пакетами с 127.0.0.1 по с 32 байтами данных:  
Ответ от 127.0.0.1: число байт=32 время<1мс TTL=128  
Ответ от 127.0.0.1: число байт=32 время<1мс TTL=128  
Ответ от 127.0.0.1: число байт=32 время<1мс TTL=128  
Ответ от 127.0.0.1: число байт=32 время<1мс TTL=128
```

```
Статистика Ping для 127.0.0.1:
```

```
Пакетов: отправлено = 4, получено = 4, потеряно = 0  
(0% потерь)
```

```
Приблизительное время приема-передачи в мс:
```

```
Минимальное = 0мсек, Максимальное = 0 мсек, Среднее = 0 мсек
```

По умолчанию команда посылает пакет 32 байта. Размер пакета может быть увеличен до 65 кбайт. Так можно обнаружить ошибки при пересылке пакетов больших размеров. За размером тестового пакета отображается время отклика удаленной системы (в нашем случае — меньше 1 миллисекунды). Потом показывается еще один параметр протокола — значение TTL. TTL — "время жизни" пакета. На практике это число маршрутизаторов, через которые может пройти пакет. каждый маршрутизатор уменьшает значение TTL на единицу. При достижении нулевого значения пакет уничтожается. Такой механизм введен для исключения случаев закливания пакетов.

Если будет показано сообщение о недостижимости адресата, то это означает ошибку установки протокола IP. В этом случае целесообразно удалить протокол из системы, перезагрузить компьютер и вновь установить поддержку протокола TCP/IP.

1. Проверка видимости локального компьютера и ближайшего компьютера сети.

Выполните команду

```
ping 80.250.180.93
```

На экран должны быть выведены примерно такие строки:

```
C:\WINDOWS\system32>ping 80.250.180.193
```

```
Обмен пакетами с 80.250.180.193 по с 32 байтами данных:  
Ответ от 80.250.180.193: число байт=32 время=74мс TTL=51  
Ответ от 80.250.180.193: число байт=32 время=73мс TTL=51  
Ответ от 80.250.180.193: число байт=32 время=71мс TTL=51  
Ответ от 80.250.180.193: число байт=32 время=71мс TTL=51
```

```
Статистика Ping для 80.250.180.193:
```

```
Пакетов: отправлено = 4, получено = 4, потеряно = 0  
(0% потерь)
```

```
Приблизительное время приема-передачи в мс:
```

```
Минимальное = 71мсек, Максимальное = 74 мсек, Среднее = 72 мсек
```

Наличие отклика свидетельствует о том, что канал связи установлен и работает.

2. Проверка работоспособности сервера имен Internet. Выполните команду

```
ping www.list.ru
```

Если система сможет различить IP-адрес этого хоста, то система распознавания имен работоспособна. На экран должен быть выведен примерно такой листинг:

```
C:\WINDOWS\system32>ping www.list.ru
```

```
Обмен пакетами с list.ru [217.69.134.229] с 32 байтами данных:  
Ответ от 217.69.134.229: число байт=32 время=18мс TTL=55  
Ответ от 217.69.134.229: число байт=32 время=17мс TTL=55  
Ответ от 217.69.134.229: число байт=32 время=17мс TTL=55  
Ответ от 217.69.134.229: число байт=32 время=17мс TTL=55
```

```
Статистика Ping для 217.69.134.229:
```

```
Пакетов: отправлено = 4, получено = 4, потеряно = 0  
(0% потерь)
```

```
Приблизительное время приема-передачи в мс:
```

```
Минимальное = 17мсек, Максимальное = 18 мсек, Среднее = 17 мсек
```

Если не будет ответа на ввод команды с именем существующего хоста, то это может свидетельствовать либо об ошибке в задании DNS-серверов, либо об их неработоспособности.

## Tracert

При работе в Сети одни информационные серверы откликаются быстрее, другие медленнее, бывают случаи недостижимости желаемого хоста. Для выяснения причин подобных ситуаций можно использовать специальные утилиты.

Например, команда *tracert*, которая обычно используется для показа пути прохождения сигнала до желаемого хоста. Зачастую это позволяет выяснить причины плохой работоспособности канала. Точка, после которой время отклика резко увеличено, свидетельствует о наличии в этом месте "узкого горлышка", не справляющегося с нагрузкой.

Нажмите кнопку *Пуск*, выберите строку меню *Выполнить*, наберите символы `cmd` и нажмите клавишу *Enter* на клавиатуре. В командной строке введите команду:

```
tracert www.rambler.ru или www.ulsu.ru.
```

Вы должны увидеть примерно такой листинг:

```
C:\WINDOWS\system32>tracert www.rambler.ru

Трассировка маршрута к www.rambler.ru [81.19.82.1]
с максимальным числом прыжков 30:

  1    1 ms    <1 мс    1 ms    KEENETIC_GIGA [192.168.1.1]
  2    1 ms    1 ms     1 ms    bras253-1o10.ul.corbina.net [89.179.76.144]
  3    *       *        *       Превышен интервал ожидания для запроса.
  4    2 ms    1 ms     2 ms    78.107.188.218
  5    3 ms    2 ms     2 ms    194.186.77.157
  6   17 ms   17 ms    17 ms    pe26.Moscow.gldn.net [79.104.225.59]
  7   17 ms   17 ms    17 ms    mx.vicigroup.ru [194.186.124.146]
  8   17 ms   16 ms    17 ms    80-64-98-229.rascom.as20764.net [80.64.98.229]
  9   20 ms   17 ms    17 ms    j-mx240-xe-1-2-0-814.m10.rambler.ru [81.19.64.73]
]
10   17 ms   17 ms    17 ms    front.head.rambler.ru [81.19.82.1]

Трассировка завершена.
```

## Nslookup

Утилита используется для получения информации от DNS-сервера. По умолчанию (после запуска без указания параметров) осуществляется подключение к указанному в настройках протокола серверу DNS. Набирая необходимые имена в качестве запроса, вы можете получить информацию о данных DNS по этому имени, найти почтовый сервер, обслуживающий домен, уточнить данные регистрации и т.п.

1. Выполните команду *nslookup*.
2. Наберите *server www.ulsu.ru* и нажмите *Enter* — этой командой мы указываем, какой DNS-сервер мы хотим использовать для получения интересующих нас данных.
3. Наберите *ulsu.ru* и нажмите *Enter* — этой командой мы запрашиваем данные по домену *ulsu.ru*

На экране вы должны получить примерно такой листинг:

```
> server www.ulsu.ru
њХњтХњ яю съюњњрњш█: host1.ulsu.ru
Address: 80.250.180.133
Aliases: www.ulsu.ru
```



```

> ulsu.ru
Ухѐтхѐ: UnKnown
Address: 192.168.1.1

Не заслуживающий доверия ответ:
ulsu.ru
      primary name server = ns.ulsu.ru
      responsible mail addr = bindmaster.ulsu.ru
      serial = 2014704500
      refresh = 10800 (3 hours)
      retry = 3600 (1 hour)
      expire = 864000 (10 days)
      default TTL = 86400 (1 day)
ulsu.ru
      RP mailbox = noc.ulsu.ru
      text location = noc.ulsu.ru
ulsu.ru text =

      "Ulyanovsk State University"
ulsu.ru MX preference = 10, mail exchanger = runnet.uven.ru
ulsu.ru internet address = 62.76.32.86
ulsu.ru nameserver = ns.uven.ru
ulsu.ru nameserver = ns2.uven.ru

```

## Netstat

Эта утилита выводит TCP/IP статистику и список TCP/IP соединений.

Нажмите кнопку *Пуск*, выберите строку меню *Выполнить*, наберите символы `cmd` и нажмите клавишу *Enter*. В командной строке введите команду: `NETSTAT`. Изучите работу команды с ключами: `NETSTAT -a`, `NETSTAT -o`, `NETSTAT -s`, `NETSTAT -e`, `NETSTAT -r`.

Замечание: прежде, чем выполнять эту команду, запустите Internet Explorer и загрузите несколько страниц.

Ниже перечислены все возможные ключи для данной программы:

```
C:\WINDOWS\system32>netstat help
```

Отображение статистики протокола и текущих сетевых подключений TCP/IP.

```
NETSTAT [-a] [-b] [-e] [-f] [-n] [-o] [-p протокол] [-r] [-s] [-x] [-t]
[интервал]
```

- a Отображение всех подключений и портов прослушивания.
- b Отображение исполняемого файла, участвующего в создании каждого подключения или порта прослушивания. Иногда известные исполняемые файлы содержат множество независимых компонентов. Тогда отображается последовательность компонентов, участвующих в создании подключения или порта прослушивания. В этом случае имя исполняемого файла находится снизу в скобках [], сверху находится вызванный им компонент, и так до тех пор, пока не достигнут TCP/IP. Заметьте, что такой подход может занять много времени и требует достаточных разрешений.
- e Отображение статистики Ethernet. Может применяться вместе с параметром -s.
- f Отображение полного имени домена (FQDN) для внешних адресов.
- n Отображение адресов и номеров портов в числовом формате.
- o Отображение ID процесса каждого подключения.
- p протокол Отображение подключений для протокола, задаваемых этим параметром. Допустимые значения: TCP, UDP, TCPv6 или UDPv6. Если используется вместе с параметром -s для отображения статистики по протоколам, допустимы следующие значения: IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP или UDPv6.
- r Отображение содержимого таблицы маршрутов.
- s Отображение статистики по протоколам. По умолчанию статистика отображается для протоколов IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP и UDPv6. Параметр -r позволяет указать подмножество выводимых данных.

Примерный листинг для этой программы:

```
C:\WINDOWS\system32>netstat
```

Активные подключения

Имя	Локальный адрес	Внешний адрес	Состояние
TCP	192.168.1.38:2869	192.168.1.33:40909	TIME_WAIT
TCP	192.168.1.38:49212	hk2sch130020624:https	ESTABLISHED
TCP	192.168.1.38:52554	um02:http	TIME_WAIT

```
C:\WINDOWS\system32>netstat -r
```

```
=====
Список интерфейсов
32...c4 85 08 c3 5b da .....Устройства Bluetooth (личной сети)
6...c4 85 08 c3 5b d7 .....Виртуальный адаптер Wi-Fi Direct (Майкрософт)
5...c6 85 08 c3 5b d6 .....Виртуальный адаптер размещенной сети (Майкрософт)
4...50 b7 c3 69 61 38 .....Контроллер семейства Realtek PCIe GBE
2...c4 85 08 c3 5b d6 .....Intel(R) Centrino(R) Advanced-N 6235
1.....Software Loopback Interface 1
=====
```

IPv4 таблица маршрута

```
=====
Активные маршруты:
Сетевой адрес          Маска сети            Адрес шлюза           Интерфейс             Метрика
0.0.0.0                0.0.0.0               192.168.1.1          192.168.1.38         25
127.0.0.0              255.0.0.0             On-link              127.0.0.1            306
127.0.0.1              255.255.255.255      On-link              127.0.0.1            306
127.255.255.255       255.255.255.255      On-link              127.0.0.1            306
192.168.1.0            255.255.255.0        On-link              192.168.1.38         281
192.168.1.38           255.255.255.255      On-link              192.168.1.38         281
192.168.1.255          255.255.255.255      On-link              192.168.1.38         281
224.0.0.0              240.0.0.0             On-link              127.0.0.1            306
224.0.0.0              240.0.0.0             On-link              192.168.1.38         281
255.255.255.255       255.255.255.255      On-link              127.0.0.1            306
255.255.255.255       255.255.255.255      On-link              192.168.1.38         281
=====
```

Постоянные маршруты:

Отсутствует

### Результаты лабораторной работы.

В результате выполнения этой работы студент должен представить преподавателю программу-отчет (выполненную в любом текстовом редакторе), которая будет содержать результаты выполнения каждой утилиты, а именно листинги выполнения этих программ. Для сохранения результатов выполнения различных команд используйте кнопку на клавиатуре Prt Scr.

## Лабораторная работа №2 Анализ протоколов сетевого и транспортного уровней.

**Тема работы:** анализ протоколов сетевого и транспортного уровней.

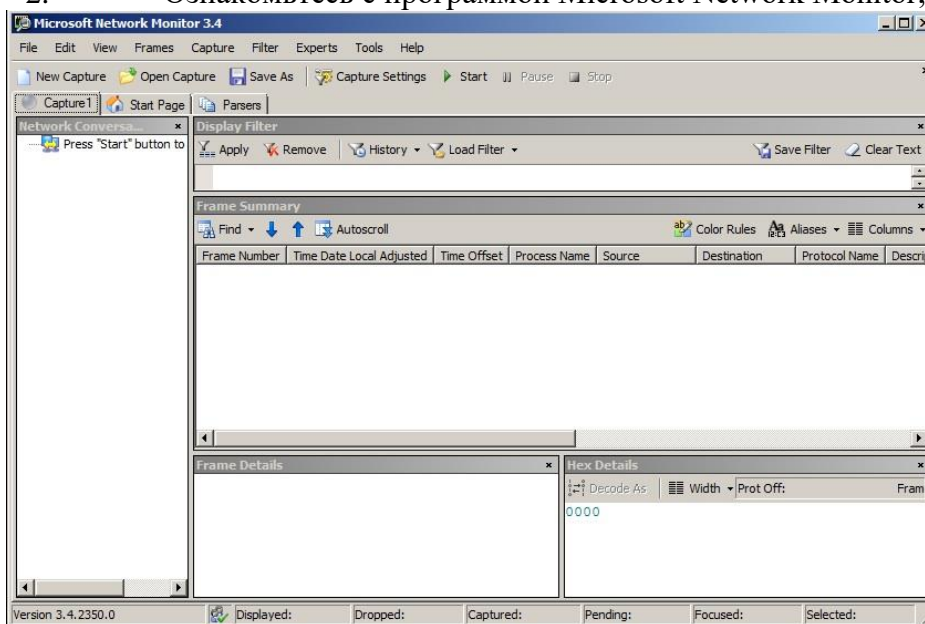
**Цель работы:** изучение протоколов сетевого и транспортного уровней стека TCP/IP и приобретение практических навыков в использовании программных средств, позволяющих контролировать сетевой трафик, на примере программы *Network Monitor*.

**Теоретическая часть:**

Сетевой монитор позволяет собирать данные, отправленные или полученные локальным компьютером из сетевого потока, и копировать их во временный файл записи. Сетевой монитор отображает статистику для собранных кадров динамически в окне сбора данных и позволяет создавать фильтр записи для кадров, удовлетворяющих определенным условиям. Сетевой монитор позволяет отображать, фильтровать, сохранять и печатать собранные данные.

Ход работы:

1. Установите программу Microsoft Network Monitor с официального сайта Microsoft, учитывая разрядность и тип процессора.
2. Ознакомьтесь с программой Microsoft Network Monitor;



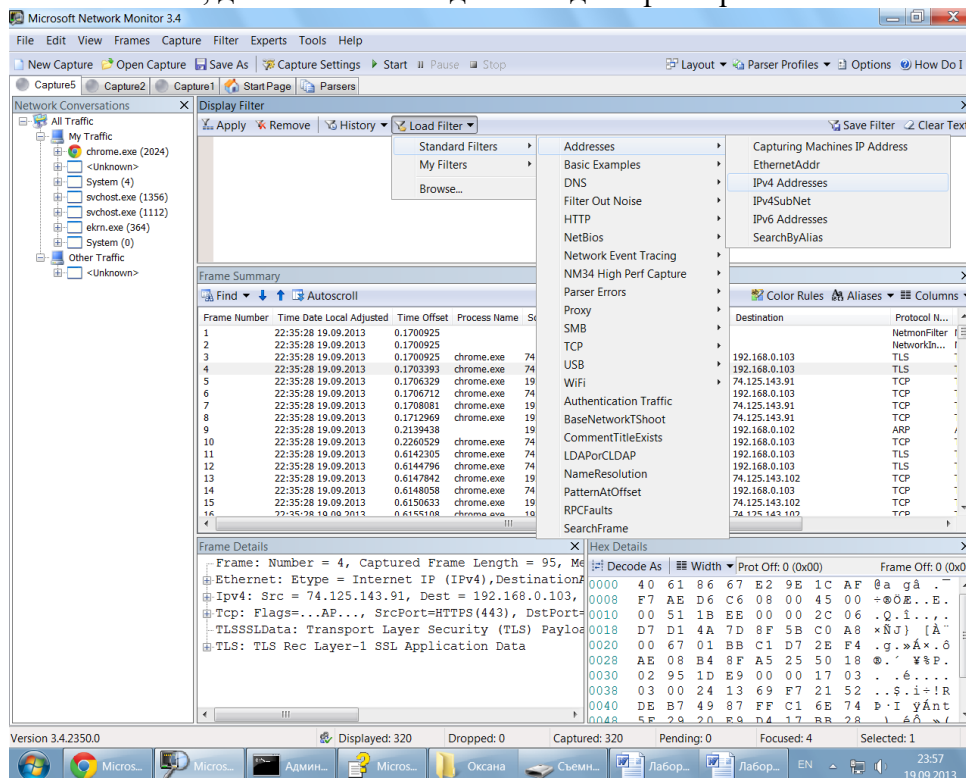
3. Выберите New capture (новый захват), нажмите Start и убедитесь, что программа захватывает сетевые пакеты. Подождите некоторое время и затем, используя браузер зайдите на любой сайт. Посмотрите, как изменяются захваченные кадры. Затем нажмите Stop и изучите полученные кадры.
4. В окне «Frame Summary» приводятся общие сведения о записанных кадрах в порядке их поступления:
  - Frame number– номер кадра, определяемый порядком записи кадров (нумерация кадров начинается с 1);
  - Time – время записи кадра в форме, заданной в параметрах отображения (абсолютное время записи кадра, время записи кадра относительно начала записи в миллисекундах или время записи кадра относительно времени записи предыдущего кадра в миллисекундах);
  - Source адрес компьютера, который отправил кадр;

- Destination - адрес компьютера, на который отправлен кадр;
- Protocol – основной протокол, используемый для отправки данных;
- Description – краткое описание содержимого кадра;

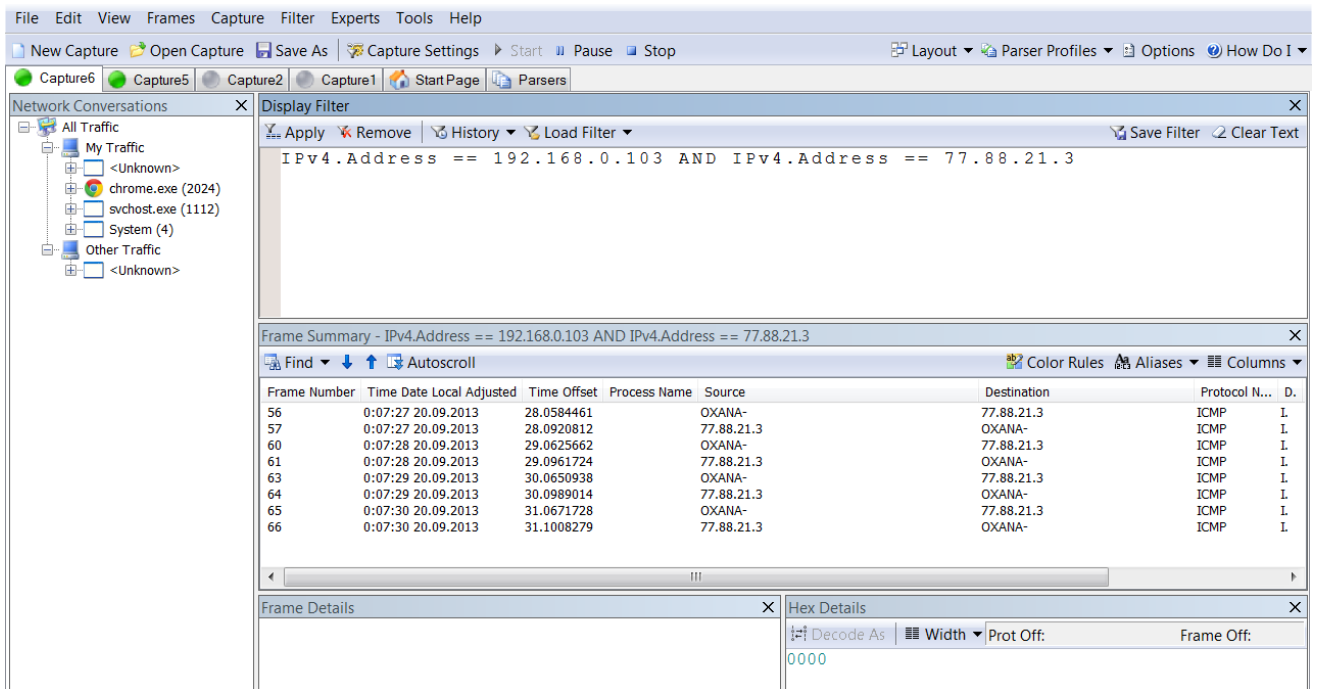
5. В панели Frame Details приводятся сведения о кадре, выделенном в панели Frame Summary, с указанием полей протоколов все уровней. Разберите несколько кадров по полям, отразите это в отчете.

6. Настройте в Network Monitor фильтр таким образом, чтобы перехватывались только данные, которые передаются между вашим и другим компьютером (например yandex.ru). Для этого

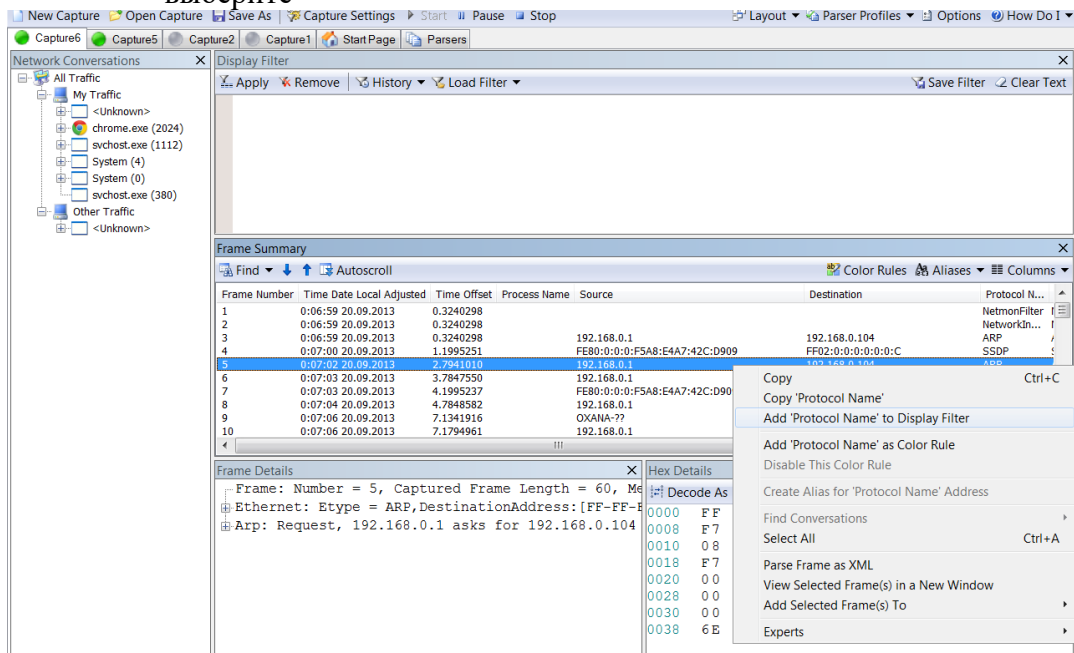
- выполните в командной строке команду *ping yandex.ru*;
- запомните IP адрес сайта и Ваш IP адрес;
- создайте новый захват кадров;
- затем в программе Microsoft Network Monitor перехватите и просмотрите эти пакеты, для этого необходимо создать фильтр



- В окне Display Filter введите, например IPv4.Address == 192.168.0.103 AND IPv4.Address == 77.88.21.3
- примените фильтр (Apply). В результате в центральном окне отразятся только кадры, передающиеся между Вашим и компьютером www.yandex.ru.



7. Далее создайте снова новый захват и примените фильтры, отбирающие кадры определенного протокола. Для этого выделите кадр с нужным протоколом, затем нажмите правой кнопкой мыши по нужному полю и выберите



8. В окне фильтров появится `ProtocolName == "ARP"`, затем примените фильтр и в центральной части отразятся кадры только этого протокола.
9. Отберите кадры других протоколов, добавляя к предыдущему фильтру новые условия, используя операторы `AND`, `OR`, `NOT`.
10. Отберите кадры, содержащие запросы и отклики от каких-либо программ. Придумайте несколько фильтров.
11. В отчете привести содержание пакетов при установлении и разрыве TCP соединения.
12. Сделайте подробный отчет с картинками по каждому пункту.

## Лабораторная работа №3 Анализ протоколов прикладного уровня.

**Тема работы:** анализ протоколов прикладного уровня.

**Цель работы:** изучение прикладных протоколов стека *TCP/IP* и приобретение практических навыков анализа протоколов с помощью программы *Network Monitor*.

**Теоретическая часть:**

### Описание протокола *HTTP*.

#### *1. Назначение и принцип работы протокола HTTP.*

Протокол прикладного уровня для передачи гипертекста *http (HyperText Transfer Protocol)* был разработан как основа Всемирной паутины *WWW (World Wide Web)* – распределенной системы, предоставляющей доступ к связанным между собой документам, расположенным на веб-серверах Интернета. Большинство ресурсов Всемирной паутины представляет собой гипертекст. Гипертекстовые документы, размещаемые во Всемирной паутине, называются веб-страницами. Несколько *web-страниц*, объединенных общей темой, дизайном, а также связанных между собой ссылками и обычно находящихся на одном и том же веб-сервере, называются веб-сайтом. Для загрузки и просмотра веб-страниц используются специальные программы – браузеры.

Основой *HTTP* является технология «клиент–сервер», т.е. программа-клиент инициирует соединение и выдает ему *HTTP-запрос*. Сервер обрабатывает этот запрос, производит необходимые действия и возвращает клиенту *HTTP-ответ* с результатом.

Протокол *HTTP* никак не связан со способом визуализации веб-страниц и определяет только метод обмена информацией между клиентом и сервером. Служба *WWW*, помимо протокола *HTTP*, базируется еще на трех стандартах:

- универсальном способе адресации ресурсов в сети *URL (Universal Resource Locator)*;
- языке гипертекстовой разметки документов *HTML (HyperText Markup Language)*;
- универсальном интерфейсе шлюзов *CGI (Common Gateway Interface)*, обеспечивающем взаимодействия *HTTP-сервера* с другими программами (например, СУБД).

Версии протокола *HTTP* используют *TCP* в качестве протокола транспортного уровня. *HTTP-клиент* сначала устанавливает *TCP-соединение* с сервером, после чего клиент и сервер начинают взаимодействовать с протоколом *TCP* через интерфейс сокетов (стандартный номер порта–80). После завершения обслуживания клиентов сервер не сохраняет о них никакой информации, т.е. протокол *HTTP* является протоколом без запоминания состояния соединения.

Протокол *HTTP* поддерживает постоянные и непостоянные соединения. Непостоянное соединение состоит из единственного сообщения-запроса и сообщения-ответа. Для получения веб-страницы требуется многократное установление и завершение соединения. При этом необходимо учитывать, что каждое соединение требует от протокола *TCP* выделения буфера, а также ряда служебных переменных как на стороне клиента, так и на стороне сервера. Так как многие веб-серверы параллельно обслуживают сотни клиентов, подобная схема серьезно затрудняет процесс взаимодействия между клиентами и сервером. Кроме того, установление соединения для каждого объекта из-за времени оборота приводит к дополнительным временным затратам.

При постоянном соединении сервер не закрывает *TCP-соединение* после обслуживания запроса, что позволяет обслужить несколько запросов в одном соединении. Передача веб-страниц через одно соединение возможна в случаях, если все объекты находятся на одном и том же сервере. Обычно закрытие *TCP-соединения* происходит в случае, когда оно не используется в течение некоторого установленного времени (интервала ожидания).

По умолчанию протокол *HTTP 1.1* настроен на использование постоянных соединений с конвейеризацией. Это позволяет новому запросу направляться к серверу, не до-

жидаясь окончания обслуживания других запросов. Аналогично сервер, получая новые запросы, начинает их немедленное обслуживание. Таким образом, уменьшается время установления соединения и значительно сокращается время простоя сервера.

## 2. Структура HTTP-запроса.

В *HTTP* существуют два типа сообщений: запросы и ответы, которые представляет собой совокупность текстовых символов в кодировке *ASCII*.

*HTTP*-запрос состоит из заголовка запроса и тела запроса, разделенных пустой строкой. Тело запроса может отсутствовать. Заголовок запроса состоит из главной (первой) строки запроса и последующих строк, уточняющих запрос в главной строке. Последующие строки могут отсутствовать.

Запрос в главной строке состоит из трех частей, разделенных пробелами.

Метод (иначе говоря, команда *HTTP*) может иметь следующие значения:

- *GET* – запрос документа, употребляется наиболее часто (в *HTTP/0.9* был единственным);
- *HEAD* – запрос заголовка документа, отличается от *GET* тем, что выдается только заголовок запроса с информацией о документе, а сам документ не выдается;
- *POST* – применяется для передачи данных *CGI*-скриптам, а сами данные – в последующих строках запроса в виде параметров;
- *PUT* – запрос на размещение документа на сервере, используется редко.

Унифицированный идентификатор ресурса *URI* (*Uniform Resource Identifier*) представляет собой путь к запрашиваемому документу (файлу) на сервере. Если запрашивается корневой файл из корневой директории веб-сервера, то *URI* может отсутствовать.

Если ресурс – просто какой-либо файл для считывания, сервер должен по этому запросу выдать его в теле ответа. Если же это путь к какому-либо *CGI*-скрипту, то сервер запускает скрипт и возвращает результат его выполнения. Благодаря такой унификации ресурсов для клиента практически безразлично, как ресурс представлен на сервере.

*HTTP/Версия* указывает на версию протокола *HTTP*, с которой работает клиентская программа (наиболее распространена версия *HTTP 1.1*).

Строки после главной строки запроса задают параметры запроса и имеют формат **Параметр: значение**. Наличие параметров необязательно, все строки после главной строки запроса могут отсутствовать; в этом случае сервер принимает их значение по умолчанию или по результатам предыдущего запроса. Некоторые наиболее употребительные параметры *HTTP*-запроса:

- *Connection* – соединение. Может принимать значения *Keep-Alive* и *close*. *Keep-Alive* («оставить в живых») означает, что после выдачи данного документа соединение с сервером не разрывается, и можно выдавать еще запросы. Большинство современных браузеров работают именно в режиме *Keep-Alive*, так как он позволяет за одно соединение с сервером получить как *html*-документ, так и рисунки веб-страницы. Единоразовый установленный режим *Keep-Alive* сохраняется до первой ошибки или до явного указания в очередном запросе *Connection: close*, требующего закрытия соединения после ответа на данный запрос;
- *User-Agent* – список названий и версий браузера, например, *User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; MRA 5.0 (build 02094))*;
- *Accept* – список поддерживаемых браузером типов содержимого в порядке их предпочтения данным браузером, например, *Accept image/gif, image/x-bitmap, image/jpeg, image/pjpeg, application/vnd.ms-excel, application/msword, application/vnd.ms-powerpoint,\*/\**.

Это параметр нужен для случая, когда сервер может выдавать один и тот же документ в разных форматах и используется в основном CGI-скриптами для формирования ответа, адаптированного для данного браузера;

- *Referer* – ссылка, т.е. URL, с которого перешли на этот ресурс;
- *Host* – имя хоста, с которого запрашивается ресурс;
- *Accept-Language* – поддерживаемый язык. Имеет значение для сервера, который может выдавать один и тот же документ в разных языковых версиях;
- *Accept-Encoding* – перечень поддерживаемых способов кодирования запрашиваемого документа при передаче. Например, большинство современных браузеров включают указание на возможность сжатия контента в каждый отсылаемый запрос: *Accept-encoding: gzip, deflate*.

Пример HTTP-запроса при обращении к главной странице сайта *alice.pnzgu.ru*:

*GET / HTTP/1.1*

*Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, application/x-ms-application, application/x-ms-xbap, application/vnd.ms-xpsdocument, application/xaml+xml, \*/\**

*Accept-Language: ru*

*UA-CPU: x86*

*Accept-Encoding: gzip, deflate*

*User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; MRA 5.0 (build 02094); Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1); .NET CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)*

*Host: alice.pnzgu.ru*

*Connection: Keep-Alive*

### 3. Структура HTTP-запроса.

Формат ответа очень похож на формат запроса: он также имеет заголовок и тело, разделенное пустой строкой. Заголовок также состоит из основной строки и строк параметров, но формат основной строки отличается от таковой в заголовке запроса.

Основная строка запроса состоит из трех полей, разделенных пробелами: HTTP/Версия Код ошибки Описание ошибки:

- версия протокола – аналогичен соответствующему параметру запроса;
- код ошибки – кодовое обозначение «успешности» выполнения запроса;
- словесное описание ошибки – расшифровка предыдущего кода.

Наиболее часто встречающиеся: *200 OK* – документ отправлен, *404 Not Found* – документ не найден (ошибка *URL*), клиент должен проверить правильность написания *URL*, если не помогает, значит, документ уже удален, *500 Internal Server Error* – внутренняя ошибка сервера.

Наиболее употребительные параметры *HTTP*-ответа:

• *Connection* – аналогичен соответствующему параметру запроса. Если сервер не поддерживает *Keep-Alive*, то значение *Connection* в ответе всегда *close*;

• *Content-Type* – тип содержимого ответа. В зависимости от значения *Content-Type* браузер воспринимает ответ как *HTML*-страницу (*Content-Type: text/html*), картинку (*Content-Type: image/jpeg*), простой текст (*Content-Type: text/plain*). Значение *Content-Type* для браузера аналогично значению расширения файла для Windows;

- *Content-Length* – длина содержимого ответа в байтах;



- *Last-Modified* – дата последнего изменения документа;
- *Date* – дата и время генерации ответа;
- *Server* – список названий и версий веб-сервера и его компонентов. *Server: Apache/2.0.53 (Linux/SUSE)*.

Пример *HTTP*-ответа, полученного при обращении к главной странице веб-сайта *alice.pnzgu.ru*:

```

HTTP/1.1 200 OK
Date: Thu, 09 Sep 2010 07:01:05 GMT
Server: Apache/2.0.53 (Linux/SUSE)
Accept-Ranges: bytes
Content-Length: 116
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html
<html> <head> <title> </title> <meta http-equiv="refresh" content="0;
url=/cms2"Content-Type> </head> <body> </body> </html>

```

Последние две строки представляют тело ответа, которое передается после заголовка.

### Описание протокола *FTP*

#### 1. Назначение и принцип работы протокола *FTP*

Протокол прикладного уровня передачи файлов *FTP* (*File Transfer Protocol*) используется службой передачи файлов в Интернете. Протокол *FTP* позволяет подключаться к серверам *FTP*, просматривать содержимое их каталогов и загружать файлы с сервера или на сервер; кроме того, возможен режим передачи файлов между серверами.

Протокол *FTP* для передачи данных использует транспортный протокол *TCP*, причем, в отличие от большинства других протоколов, *FTP* использует сразу два *TCP*-соединения: одно для управления, а другое для собственно передачи данных. Порт 21 используется для передачи команд, а порт 20 для передачи данных (порт для канала данных может назначаться сервером и из нестандартных портов с номерами > 1024).

Служба *FTP* построена по хорошо известной схеме клиент-сервер. *FTP*-клиент посылает запросы серверу и принимает файлы.

*FTP*-сервер обрабатывает запросы клиента на получение файла. Схема взаимодействия клиента и сервера показана на рис.2.

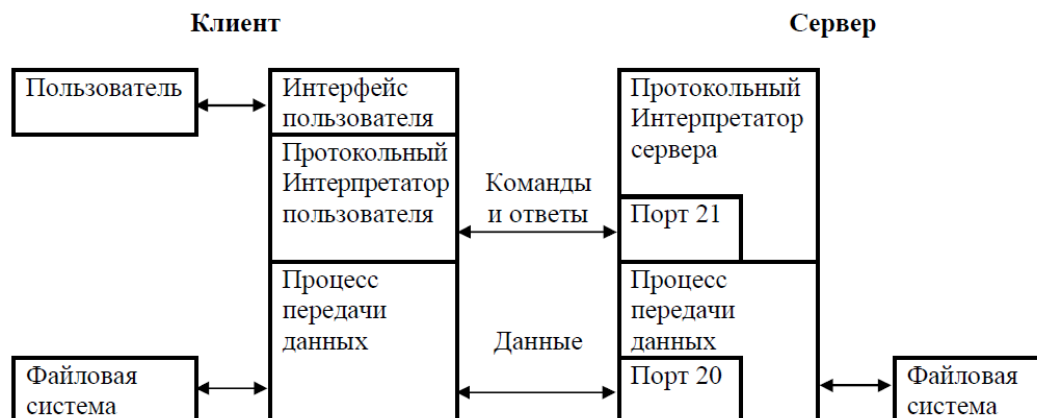


Рисунок 2. Схема взаимодействия клиента и сервера *FTP*.

Возможны два режима установления соединений в протоколе *FTP* – активный и пассивный.

Действия сервера и клиента в активном режиме:

- клиент инициирует *TCP*-соединение с динамического порта  $N$  ( $N > 1024$ ) к порту номер 21 на *FTP*-сервере;
- сервер посылает ответ на порт  $N$  клиента;
- сервер устанавливает связь для передачи данных по порту 20 на порт клиента  $N+1$ .

Действия сервера и клиента в пассивном режиме:

- клиент устанавливает связь и посылает запрос, сообщая, что надо работать в пассивном режиме, на 21-й порт сервера с динамического порта  $N$  ( $N > 1024$ );
- сервер посылает ответ и сообщает номер порта для канала данных  $P$  ( $P > 1024$ ) на порт  $N$  клиента;
- клиент устанавливает связь для передачи данных по порту  $N+1$  на порт сервера  $P$ .

Главное отличие между активным и пассивным режимами *FTP* – это та сторона, которая открывает соединение для передачи данных.

В активном режиме клиент должен принять соединение от *FTP*-сервера. В пассивном режиме клиент всегда инициирует соединение.

Хотя активный *FTP* и удобен для сервера, но попытка соединения со стороны сервера с высокими (по номеру) портами на клиенте будет заблокирована брандмауэром на стороне клиента.

Пассивный режим предназначен для соединения через брандмауэр на стороне клиента, инициатором установления всех соединений является *FTP*-клиент.

## 2. Управление обменом файлов в протоколе *FTP*

Работа *FTP* на пользовательском уровне содержит несколько этапов:

- идентификация (ввод имени и пароля);
- выбор каталога;
- определение режима обмена (поблочный, поточный, *ASCII* или двоичный);
- выполнение команд обмена;
- завершение процедуры.

Протокол *FTP* определяет запрос-ответный способ взаимодействия между клиентом и сервером, который реализуется обменом командами и ответами.

Команды передаются серверу в текстовом виде. Команда состоит из четырехбуквенного имени, за которым может следовать аргумент (там, где он требуется). Аргумент отделяется от команды пробелом.

Например, команда авторизации, передающая на сервер имя учетной записи «*anonymus*», выглядит так: *user anonymus*.

Основные команды *FTP*:

- *ABOR* – прервать предыдущую команду *FTP* и любую передачу данных;
- *CWD* имя директории – имя новой рабочей директории;
- *CDUP* – перейти на один уровень директории вверх;
- *LIST* список файлов – список файлов или директорий;
- *MODE* режим передачи – режим передачи данных (S – потоковый, B – блочный, C – сжатый);
- *PASS* пароль – пароль на сервере;

- *PORT*  $n1, n2, n3, n4, n5, n6$  – IP-адрес клиента ( $n1.n2.n3.n4$ ) и порт ( $n5*256 + n6$ );
- *QUIT* – закрыть бюджет на сервере;
- *RETR* имя файла – получить файл;
- *STOR* имя файла – положить файл;
- *SYST* сервер – возвращает тип системы;
- *TYPE* тип – указать тип файла (*A* для *ASCII*, *I* для двоичного);
- *USER* имя пользователя – имя пользователя на сервере.

Ответы сервера представляют собой код результата выполнения команды, состоящий из трех цифр в формате *ASCII*, за которым следует текст, отделенный от кода пробелом. Код предназначен для анализа *FTP*-клиентом. По нему можно однозначно определить статус выполнения команды. Текст является комментарием к коду, предназначенным для пользователя. Пример ответа сервера на команду *user anonimus: 331 Guest login ok, send your complete e-mail address as password*.

Каждая из трех цифр в коде отклика имеет собственный смысл.

Так, значения первых и вторых цифр в коде отклика означают:

- *1yz* – положительный предварительный отклик, т.е. действие началось, однако необходимо дождаться еще одного отклика перед отправкой следующей команды;
- *2yz* – положительный отклик о завершении, может быть отправлена новая команда;
- *3yz* – положительный промежуточный отклик, когда команда принята, однако необходимо отправить еще одну команду;
- *4yz* – временный отрицательный отклик о завершении, требуемое действие не произошло, однако ошибка временная, поэтому команду необходимо повторить позже;
- *5yz* – постоянный отрицательный отклик о завершении, когда команда не была воспринята и повторять ее не стоит;
- *x0z* – синтаксическая ошибка;
- *x1z* – информация;
- *x2z* – соединения. Отклики имеют отношение либо к управляющему, либо к соединению данных;
- *x3z* – отклик имеет отношение к аутентификации или командам, связанным с бюджетом;
- *x4z* – не определено;
- *x5z* – состояние файловой системы.

Третья цифра дает дополнительное объяснение сообщению об ошибке. Ниже приведены некоторые типичные отклики с возможными объясняющими строками:

- 125 Соединение данных уже открыто; начало передачи;
- 200 Команда исполнена;
- 214 Сообщение о помощи (для пользователя);
- 331 Имя пользователя принято, требуется пароль;
- 425 Невозможно открыть соединение данных;

- 452 Ошибка записи файла;
- 500 Синтаксическая ошибка (неизвестная команда);
- 501 Синтаксическая ошибка (неверные аргументы).

Перед передачей файла данных по протоколу FTP необходимо определить тип данных. Основные типы данных: *ASCII (TYPE A)* – передача текстовой информации и *IMAGE (TYPE I)* – передача бинарных файлов.

Стандарт протокола *FTP* обеспечивает три режима передачи данных по сети:

- потоковый, когда данные передаются как поток байтов;
- блочный, когда файл передается как последовательность блоков данных; каждый блок данных включает заголовок, содержащий описатель и длину блока;
- сжатый, при котором перед передачей происходит уплотнение информации с целью уменьшения объемов передаваемых данных.

### 3. Использование программы *Total Commander* как *FTP*-клиента.

Популярный файловый менеджер *Total Commander* имеет встроенного *FTP*-клиента, которого можно использовать для исследования протокола *FTP*.

Ядро работы с *FTP*-протоколом составляет окно соединения с *FTP*-сервером, которое вызывается либо по нажатию на иконке *FTP* панели инструментов, либо из пункта Соединится с *FTP*-сервером ... меню Сеть, либо через *Ctrl+F 99*.

Рассмотрим настройку *FTP*-клиента и выполнение работ с *FTP*-сервером на примере обращений к серверу *ftp://ftp.prosoft.ru/pub/Hardware/*.

Для создания учетной записи нажать кнопку **Добавить** и заполнить формы окна (рис.3).

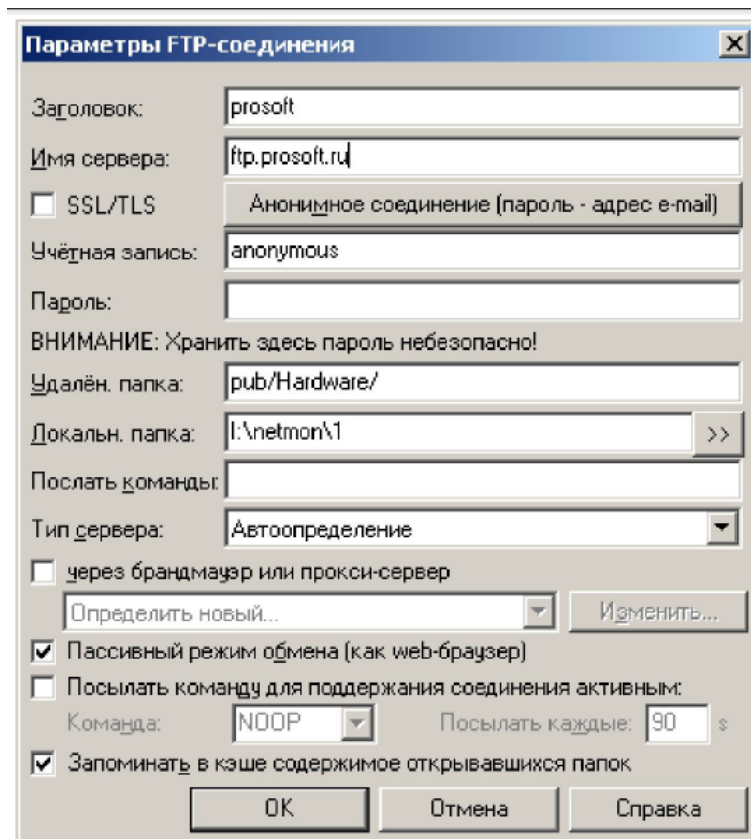


Рисунок 3. Окно *Параметры FTP-соединения*

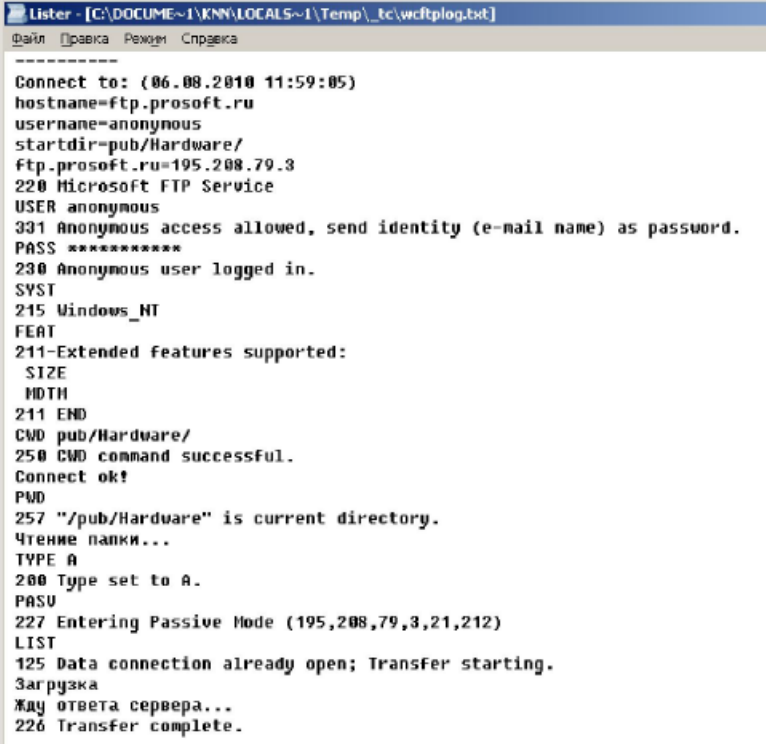
- Заголовок – имя новой учетной записи *prosoft*;
- Имя сервера – в данном случае берется из URL *ftp.prosoft.ru*;
- Учетная запись – имя *FTP*-пользователя, в данном случае *anonymous*;
- для анонимного соединения нажать соответствующую кнопку и в качестве пароля ввести адрес электронной почты;
- Удаленная папка – вводится путь к искомой папке на сервере, в данном случае *pub/Hardware* (если форму не заполнять, то откроется корневой каталог *FTP*-сервера);
- Тип сервера – установить автоопределение;
- обязательно установить Пассивный режим обмена;
- если клиент работает через прокси-сервер или брандмауэр, надо указать его *IP*-адрес и номер порта;
- заполнение остальных окон необязательно.

Для активации учетной записи нажать кнопку **ОК**.

Для обращения к серверу по созданному соединению выделить имя соединения в окне Соединение с *FTP*-сервером и нажать кнопку **Соединение**.

При успешном соединении в главном окне появится строка *FTP* с окном Режим обмена, где отображаются установленные параметры соединения, кнопка Отключение и окно протокола, двойной щелчок по которому позволяет просмотреть протокол обмена управляющей информацией при работе установленного *FTP*-соединения в редакторе

*Lister*, встроенном в программу *Total Commander* (рис. 4).



```

Lister - [C:\DOCUME~1\KNN\LOCAL5~1\Temp\_tc\wcdtprog.txt]
Файл  Правка  Режим  Справка
-----
Connect to: (06.08.2010 11:59:05)
hostname=ftp.prosoft.ru
username=anonymous
startdir=pub/Hardware/
ftp.prosoft.ru=195.208.79.3
220 Microsoft FTP Service
USER anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
PASS *****
230 Anonymous user logged in.
SVST
215 Windows_NT
FEAT
211-Extended features supported:
SIZE
MDTM
211 END
CWD pub/Hardware/
250 CWD command successFul.
Connect ok!
PWD
257 "/pub/Hardware" is current directory.
Чтение папки...
TYPE A
200 Type set to A.
PASV
227 Entering Passive Mode (195,208,79,3,21,212)
LIST
125 Data connection already open; Transfer starting.
Загрузка
Жду ответа сервера...
226 Transfer complete.

```

Рисунок 4. Протокол обмена командами и ответами

В левой и в правой панелях *Total Commander* будут открыты заданные папки на клиентской машине и *FTP*-сервере.

Для закрытия *FTP*-соединения нажать кнопку **Отключение**.

### Ход работы:

1. С помощью программы *Network Monitor* выполнить мониторинг обмена информацией при обращении к веб-сайту, для чего:

- настроить Сетевой монитор, установив фильтры по протоколам *IP*, по адресам – *IP*-адрес своего компьютера и любой сторонний;
- запустить браузер, загрузить с его помощью главную страницу с веб-сайта по указанию преподавателя, сохранить ее в файле, последовательно перейти на две внутренние страницы, закрыть браузер;
- остановить сбор данных и отобразить захваченные кадры;
- сохранить собранные кадры в файле.

2. С помощью программы *Network Monitor* проанализировать работу протокола *HTTP* при обращении к веб-сайту. Для этого:

- настроить Сетевой монитор, выполнив фильтрацию записанных данных по протоколу *HTTP*;
- проанализировать заголовки запросов и ответов, определить режимы работы браузера и сервера;
- открыть с помощью браузера сохраненную страницу и по ее *HTML*-коду определить количество и назначение запросов к серверу, выполняемых при ее открытии, проверить по содержанию перехваченные Сетевым анализатором пакеты.

3. В отчете привести:

- содержание заголовков пакетов при обращении к веб-сайту, дать объяснение всем параметрам;
- текст с *HTML*-кодом открытой страницы, указать, какие теги вызвали запрос к веб-серверу, перечислить назначение и наименование файлов, загруженных при открытии веб-страницы.

4. С помощью программ *Network Monitor* и *Total Commander* выполнить мониторинг обмена информацией при обращении к *FTP*-серверу, для чего:

- в программе *Total Commander* создать и настроить *FTP*-соединение, адрес сервера и параметры соединения получить у преподавателя;
- настроить Сетевой монитор, установив фильтры по протоколам *IP*, по адресам – *IP*-адрес своего компьютера и любой сторонний;
- запустить захват данных в программе *Network Monitor*;
- запустить созданное *FTP*-соединение в программе *Total Commander*;
- открыть в редакторе *Lister* протоколирование работы установленного *FTP*-соединения;
- зайти на заданную *FTP*-серверу папку и скачать один файл;
- закрыть *FTP*-соединение;
- остановить сбор данных и отобразить захваченные кадры;
- сохранить собранные кадры в файле;
- сохранить в файле результаты протоколирования работы установленного *FTP*-соединения, открытые редактором *Lister*.

5. С помощью программы *Network Monitor* проанализировать работу при обращении к *FTP*-серверу, для чего:

- настроить Сетевой монитор, выполнив фильтрацию записанных данных по протоколу *FTP*;

- проанализировать заголовки запросов и ответов, определить режимы работы клиента и сервера;

- установить соответствие с информацией, полученной с помощью программы Total Commander.

6. В отчете привести:

- содержание заголовков пакетов при обращении к *FTP*-серверу на этапах идентификации, выбора каталога, чтения файла, завершения процедуры;

- результаты протоколирования работы установленного *FTP*-соединения с комментариями.

### **Вопросы для самопроверки.**

1. Какую роль выполняет протокол *HTTP* в сети *Internet*?

2. Какие параметры заголовка протокола *HTTP* определяют список допустимых форматов ресурса?

3. Что такое *URI*?

4. Какие параметры *HTTP*-запроса определяют поддерживаемый язык?

5. В чем особенность работы веб-сервера при обслуживании запроса в режиме *Keep-Alive*?

6. Как отфильтровать информацию, принадлежащую протоколу *HTTP* в захваченных данных?

7. Каково назначение протокола *FTP*?

8. В чем отличие активного и пассивного режимов работы протокола *FTP*?

9. Что определяет команда *TYPE A*?

10. Почему протокол *FTP* использует два *TCP*-соединения для передачи данных?

11. Как выполняется в протоколе *FTP* аутентификация пользователя?

## Лабораторная работа №4 Анализатор протоколов CommView.

**Тема работы:** анализатор протоколов CommView.

**Цель работы:** анализ различных протоколов и приобретение практических навыков в использовании программных средств, позволяющих контролировать сетевой трафик, на примере программы CommView.

**Теоретическая часть:**

Рассмотрим программное средство CommView (<http://www.tamos.ru/products/commra/>) компании TamoSoft в качестве анализатора протоколов. Для работы в учебных целях будем использовать 30-дневную пробную версию программы. В рамках этого срока функционал CommView не ограничен, и программа может реализовать большинство требуемых функций анализатора.

Перед началом работы анализатора необходимо выбрать сетевой интерфейс (сетевую карту) через меню «Настройки/установки» и начать захват сетевых пакетов через меню «Файл/начать захват».

Если захват пакетов успешно стартовал в главном окне программы, то отразится сетевая статистика. Пример приведен на рис.5.

Локальный IP	Удаленный IP	Входящие	Исходящие	Направление	Сессии	Порты	Имя хоста	Байт
10.70.19.45	239.255.255.250	0	3	Транз.	0	1309,1900,1340,1633		525
10.70.19.76	239.255.255.250	0	1	Транз.	0	1145,1900		175
10.70.19.77	239.255.255.250	0	3	Транз.	0	1127,1900,1133,1136		525
10.70.19.78	239.255.255.250	0	1	Транз.	0	49203,1900		175
10.70.19.97	239.255.255.250	0	2	Транз.	0	1046,1900		350
10.70.19.109	239.255.255.250	0	1	Транз.	0	1040,1900		175
10.70.19.164	239.255.255.250	0	2	Транз.	0	1793,1900,1038		350
10.70.19.235	213.180.204.11	0	3	Исход.	0	http	yandex.ru	186
10.70.19.235	192.168.200.1	166	266	Вход.	27	1525,1526,1527,1528,1529,1...		98 360
10.70.19.235	72.36.139.138	0	3	Исход.	0	http	protonhosting.com	186
10.70.19.235	10.70.19.254	5	1	Исход.	0			522
10.70.19.235	10.70.19.193	23	30	Вход.	1	netbios-ns,netbios-dgm,1564		5 120
10.70.19.235	10.70.19.99	0	2	Исход.	0			148
10.70.19.235	10.8.0.1	4	0	Вход.	0	1454,1400		714
10.70.19.235	239.255.255.250	0	24	Исход.	0			2 416
10.70.19.235	10.70.19.164	1	13	Вход.	0	netbios-ns,netbios-dgm		3 107

Рисунок 5. Сетевая статистика пакетов ЛВС.

Статистика на рис. 5 отражает получателей сетевых пакетов (колонка удаленный IP), отправителей (локальный IP), число пакетов, направление, порты по которым происходит обмен и другую информацию по сетевым пакетам.

На вкладке «Пакеты» главного окна программы можно получить детальную информацию по содержанию сетевого пакета, выбрав его из перечня как на рис. 6. На рисунке выбран пакет с номером 67 протокола IP/TCP. В центральной части окна указаны основные параметры пакета (внутренний номер пакета, протокол, МАК-адреса и др.) Справа указана детальная информация с декодированием структуры пакета, которая приведена в нижней части окна.



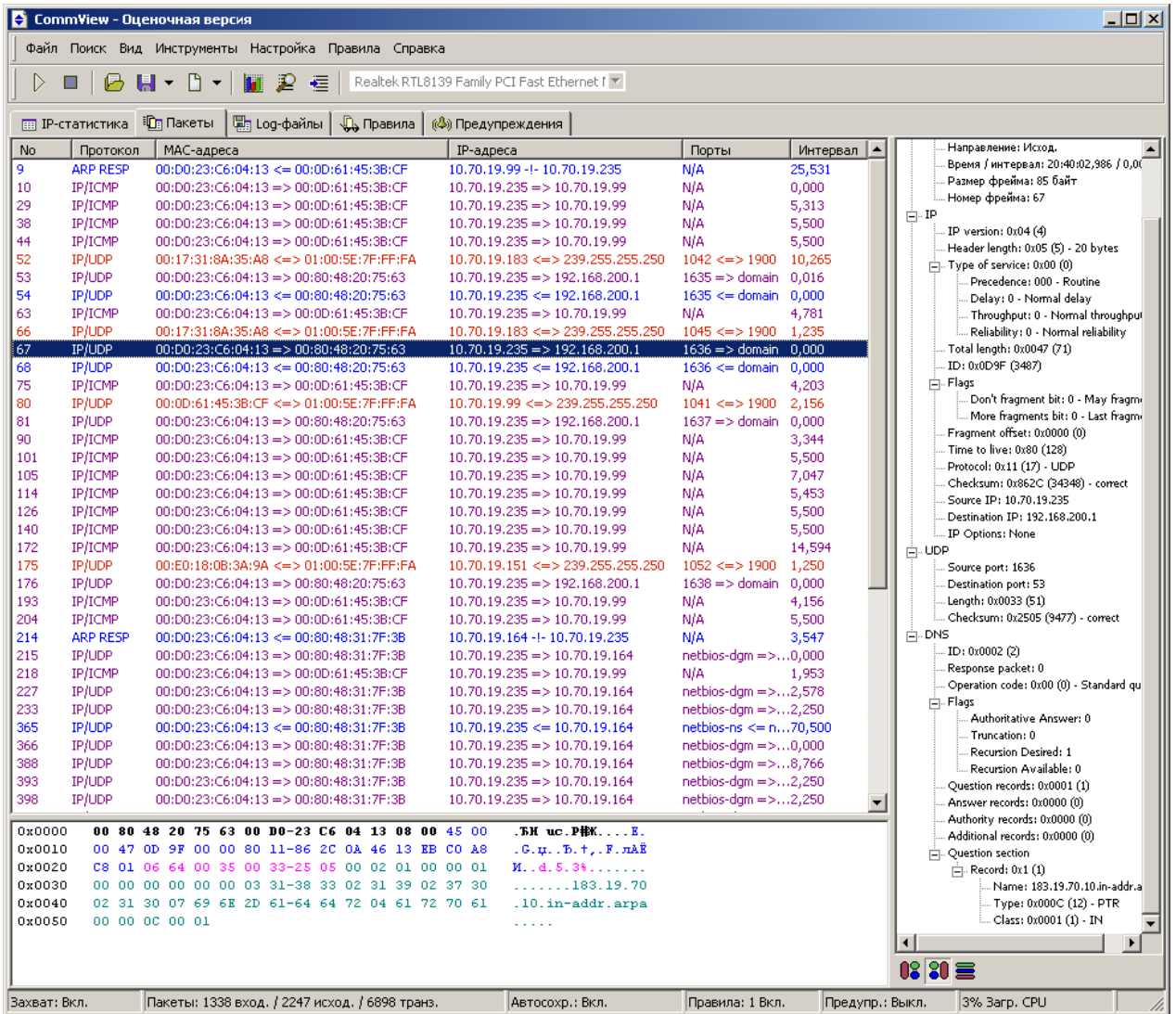


Рисунок 6. Детальное изучение сетевого пакета.

Часто анализ всей статистики малоэффективен из-за большого числа не требуемых нам для исследования пакетов. Это потребует для настройки некоторых правил фильтрации сетевых пакетов, для упрощения задачи мониторинга и анализа.

Пример настройки правил приведен на рис. 7. В качестве примера создано правило для игнорирования при сборе сетевых пакетов с широковещательным MAC- адресом. Для этого необходимо перейти на вкладку «Правила»/ «MAC-адреса» далее поставить галочку «Включить правила для MAC-адресов» и ввести MAC-адрес FF FF FF FF FF FF. Указать «Добавить запись» в любом направлении и в качестве «Действия» игнорировать. С более подробным синтаксисом написания правил можно ознакомиться в справочной системе программы (через главное меню «Справка») или нажав клавишу F1.

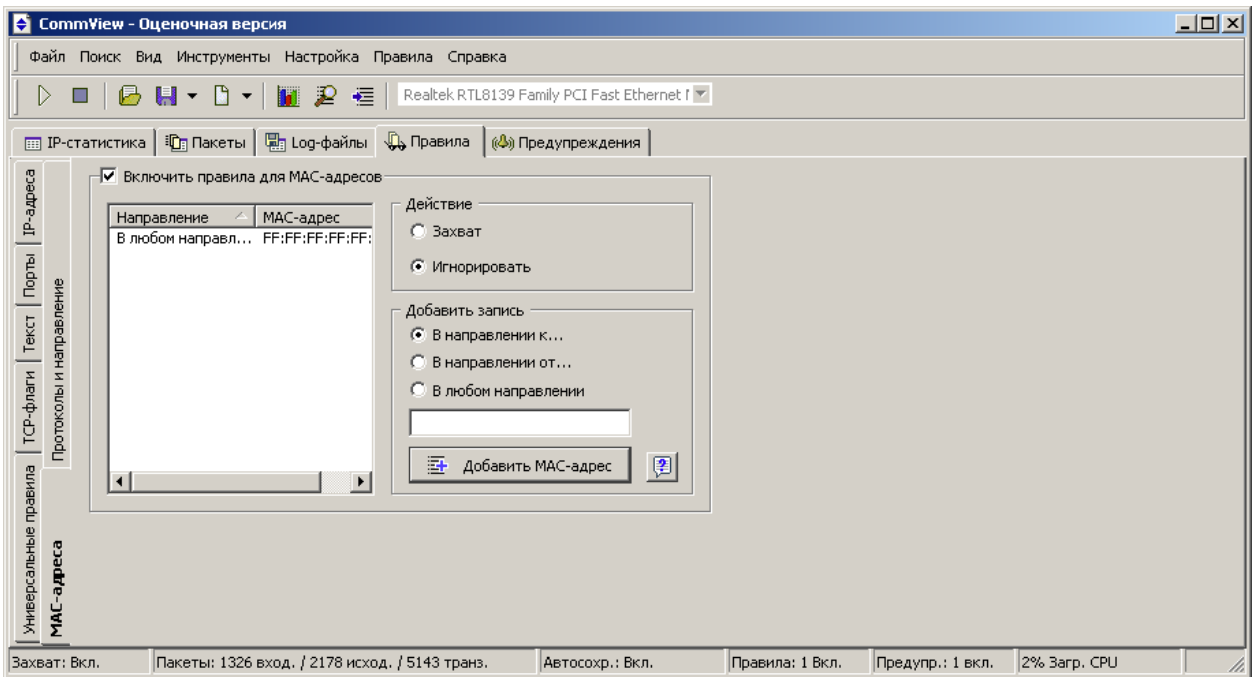


Рисунок 7. Настройка правил фильтрации сетевых пакетов.

Иногда возникает необходимость реакции на появление некоторого условия, например при адресации от одного IP-адреса к другому. В этом случае потребуется настроить предупреждения на соответствующей вкладке. Синтаксис предупреждения целиком аналогичен созданию правила. Так например для возникновения предупреждения при адресации IP=172.16.4.21 к IP=172.16.4.22 необходимо добавить в текст предупреждения: (sip=172.16.4.21 and dip=172.16.4.22) or (sip=172.16.4.22 and dip=172.16.4.21) и указать сообщение для показа (на рисунке это «Предупреждение1»). Таким образом можно отслеживать пакеты по некоторым условиям и реагировать на них (рис.8).

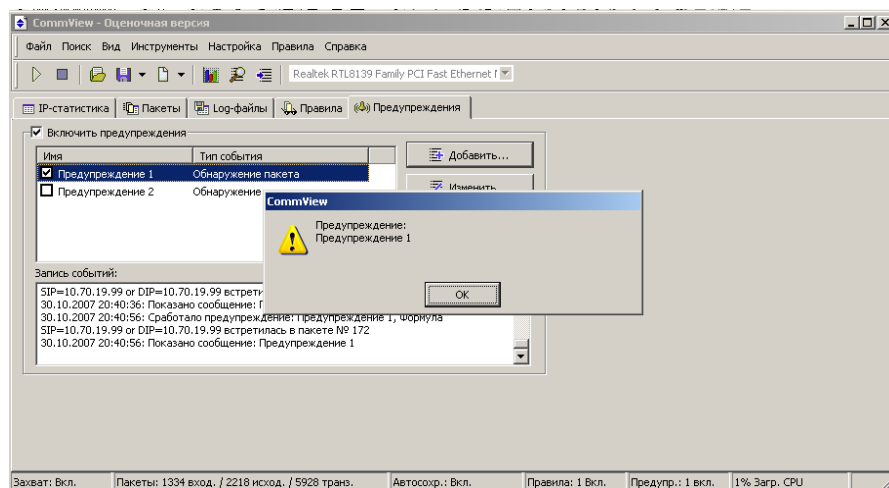


Рисунок 8. Настройка и работа предупреждений.

Программа позволяет настроить число отражаемых пакетов в главном окне программы, остальная информация будет записана в файлы журнала и помещена в сводную статистику.

При необходимости просмотра все пакетов это можно сделать через главное меню «Файл \ Просмотр log- файлов». Параметры log-файлов можно настроить на вкладке log-файлы.

Для отражения общей картины по сетевым пакетам можно воспользоваться расширенной статистикой через главное меню «Вид \ Статистика» (рис. 9).

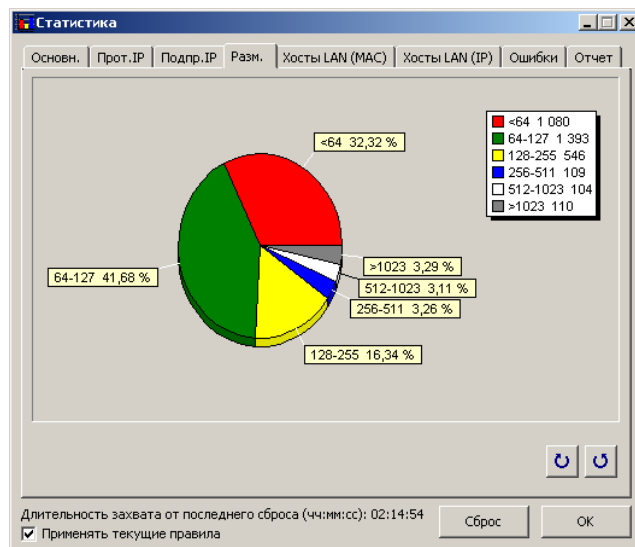
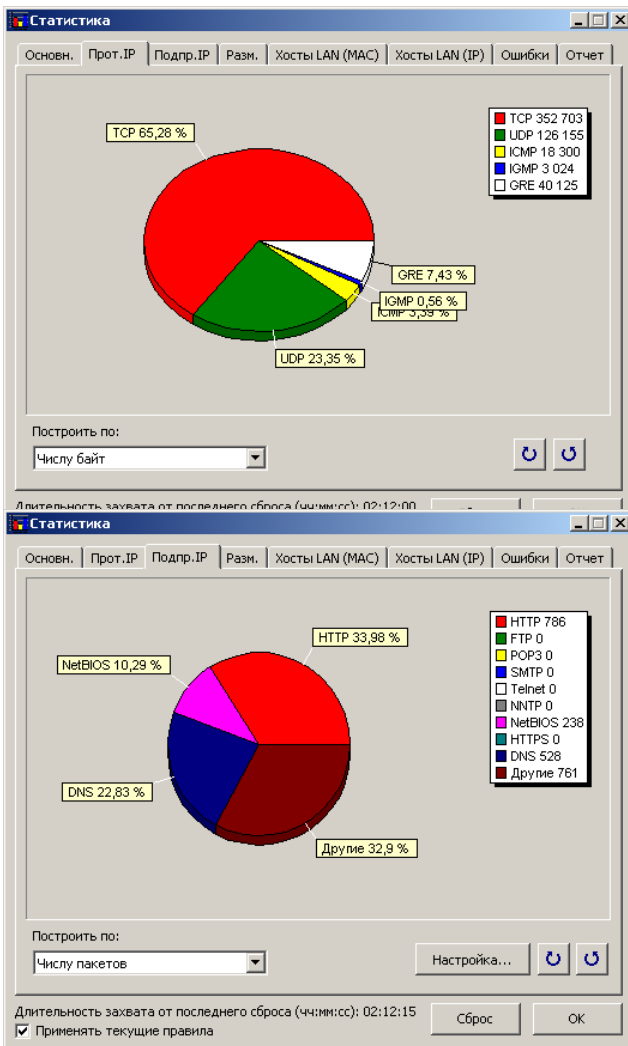


Рисунок 9. Детальная статистика по сетевым пакетам.

В программе возможно на основе данных статистики сгенерировать HTML-отчет. Для этого в окне «Статистика» на вкладка «Отчет» необходимо проверить и при необходимости настроить параметры отчета и нажать кнопку «Просмотреть». Пример формы настройки отчета приведен на рис 10.

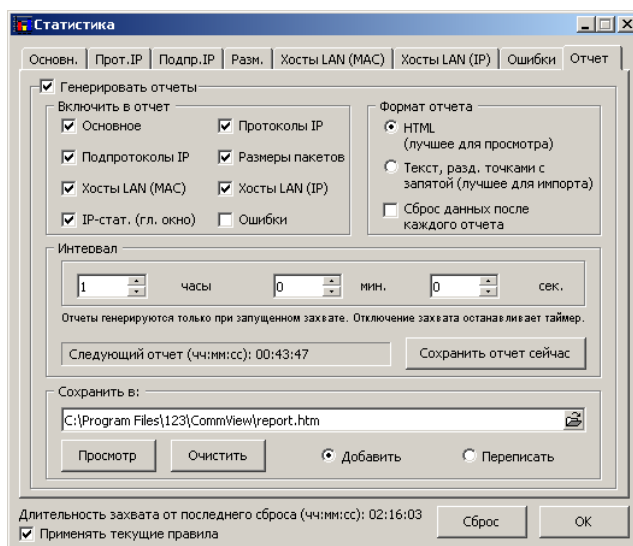


Рисунок 10. Настройка HTML-отчета по статистике.

После нажатия кнопки «Просмотр» откроется HTML-страница с данными статистики с учетом выбранных параметров настройки. Пример страницы приведен на рис. 11.

Статистический отчет

Сгенерировано [CommView](#), 28.10.2007 в 18:40:58 Длительность захвата от последнего сброса (чч:мм:сс): 00:55:37

**Основн.**

Среднее значение пакетов в сек.	16
Среднее значение байт в сек.	4 655
Всего пакетов	55 044
Всего байт	15 530 492

Показатель \ Направление	Входящие	Исходящие	Транзитные
Пакеты	13 400	14 459	27 185
Байт	11 084 179	1 530 445	2 915 868
Байт в сек.	3 323	459	874

**IP-протоколы**

Протокол	Пакеты	Процент
TCP	26372	57,82
UDP	19091	41,86
ICMP	82	0,18
IGMP	60	0,13
255[unknown]	4	0,01

**IP-подпротоколы**

Подпротокол	Пакеты	Процент
HTTP	84	0,18
FTP	18	0,04
POP3	0	0,00
SMTP	0	0,00
Telnet	0	0,00
NNTP	0	0,00
NetBIOS	17955	39,49
HTTPS	18	0,04
DNS	36	0,08
Другие	27352	60,16

**Распределение пакетов по размеру**

Рисунок 11. HTML-отчет по сетевой статистике.

### Задание для самостоятельной работы.

1. Установить программу CommView.
2. Установить фильтры по протоколам «все», по адресам – MAC-адрес своего компьютера и любой сторонний. Запустить захват, контролировать трафик сети.

3. В отчете привести собранную статистику мониторинга сети.
4. Настроить систему фильтрации сетевых пакетов по условию (фильтр)
  - по IP- адресам (sip, dip);
  - по номерам портов (sport, dport);
  - типу протокола;
  - MAC-адресу;
  - комбинированный фильтр по IP и номеру порта;
  - комбинированный фильтр по типу протокола и номеру порта.
5. Определить правила, предупреждения.
6. Исследовать сбор и анализ статистики по сетевым пакетам ЛВС.
7. Всю работу отразить в отчете:
  - порядок действий для настройки захвата пакетов с помощью фильтра и без него;
  - статистика работы анализатора за 10 минут работы;
  - анализ структуры 2-3 пакетов разных протоколов (ICMP, UDP, TCP);
  - комбинированный фильтр по IP и номеру порта.

## Лабораторная работа №5 WWW-сервис, поиск информации в Интернет.

**Тема работы:** WWW-сервис, поиск информации в Интернета с использованием различных поисковых систем и средств поиска. Поиск людей в Интернет.

**Цель работы:** изучение работы поисковых систем, умение быстро находить желаемую информацию, используя возможности различных поисковых систем и средств поиска (простой, расширенный, специальный и контекстный поиск), знакомство с "белыми страницами" Интернет, умение сохранять найденную информацию на диске.

### Теоретическая часть:

World Wide Web представляет собой множество Internet-серверов, содержимое которых, представлено в виде Web-страниц или просто страниц. Web-страница – это файл (совокупность файлов), при создании которого используется язык HTML, позволяющий хранить и организовывать информацию, предназначенную для размещения в Интернете. Web-страница может содержать текстовую, графическую, звуковую и видео-информацию, баннеры, формы, встроенные программы на языке Java и фреймы.

### Работа с программой Microsoft Internet Explorer.

Окно Internet Explorer содержит **главное меню, кнопки управления, адресную строку**, в центральной части отображается вызванный ресурс Internet, внизу окна имеется **строка состояния**.

**Главное меню** содержит команды по управлению Internet Explorer.

**Строка кнопок** обычно содержит следующие кнопки:



кнопки **Назад** (на предыдущую страницу) - **Вперед** (на следующую страницу)



кнопка **Остановить** - нажимается при необходимости прервать загрузку страницы



кнопка **Обновить** – служит для повторной загрузки текущей Web-страницы



кнопка **Домой** – возвращение на домашнюю страницу



кнопка **Поиск** – выполнение поиска в сети Интернет



кнопка **Избранное** – отображается список отобранных пользователем ресурсов.

Папка **Избранное** - это своего рода «записная книжка» путешественника по Сети. В нее вы можете складывать ссылки на интересные страницы, чтобы потом открыть их одним щелчком мыши. Эти сохраненные ссылки называются закладками.

Открыть папку **Избранное** можно двумя способами:

- щелкнуть по кнопке Избранное. В этом случае коллекция ссылок будет доступна в виде отдельной панели в левой части браузера. Чтобы закрыть эту панель нужно повторно щелкнуть по кнопке Избранное.

- щелкнуть по Избранное в главном меню. Тогда ссылки откроются в виде обычной системы выпадающих меню.

Добавлять ссылки в Избранное можно следующими способами:

- с помощью пункта Добавить в Избранное меню Избранное (или панели Избранное),

- «перетаскивая» их мышью из окна браузера на панель **Избранное** справа,
- щелкнув правой кнопкой мыши по странице и, выбрав в контекстном меню **Добавить в Избранное**,
- нажав Ctrl D.

Закладку при добавлении можно переименовывать.

Рекомендуется в папке **Избранное** организовывать ряд вложенных папок, рассортированных по тематикам ссылок. Это можно сделать используя кнопку **Создать папку** в окне **Добавление в Избранное**. В папке **Избранное** можно навести порядок, используя кнопку **Упорядочить Избранное** на панели **Избранное**.

На жестком диске адрес папки Избранное для Windows XP – это C:\Documents and Settings\User\Избранное.



кнопка **Журнал** – просмотр хронологии работы.

Нажав эту кнопку, открывается панель **Журнал**. Это тоже коллекция ссылок, но создает их не пользователь, а браузер. Каждый раз, когда мы открываем новую страницу, программа фиксирует это в Журнале. По желанию ссылки на страницы могут быть отсортированы в разных режимах. Переключиться между режимами можно с помощью кнопки **Вид** в левом углу панели **Журнал**.

#### Средства навигации в WWW:

- использование адресной строки,
- использование кнопок навигации,
- гиперссылки,
- использование закладок папки Избранное,
- использование ссылок из Журнала.

#### Сохранение Web-страниц.

Загруженную в окно обозревателя Web-страницу, можно сохранить на жестком диске для дальнейшего использования. Для этого нужно выполнить команду: **Файл/Сохранить как**. В списке **Тип файла** на выбор оказываются следующие форматы:

- веб-страница полностью,
- веб-страница, только HTML (скачивается быстрее, но без картинок),
- текстовый файл,
- веб-архив (mht) – удобный формат, вся веб-страница, вместе с рисунками и оформлением помещается в один файл.

#### Сохранение рисунков Web-страницы.

Для сохранения любого рисунка со страницы необходимо:

- в зоне любого рисунка щелкнуть правой кнопкой мыши,
- из контекстного меню выбрать **Сохранить рисунок как**,
- указать в появившемся окне диск, папку, имя.

#### Сохранение фрагментов текста Web-страницы.

Используется буфер обмена. Для сохранения текста нужно:

- выделить мышью на странице выделяемый текст,
- щелкнуть правой кнопкой мыши в любой выделенной зоне,
- в контекстном меню выбрать **Копировать**,
- перейти в текстовый редактор Word (или другой текстовый редактор),
- выбрать курсором нужную позицию и **Правка/вставить (Ctrl V)**.

Выбор шрифта.

Если загруженная в окно страница отображается в нечитабельном виде, то необходимо выбрать шрифт для данной страницы: **Вид/Кодировка**.

### Настройки программы Internet Explorer.

Меню настроек программы Internet Explorer можно вызвать: **Сервис/Свойства обозревателя.**

На вкладке **Общие** можно задать **стартовую (домашнюю) страницу**, с которой будет начинаться путешествие по Интернет.

Уточнить, в течении какого времени будут храниться ссылки в Журнале можно с помощью вкладки **Журнал.**

Используя вкладку **Содержание** можно ограничить доступ к сайтам, содержащим насилие, порнографию и т.д. Здесь, же в **Личные данные** можно ввести свои данные для **Автозаполнения форм** или каких-то анкет на WWW-страницах.

На вкладке **Дополнительно/Мультимедиа** можно отключить любой из переключателей: **Отображать рисунки, Воспроизводить анимацию, Воспроизводить видео, Воспроизводить звуки.** Чтобы восстановить возможность загрузки всех элементов нужно использовать **Восстановить значения по умолчанию.**

### Поисковые системы

**Поисковый каталог** - поисковая система с классифицированным по темам списком аннотаций со ссылками на web-ресурсы. Такая классификация, как правило, проводится людьми.

Поиск в каталоге очень удобен и проводится посредством последовательного уточнения тем, разделов, подразделов и т.д. Кроме этого, каталоги поддерживают возможность быстрого поиска определенной категории или страницы по ключевым словам с помощью локальной поисковой машины. База данных ссылок (индекс) каталога обычно имеет ограниченный объем, заполняется вручную специалистами, создающими каталог. Некоторые каталоги используют автоматическое обновление индекса.

Результат поиска в каталоге представляется в виде списка, состоящего из краткого описания (аннотации) документов с гипертекстовой ссылкой на первоисточник.

Адреса популярных каталогов:

**Yahoo** ([www.yahoo.com](http://www.yahoo.com)) – признан наиболее популярным каталогом во всем мире. Возможен поиск на русском языке.

Российские каталоги:

«**List.Ru**» ([www.list.ru](http://www.list.ru)),

«**Созвездие Интернет**» ([www.stars.ru](http://www.stars.ru)),

«**Russia on the Net**» ([www.ru](http://www.ru)) и другие.

**Поисковая машина** - поисковая система с формируемой программным "роботом" базой данных, содержащей информацию об информационных ресурсах.

Все поисковые машины, предназначенные для сети Интернет, имеют более или менее схожие принципы работы. Поиск в такой системе проводится по запросу, составляемому пользователем, состоящему из набора ключевых слов или фразы, заключенной в кавычки. Индекс формируется и поддерживается в актуальном состоянии роботами-индексировщиками.

Некоторые зарубежные поисковые машины:

**Fast Search** ([www.alltheweb.com](http://www.alltheweb.com)),

**Alta Vista** ([www.altavista.com](http://www.altavista.com)),

**Google** ([www.google.com](http://www.google.com)) – возможен поиск на русском языке,

**Northern Light** ([www.northernlight.com](http://www.northernlight.com))

Российские поисковые машины:

**Рамблер** ([www.rambler.ru](http://www.rambler.ru)),

**Апорт2000** ([www.aport.ru](http://www.aport.ru)),

**Яндекс** ([www.yandex.ru](http://www.yandex.ru)),

**Lupa.ru** ([www.lupa.ru](http://www.lupa.ru)).



*Релевантность* - это соответствие результатов поиска сформулированному запросу.

#### Средства поиска информации.

Каждая поисковая машина использует свой язык запросов, поэтому при использовании разных поисковых систем надо знать особенности каждой. Но есть общий принцип, согласно которому все команды можно разделить на три группы:

- команды простого поиска,
- команды расширенного поиска,
- команды специального поиска.

Рассмотрим приемы формирования запросов на примере поисковой машины Апорт2000 ([www.aport.ru](http://www.aport.ru)).

#### **Средства простого поиска.**

Поиск группы слов. Большинство российских поисковых систем, как и поисковая машина Апорт, воспринимает группу слов так, как будто между ними стоит союз И. Например, *лица в розыске*.

Поиск словоформ. В большинстве случаев Апорт позволяет находить разные словоформы, например, запрос *допрос несовершеннолетнего* можно было сформулировать *несовершеннолетний допрос*.

Поиск точных форм. Иногда нужно от словоформ отказаться. Допустим нас интересует слово *престол*, а не *престольный* и не *первопрестольный*. В этом случае Апорт использует восклицательный знак, например: *!престол*.

Роль заглавных букв. Для большинства поисковых систем заглавные буквы воспринимаются буквально. Так, например поиск по словам укажет только на те страницы, где встречаются слова *Красная Шапочка*. А поиск по запросу *красная шапочка* укажет на страницы, где встречаются слова *Красная Шапочка*, и *Красная шапочка*, и *красная Шапочка*, и *красная шапочка*.

Значение подстановочных символов. Апорт позволяет использовать символ \* вместо любого количества символов до конца слова. Например, если нужны документы, в которые входят слова *Татарская республика* и , то можно использовать только: *республика Татарс\**.

Учет зарегистрированных слов. Зарегистрированными словами (стоп-словами) считаются те, которые не учитываются при поиске. Обычно к ним относятся все слова, которые меньше 4 символов – предлоги, союзы, местоимения, артикли и т.д. Так, например в запросе *все о юристах* будет учитываться только слово *юристах* и его словоформы. В специализированных поисковых системах к зарегистрированным могут относиться слова, характерные для этих систем. Например, если поисковая система ориентирована на поиск книг, то вероятнее всего слово *книга* в ней при обработке запроса не будет учитываться.

Средства контекстного поиска. Взятая в кавычки фраза ищется буквально, то есть так, как она написана, без словоформ. Это очень полезный вид поиска.

#### **Средства расширенного поиска.**

В расширенном поиске кроме ключевых слов, можно использовать простейшие логические операторы и логические скобки.

Оператор И (AND,&,+). С помощью этого оператора объединяются 2 или более слов. В поисковой системе Апорт можно этим оператором не пользоваться, т.к. в ней списки слов и так воспринимаются, как будто между ними стоит оператор И. Но так бывает не всегда (например Alta Vista по умолчанию считает, что ключевые слова связаны соотношением ИЛИ).

Оператор ИЛИ (OR,|). Этот оператор обеспечивает поиск по любому из слов группы, например *образование ИЛИ обучение*.

Логические скобки. Скобки необходимы, когда важен порядок действия логических операторов, например, *Ломоносов ИЛИ(Михаил И Васильевич)*.

Оператор НЕТ (-,NOT). Этот оператор употребляется, когда из результатов поиска нужно исключить какое-либо ключевое слово. Например, по запросу *допрос NOT несовершеннолетний*, будут возвращены документы, где есть слово *допрос*, но нет слова *несовершеннолетний*.

Поиск с указанием расстояния. Этот вид поиска позволяет указать, на каком расстоянии друг от друга могут располагаться слова в документе. Например, *сл5(приостановление следствия)*. В данном случае оба слова должны принадлежать одной группе, длиной не более 5 слов.

В таблице 1 приведены логические операторы для некоторых поисковых машин.

Таблица 1.

Оператор	Действие	Rambler	Yandex	Aport
<b>И</b>	Ищет страницы со <i>Словом1 и Словом2</i>	<i>Слово1 AND Слово2</i> <i>Слово1 &amp; Слово2</i> <i>Слово1 Слово2</i> (пробел)	<i>Слово1 AND Слово2</i> ; <i>Слово1 &amp; Слово2</i> ; <i>Слово1 Слово2</i> (пробел); <i>+Слово1 + Слово2</i>	<i>Слово1 AND Слово2</i> ; <i>Слово1 &amp; Слово2</i> ; <i>Слово1 Слово2</i> (пробел); <i>+Слово1 + Слово2</i> ; <i>Слово1 И Слово2</i>
<b>ИЛИ</b>	Ищет страницы со <i>Словом1 или Словом2</i>	<i>Слово1 OR Слово2</i> <i>Слово1   Слово2</i>	<i>Слово1 OR Слово2</i> <i>Слово1   Слово2</i>	<i>Слово1 OR Слово2</i> <i>Слово1   Слово2</i> <i>Слово1 ИЛИ Слово2</i>
<b>НЕТ</b>	Ищет страницы, в которых <b>нет</b> Слова	<i>NOT Слово</i> ; <i>! Слово</i>	<i>NOT Слово</i> ; <i>-Слово</i> ; <i>~Слово</i>	<i>NOT Слово</i> ; <i>-Слово</i> <i>НЕТ Слово</i>

### Средства специального поиска.

Поиск с указанием даты. Оператор *data* позволяет выполнять поиск документов по датам, например *data=01/02/2000 образование*. Поэтому запросу будут выданы документы, содержащие слово *образование* и датированные 1 февраля 2000 г.

Поиск по ссылкам на определенный адрес URL. Оператор *url* осуществляет поиск по ссылкам на названный ресурс. Например, по запросу *url=www.ulsu.ru* ,будут выданы все документы проиндексированные на сервере *www.ulsu.ru*.

Поиск по заголовкам. Команда **title** позволяет находить страницы по их заголовкам. Например, по запросу **title=(Нургалиев Р.Г.)** будут найдены документы, в заголовках которых присутствует фамилия Министра МВД.

Поисковые системы могут иметь страницу **Расширенный поиск**, где, задавая параметры поиска, можно построить запрос, не используя напрямую язык запросов. Поэтому перед широким применением любой поисковой системы полезно внимательно ознакомиться с принятыми в ней правилами поиска.

### **Поиск людей - "Белые страницы".**

Каталоги, включающие в себя адреса электронной почты и почтовые адреса, а также телефонные номера, часто называются «белыми страницами»

Некоторые **зарубежные поисковые системы**, позволяющие найти людей по фамилии или e-mail:

[www.bigfoot.com](http://www.bigfoot.com) – осуществляет E-mail Search; Find Friends; White Pages; Find People и т.д.;

[www.addresses.com](http://www.addresses.com) – имеется каталог E-mail;

[www.yahoo.com/search/people](http://www.yahoo.com/search/people);

[www.whowhere.com](http://www.whowhere.com).

Некоторые российские поисковые службы:

<http://whitepages.rin.ru> – Российская Информационная Сеть. При выборе нужного человека, предлагается ему послать письмо со своими личными данными и вопросами к нему. Есть регистрация.

[www.mates.ru](http://www.mates.ru) – Каталог выпускников школ и вузов разных лет. Есть рубрики: поиск людей, найди меня, одноклассники, однокурсники, регистрация и т.д. О людях содержится много данных, но конкретный e-mail не выдается, предлагается связаться с человеком, и если он пожелает, то ответит Вам.

<http://poisk.goon.ru> – Поиск людей – одноклассников, родственников и т.д.

[www.sherlok.ru](http://www.sherlok.ru) – платный поиск людей по фамилиям, телефонам и др. данным.

[www.centroisk.narod.ru](http://www.centroisk.narod.ru) - платный поиск людей.

Один из самых надежных способов нахождения информации о персональных контактах, если Вы знаете организацию, к которой относится человек, - это обратиться к «начальной странице» организации (часто есть ссылка на внутренний каталог).

Для перевода информации из зарубежных сайтов, можно воспользоваться переводчиками on-line (в режиме реального времени):

<http://www.perevodov.net> – Переводчики on-line. Внизу главной страницы выберите **бесплатные переводы on-line/ англо-русский**. Есть возможность перевода предложений.

[www.rambler.ru/dict/](http://www.rambler.ru/dict/) - словарь на Рамблере.

### **Ход работы:**

1. Зайдите на поисковую систему **Rambler**. Используя контекстный поиск, найдите официальный сайт Санкт-Петербургского государственного университета. Ответьте на следующие вопросы:

2. Кто сегодня является ректором университета?

Найдите страничку юридического факультета; назовите известных выпускников факультета. Ответы на эти вопросы включите в файл-отчет, главную страницу сайта добавьте в папку Избранное.

3. Зайдите на поисковую систему **Апорт** или др.

Найдите сайт экскурсионного бюро Кремля. Зайдите на сайт Кремлевского музея. Ответьте на вопросы:

Чему равна стоимость билета на посещение Оружейной палаты?

Какие экспозиции представлены в музее Кремля?

Ответы на эти вопросы включите в файл-отчет, главную страницу сайта Музеи Московского Кремля добавьте в папку Избранное.

4. Используя поисковый каталог [www.list.ru](http://www.list.ru), зайдите на сайт Музеи Московского Кремля, При выполнении этого задания, обязательно используйте свойства каталогов: тема -> подтема -> и т.д. Порядок выполнения обязательно зафиксируйте в отчете. Сравните качество, быстродействие, удобство поиска с п.2, выводы включите в отчет.

5. Используя поисковую систему **Google**, узнайте с какого года выходит японская газета либерального направления АСАХИ в г. Токио? Ответ на вопрос и ссылки на источник информации запишите в файл-отчет.

6. Зайдите на любую поисковую систему и найдите материал, который поможет ответить вам на следующие вопросы:

- Где и когда родился Андрей Дмитриевич Сахаров? В каком институте, и на каком факультете он обучался?
- Перечислите основные этапы жизни и деятельности А.Д.Сахарова. Почему его имя вошло в историю
- Найдите и сохраните в отчете фотографию А.Д.Сахарова.

Используйте средства специального поиска, поиск в найденном или расширенный поиск. Ответы на эти вопросы включите в файл-отчет.

7. Изучите «Рамблер Top 100». Просмотрите Рейтинг/ Экспертиза на [www.rambler.ru](http://www.rambler.ru).

На главной странице Рамблера, используя средства специального поиска по дате, найдите страницы, где содержится Федеральный закон №24-ФЗ от 20 февраля 1995г. Об информации, информатизации и защите информации. Фрагмент этого закона включите в отчет.

8. На [www.addresses.com](http://www.addresses.com) попытайтесь найти информацию об Александре Иванове. Используйте <http://www.perevodov.net>. Результаты поиска зафиксируйте в отчете.

9. Изучите [www.mates.ru](http://www.mates.ru). Найдите информацию о выпускниках МГУ 1997 года. Результаты поиска зафиксируйте в отчете.

10. Зарегистрируйтесь на <http://whitepages.rin.ru>, попытайтесь найти друг друга. Страницу с результатами поиска сохраните (сохранение веб-страницы, только HTML) на диске в личной папке.

11. Изучите персональные страницы Губернатора Московской области и Министра образования Московской области, используя сайт правительства Московской области ([www.mosreg.ru](http://www.mosreg.ru)). Сохраните одну из найденных веб-страниц, как веб-архив в личной папке.

12. Созданный файл-отчет сохраните в личной папке и покажите преподавателю.

### **Результаты лабораторной работы.**

1. Подготовка отчета.
2. Знание основных понятий:
  - WWW-сервис.
  - Принципы поиска информации в Интернете.
  - Типы поисковых систем в Интернете.
  - Принцип работы поисковой машины.
  - Правила составления запроса из ключевых слов.
  - Дополнительные возможности расширенного поиска.
  - Возможности поиска по каталогам.
  - Поиск людей в Интернет.
  - Сохранение найденной информации.
  - Настройка браузера.

## Лабораторная работа №6 Работа в режимах FTP .

**Тема работы:** Работа в режимах FTP .

**Цель работы:** Изучить работу FTP – протокола (протокола передачи файлов).

**Теоретическая часть:**

Возможность получать файлы из FTP-архива – один из первых сервисов Интернета, который появился значительно раньше Всемирной паутины. Аббревиатура FTP означает File Transfer Protocol – протокол для пересылки файлов. Эта программа позволяет пользователю передавать файлы между двумя компьютерами, связанными между собой локальной (LAN) или глобальной (WAN) сетью. При этом компьютерные платформы могут быть различных типов. В этом и заключается главная особенность FTP в сети.

FTP-server - это понятие, за которым скрывается обычный компьютер. Но, так как он содержит общедоступные файлы и настроен на поддержку протокола FTP, то его называют сервером - поставщиком информации. Соединение выполняется с помощью FTP клиента.

Команды FTP: [команда[аргументы]]

**dir** [удаленная\_директория] [локальный\_файл] Выводит список файлов.

**ls** [удаленная\_директория] [локальный\_файл] Выводит список файлов в директории либо не стандартный вывод, либо, если указано имя локального файла, в этот файл.

**get** [удаленный\_файл] [локальный\_файл] Вызывает передачу копии удаленного файла на ваш компьютер. В случае, если имя локального файла не было задано, то оно совпадает с именем удаленного файла.

**mget** [удаленные\_файлы] Для получения нескольких файлов.

**cd** [удаленная\_директория] Сменить директорию.

**lcd** [локальная\_директория] Меняет рабочую директорию на локальной машине (без аргумента - переход в домашнюю директорию пользователя)

**bin** (или **binary**) Переключает в режим передачи двоичных файлов

**ascii** Переключает в режим передачи текстовых файлов (обычно по умолчанию).

**prompt** Переключает интерактивную подсказку. Часто при использовании команды 'mget' желательно предварительно набрать 'prompt', чтобы не давать многократные подтверждения.

**pwd** Выводит имя удаленной рабочей директории.

**mkdir** [имя\_директории] Создает директорию на удаленной машине

**open** хост [порт] Устанавливает соединение с заданным FTP сервером

**put** [локальный\_файл] [удаленный\_файл] Пересылает файл на удаленную систему. Если имя удаленного файла не указано, то оно совпадает с именем на локальной системе.

**quit** Синоним для 'bye'

**recv** [удаленный\_файл] [локальный\_файл] Синоним для команды 'get'

**reget** [удаленный\_файл] [локальный\_файл] "Дополучение" удаленного файла в том случае, когда часть его уже есть на локальной машине. Команда особенно полезна для получения больших файлов при возможных разрывах соединения.

**delete** [удаленный\_файл] Стирает удаленный файл

**close** Обрывает FTP сеанс с удаленным сервером и возвращает к командному интерпретатору

**bye** Заканчивает работу с FTP сервером и приводит к выходу из интерпретатора.

Существуют сайты для поиска требуемых файлов. Их можно разделить на два вида:

- Предоставляют посетителю интерфейс веб-страницы для работы с их собственным FTP-архивом. Например FTP-архив компании "Релком. ДС", расположенный по адресу [www.kiarchive.ru](http://www.kiarchive.ru) . Архив имеет тематическую струк-

туру, а также собственную систему поиска, что существенно облегчает процесс отбора требуемой информации.

- Обеспечивают поиск требуемого файла по всем известным им FTP-архивам, это так называемые машины FTP-поиска. Например русскоязычный ресурс FileSearch.ru ([www.filesearch.ru](http://www.filesearch.ru)), FTPplanet ([www.ftpplanet.com](http://www.ftpplanet.com)).

Начиная работать с архивом, целесообразно скопировать текстовый файл с именем `readme`, `index` или каким-либо аналогичным названием, находящийся в корневом каталоге. В нем вы найдете общее описание архива и сведения о содержании тематических каталогов. В каждом из каталогов, как правило, также имеется файл `index`, в котором хранится описание подкаталогов и файлов данного каталога. В больших и хорошо сопровождаемых архивах, наряду с общим указателем файлов, имеются указатели файлов, поступивших за последние сутки, неделю или месяц.

Имена тематических каталогов в файловых архивах могут, разумеется, быть произвольными, однако существует целый ряд общепринятых названий разделов, понимание которых помогает ориентироваться в архивах. Наиболее употребительные названия каталогов с пояснением их типичного содержания:

`apple` (или `mac`) - программное обеспечение (ПО) для компьютеров типа Macintosh

`doc` - всевозможная документация

`games` - компьютерные игры

`gnu` - свободно распространяемое в рамках проекта GNU программное обеспечение (проект создания системы, совместимой с ОС UNIX)

`FAQ` - сборники ответов на часто задаваемые вопросы (Frequently Asked Questions)

`humor` - юмор на компьютерные и другие темы

`internet` - ПО и документация о сети Internet

`linux` - ПО для операционной системы Linux

`misc` - материалы по различной тематике

`msdos` - ПО для операционной системы MS-DOS

`pictures` - графические файлы

`pub` - свободно распространяемые материалы (от слова `public`)

`rfc` - сетевые стандарты, руководства, справочники Internet (RFC -- Request for Comments)

`unix` - ПО для операционной системы UNIX

`usenet` - информация из телеконференций Usenet

`vms` - ПО для операционной системы VAX/VMS

`windows` - все для MS Windows и Windows NT

Особую роль играет каталог с именем `incoming`. В этот каталог, имеющийся на многих серверах, пользователь может помещать свои материалы для пополнения собрания архива. При этом правила хорошего тона требуют помещения вместе с файлами их краткого описания, а также согласования необходимости вашего отправления с администратором архива.

### **Ход работы:**

#### Работа в режиме FTP.

1. Создайте на диске D папку с собственной фамилией FAMSTUDENT.
2. Откройте Far (Пуск>Программы>Far manager) или Total Commander.
3. Выберите в меню НовоеFTP соединение, в появившемся окне введите `ftp://moonhe.jinr.ru`. В окне анонимное соединение должна стоять галочка.
4. На одной из панелей менеджера файлов появится корневой каталог сервера `ftp://moonhe.jinr.ru`.
5. Зайдите в каталог **pub** выберите любой подкаталог, и любой файл из этого подкаталога и скопируйте этот файл в папку FAMSTUDENT. Это можно сделать с помощью клавиши F5.

6. Повторите копирование любого файла с сервера **ftp://moonhe.jinr.ru**.
7. Откройте браузер Internet Explorer (Пуск>Программы>Internet Explorer).
8. В строке адреса ресурса введите **ftp://moonhe.jinr.ru**. В окне браузера появится корневой каталог сервера.
9. Скопируйте из папки **pub** файл **idog.jpg** в каталог FAMSTUDENT. Сравните копирование файлов по протоколу FTP с помощью программы Far и Internet Explorer. Где вы получаете больше информации о передаваемом файле?

#### Работа команд FTP.

1. На сайте [www.filesearch.ru](http://www.filesearch.ru) выберите любой действующий ftp-сервер (далее СЕРВЕР).
2. Откройте КОМАНДНУЮ СТРОКУ (Пуск>Программы>Командная строка).
3. Анонимные FTP серверы позволяют вам войти в них под именем пользователя 'anonymous' или 'ftp', наберите: ftp СЕРВЕР.
4. Когда появится подсказка с именем системы: СЕРВЕР, напечатайте **anonymous** или **ftp**. На появившуюся подсказку о пароле Password введите: ваш\_адрес\_электронной\_почты.
5. После этого вы входите в систему и можете выполнять в ней различные команды в пределах интерпретатора FTP. Вместо имени FTP сервера вы можете использовать его IP адрес.
6. Наберите в командной строке **DIR**. На экране появится список доступных директорий. Справа будут обозначены названия директорий.
7. Введите следующую команду: **CD PUB** (или любую другую директорию) – произошла смена директории. После успешного выполнения этой команды введите **PWD** – проверка текущей директории. Вы увидите имя текущей директории – PUB.
8. Введите **LS**. Появится список файлов текущей директории. В этой директории будет присутствовать файл README.TXT или другие файлы.
9. Измените текущую директорию на Вашей машине - выполните команду **LCD D:\FAMSTUDENT**.
10. Скопируйте файл README.TXT с помощью команды **GET README.TXT** и этот файл будет скопирован в текущую директорию, т.е. D:\FAMSTUDENT. Проверьте наличие файла в директории.
11. Закончите работу с удаленным сервером, набрав команду: **CLOSE**.
12. Выйдите из интерпретатора, используя команду: **BYE**. Просмотрите полученный файл.

#### **Задание для самостоятельной работы.**

1. Показать умение подключаться к *ftp*-серверу и работать с ним, используя три режима:
  - с помощью браузера;
  - с помощью любого файлового менеджера;
  - из командной строки с помощью программы *ftp*.
2. Используя, командную строку скопировать файл *irfanview\_wallpaper.bmp* из *ftp://moonhe.jinr.ru/pub/pict/* в свою папку на диске D. Листинг работы в интерпретаторе необходимо показать преподавателю.
3. Используя файловый менеджер, из *ftp.biysk.ru* из папок *pub/sw/docs/Internet* скопируйте в свою папку файл *HTML\_Справочник*.
4. Используя браузер, из *ftp.biysk.ru* из папок *pub/sw/docs/Internet* скопируйте в свою папку файл Инструментарий.

## Лабораторная работа №7 Телеконференции.

**Тема работы:** Настройка учетной записи телеконференции. NNTP-сервер и программа Microsoft Outlook Express.

**Цель работы:** Создание учетной записи телеконференции, умение просмотреть список телеконференций, поддерживаемых NNTP-сервером, подписаться на некоторые из них, получить выбранные статьи, прочитать их и сохранить в архиве на локальном диске, а также ответить на публикацию или отправить новую статью в ту или иную телеконференцию.

### Теоретическая часть:

**NETNEWS** (или Usenet, RFC-1036) - всемирная система обмена сообщениями или международная сеть телеконференций, использующая для этого единый формат. Usenet была разработана в 1979 году для системы UNIX. В настоящее время в сети новостей работает несколько тысяч узлов, охватывающих практически весь земной шар.

Сообщения рассортированы по темам, которые носят названия **newsgroups** (группы новостей) или телеконференции. Телеконференции проводятся на специальном сервере новостей. Материалом для телеконференций служат присылаемые пользователями новые статьи и ответы на опубликованные ранее.

Телеконференции (Netnews, Newsgroups), наряду с электронной почтой и FTP, относятся к "классическим" видам сервиса Internet, имеющим достаточно продолжительную историю.

Телеконференции являются сетевым сервисом, ориентированным на поддержку коллективных дискуссий, в которых могут принимать участие тысячи пользователей глобальных компьютерных сетей, и основная цель телеконференций - предоставление оперативной информации. Телеконференции можно сравнить с гигантскими досками объявлений, на которые пользователи помещают свои сообщения, доступные для чтения другим пользователям, или ответы и комментарии на ранее отправленные сообщения. Размещение сообщений в телеконференциях часто называют публикацией (posting), а само сообщение - статьей (article). Сообщения эти могут быть совершенно разного характера: от сообщения о только что произошедшем великом событии, до вопроса о том как правильно делать фарш для пельменей. Узлы сети, занимающиеся обслуживанием системы новостей, по получении пакета новостей рассылают его своим соседям, если те еще не получили такой новости. Получается лавинообразное широковещение, обеспечивающее быструю рассылку новостного сообщения по всей сети.

В настоящее время в Internet ведутся дискуссии более чем в 50 тысячах телеконференций. Каждая телеконференция имеет свое уникальное имя и представляет собой что-то вроде клуба по интересам. В тематике столь большого числа телеконференций удается ориентироваться благодаря принятым правилам составления названий телеконференций, которые имеют иерархическую структуру и состоят из нескольких слов, разделенных точками.

Название телеконференции начинается с имени иерархии или группы (категории верхнего уровня), объединяющей большое число телеконференций и часто имеющей несколько подразделов. Например, во всемирной системе телеконференций Usenet существуют следующие основные иерархии:

- comp - вопросы, связанные с компьютерами и программированием,
- news - вопросы организации, функционирования и развития системы телеконференций,
- rec - отдых, хобби,
- sci - наука,
- soc - социальные вопросы,



talk - обсуждение спорных вопросов, беседы (здесь же о религии, об искусстве),  
misc - темы, не входящие в перечисленные категории.

Есть еще более мелкие группы, которые также включены в список USENET, например,

alt – "альтернативные взгляды на окружающее". Дочерние группы называются присоединением через точку справа от родительского названия подгруппы, например, alt.music и т.д.

de – группы, относящиеся к Германии;

fido – группы поклонников сети ФИДО,

relcom – группы сети Relcom.

К русскоязычным телеконференциям относятся конференции, входящие в группу **relcom** и **fido7**. Группа **fido7** охватывает не только Россию, но и республики бывшего СССР, так что внутри нее можно найти группы **fido7.ru** (Россия) и **fido7.ua** (Украина) и др.

После имени группы последовательно идут имена подразделов, сужающие тематику телеконференции до конкретного круга вопросов. Например, телеконференция relcom.commerce.estate посвящена купле-продаже недвижимости, а relcom.commerce. food - продуктам питания.

Работа системы телеконференций реализуется с помощью специального программного обеспечения - серверов телеконференций (news-серверов), которые решают следующие задачи:

- предоставление пользователям информации об имеющихся телеконференциях и возможности просмотра их содержания, выбора и получения статей;
- получение от пользователей новых статей и сообщений, отправленных в ответ на ранее появившиеся публикации, и размещение их в соответствующих телеконференциях;
- обмен информацией с другими серверами телеконференций с целью получения и передачи новых публикаций, то есть ретрансляцию телеконференций между узлами.

Работа пользователей с системой телеконференций возможна как в режиме on-line, так и в пакетном режиме посредством электронной почты. В первом случае пользователь, имеющий IP-подключение или работающий в режиме терминала хост-компьютера, обращается к серверу телеконференций с помощью клиентской программы чтения телеконференций (news reader). При этом используется входящий в семейство TCP/IP протокол NNTP (NetNews Transfer Protocol), поэтому сервер телеконференций называют также NNTP-сервером, а программу чтения - NNTP-клиентом.

Существует большое число NNTP-клиентов, например Outlook Express. С помощью NNTP-клиента пользователь может в режиме on-line просмотреть список телеконференций, поддерживаемых NNTP-сервером, подписаться на некоторые из них, получить выбранные статьи, прочитать их и положить в архив на локальный диск, а также ответить на публикацию или отправить новую статью в ту или иную телеконференцию. Ответы на статью могут быть представлены в двух вариантах: в виде отзыва в Newsgroups, либо в виде электронного письма, отправленного непосредственно автору статьи. Хорошим тоном считается ситуация, когда отвечающие пишут на личный адрес электронной почты задавшего вопрос (это позволяет избежать массы идентичных ответов в группу), а человек, получивший ответы, отправляет обобщенный результат в группу новостей, чтобы все участники конференции могли с ним ознакомиться.

### Ход работы:

1. Запустить **Outlook Express**.
2. Создайте новую учетную запись новостей: **Сервис/Учетные записи/добавить/Новости**.

3. В появившемся окне следуйте указаниям мастера: Запишите свое имя; адрес электронной почты; сервер новостей NNTP (например **news.fido7.ru**, **ddt.demos.su**, **news.vsi.ru**, **news.mv.ru** и т.д.).
4. В появившемся окне на вопрос загрузить ли группы новостей из добавленной учетной записи службы новостей ответьте утвердительно. Список групп обширен, загрузка может быть продолжительной, но это обычно делается однократно.
5. По окончании загрузки откроется окно **Список группы новостей**. Выберите несколько интересных вам тематических групп. При этом используйте поле **Отобразить группы новостей, содержащие:** введите какое-то значимое для вас слово (например, music, comp, kitchen, talk и т.д.). Телеконференцию (группу) отмечайте левой кнопкой мыши, а затем нажимайте **Подписаться**. Если используете сервер **news.fido7.ru**, то подпишитесь на конференцию **fido7.su.kitchen**.
6. Для загрузки сообщений выбранной конференции, выделите нужную конференцию на панели **Папки**, затем **Сервис/Синхронизировать группу новостей**. В появившемся окне выберите **Загрузить** и параметры синхронизации, например **Все сообщения**.
7. Изучите заголовки полученных сообщений, изучите обсуждения некоторых вопросов. В телеконференциях сообщения группируются по обсуждениям некоторых тем. Свернутые обсуждения помечены значком +, а развернутые значком -.
8. Выделите понравившееся сообщение и, используя контекстное меню ответьте отправителю или пошлите ответ в группу.
9. Самостоятельно изучите работу с сообщениями – фильтрация, создание правил; блокировка отправителя; отказ от подписки на какую-нибудь конференцию.
10. Удалите ненужные или неуспешно загруженные телеконференции, а также ненужные учетные записи серверов новостей.
11. Создайте отчет (файл сохранить на D:\FAMSTUDENT\ ФАЙЛ.DOC), где будет содержаться следующая информация:
  - сервер новостей, к которому вы подключались;
  - на какие телеконференции подписывались и насколько успешно;
  - опишите несколько изученных вами обсуждений;
  - укажите количество загруженных сообщений для каждой конференции.

## Лабораторная работа №8 Электронная почта.

**Тема работы:** настройка учетной записи электронной почты. E-mail и программа Microsoft Outlook Express. Web-mail, создание собственного почтового ящика.

**Цель работы:** узнать принцип работы почтовых программ, назначение почтового сервера; научиться создавать учетную запись в Outlook Express; получить практические навыки при работе с электронными сообщениями, изучить дополнительные возможности и настройки Outlook Express, научиться создавать и использовать подписи (сигнатуры), осуществлять подписку на рассылку сообщений, уметь отправлять, получать, сортировать, фильтровать, пересылать почту, изучить назначение и порядок работы с адресной книгой.

### **Теоретическая часть:**

Электронная почта служит для пересылки писем через Интернет. Принципиально отправление документов по электронной почте имеет много общего с обычной почтой. Роль почтовых отделений играют узлы Интернет, где абонентам организуются специальные почтовые ящики.

Рассмотрим две службы электронной почты. Это классическая электронная почта E-mail и электронная почта, основанная на World Wide Web – Web-mail.

### Программа Microsoft Outlook Express.

Эта программа входит в состав Microsoft Internet Explorer и устанавливается одновременно с ним. При запуске программы начинается процесс соединения с удаленным компьютером–сервером и считывание с него почты. При отсутствии соединения программа переходит в автономный режим работы, который может использоваться для просмотра ранее полученных сообщений, подготовки новых документов и т.д.

Основные элементы интерфейса Outlook Express.

**Строка заголовка** – содержит стандартные элементы окна Windows-приложения,

**Строка меню** – содержит заголовки меню, предоставляющие доступ ко всем функциям, необходимым для работы с Outlook Express. Здесь также содержится много пунктов, характерных для всех приложений Windows.

**Панель инструментов** – предназначена для быстрого доступа к некоторым наиболее часто используемым командам Outlook Express.

**Панель Папки** – позволяет вывести на экран списки писем и их содержимое, которые хранятся в одной из 5-ти стандартных папок Outlook Express:

**Входящие** – сюда по умолчанию поступает вся новая почта и здесь хранятся все пришедшие сообщения.

**Исходящие** – эта папка предназначена для временного хранения отправляемых писем. Чтобы не соединяться каждый раз с Интернетом для отправки очередного письма, письма временно накапливаются в этой папке. Затем при нажатии на кнопку **Доставить почту** они одновременно уходят на сервер провайдера.

**Отправленные** – здесь по умолчанию хранятся копии отправленных сообщений.

**Удаленные** – здесь по умолчанию хранятся копии сообщений удаленных из других папок. Для окончательного удаления писем нужно вызвать контекстное меню, щелкнув правой кнопкой мыши по папке **Удаленные** и выбрать **Очистить папку**.

**Черновики** – здесь хранятся «недописанные» письма. Чтобы поместить письмо в эту папку надо, Создать сообщение, затем выполнить Файл/Сохранить. Чтобы впоследствии продолжить работу над письмом нужно открыть эту папку и дважды щелкнуть по письму.

**Панель Контакты** – в этом окне расположено содержимое адресной книги.

**Область просмотра Outlook Express** разделена на две части: вверху список сообщений из текущей папки, в нижней части показано содержимое отмеченного письма. Со-

держимое письма можно просматривать и в отдельном окне, если сделать двойной щелчок по строке с выбранным письмом.

*Строка состояния.*

Так как сообщения поступают и отправляются через сервер, то программе надо указать информацию об используемом сервере. Эта информация хранится в виде **учетной записи**.

Для создания учетной записи нужно выполнить: **Сервис/Учетные записи**. В окне **Учетные записи** выбрать подменю **Почта** и **Добавить**. В окне **Мастер Подключения** заполнить поля, согласно данным, полученным от провайдера.

**Значки с изображениями конверта** в области просмотра следует понимать так:

- открытый конверт – письмо уже прочитано,
- закрытый конверт и жирный шрифт – непрочитанное письмо.

**Скрепка у письма** говорит о том, что в письмо вложен отдельный файл (например, фото или текст, в формате Word). Двойной щелчок по имени вложенного файла позволит просмотреть его содержимое с помощью соответствующей программы.

### Дополнительные функции программы.

#### *Подписи*

Функция подписи предоставляет возможность автоматически добавлять файл или текстовую метку каждому исходящему сообщению. Подпись можно создавать, используя Outlook Express, а можно заранее подготовить и вставлять из файла.

Создание и изменение подписей выполняется, при использовании: **Сервис\Подписи**.

#### *Адресные книги*

С помощью адресных книг можно хранить как индивидуальные адреса Интернет, так и групповые, т.е. “списки групп”. Работа с адресной книгой возможна при использовании:

**Сервис\Адресная книга** или **Ctrl+Shift+B** или при использовании контекстного меню.

#### *Папки*

Можно создать дополнительные папки и настроить Outlook Express так, чтобы при поступлении писем вся почта автоматически разбиралась и складывалась по отдельным папкам.

На панели папок щелкнуть правой кнопкой мыши по **Локальные папки** и выбрать **Создать папку**, задать ее имя.

#### *Фильтрация и переадресация*

Благодаря средствам фильтрации или переадресации можно задавать правила, по которым программа должна обрабатывать входящие сообщения. Создать правила для сортировки сообщений можно используя:

**Сервис\Создать правило для почты**, затем заполнить все поля, указанные в данном окне.

#### *Блокировка*

Благодаря, блокировке можно не получать писем от нежелательных отправителей. Для этого нужно выделить письмо, а затем выбрать в меню **Сообщение\Блокировать отправителя**.

Чтобы просмотреть список заблокированных отправителей, выберите в меню:

**Сервис/Правила для сообщений/Список блокируемых отправителей**, в этом же окне можно добавлять, удалять и изменять отправителей из данного списка.

За долгие годы использования электронной почтой выработался этикет общения через e-mail. В частности, для выражения эмоций в письмах используют схематические изображения человеческого лица – смайлики:

:-) улыбка	;-) хитрец
:~))) хохот	:-* съел горькое
:-Q курит	:-& поклялся молчать
:-@ кричит	O-) аквалангист
:-( грустит	=8-) носит очки

### Web-mail

В качестве серверов почты Web-mail выступают обычные Web-серверы. Они работают в паре с базой данных и каждому клиенту при его подключении формируют Web-страницу, соответствующую текущему состоянию его учетной записи в базе данных. В качестве клиентской программы выступает обычный браузер. Web-mail в отличие от E-mail, не является самостоятельной службой – это дополнительный сервис WWW. Адреса Web-mail записываются точно так же, как и адреса E-mail.

Существует множество Web-серверов, предоставляющих систему бесплатной электронной почты. Рассмотрим некоторые из них (таблица 2).

Таблица 2.

Служба	Адрес службы	Образец адреса
<b>Российские службы</b>		
«Яндекс Почта»	narod.yandex.ru	Ваше_имя@yandex.ru
«Апорт»	Webmail.aport.ru	Ваше_имя@aport.ru Ваше_имя@aport2000.ru
«Атрус»	www.atrus.ru	Ваше_имя@atrus.ru Ваше_имя@au.ru
«Омен»	www.omen.ru	Ваше_имя@omen.ru
«Рамблер»	www.rambler.ru	Ваше_имя@rambler.ru
«Mail.Ru»	www.mail.port.ru	Ваше_имя@mail.ru
<b>Зарубежные службы</b>		
Hotmail	www.hotmail.ru	Ваше_имя@hotmail.com
USA.NET	www.usa.net	Ваше_имя@usa.net

Чтобы получить бесплатный почтовый ящик на таком сервере, необходимо зарегистрироваться.

### Ход работы:

1. Запустить **Outlook Express** и изучите интерфейс программы.
2. Создайте новую учетную запись (почтовый электронный адрес и пароль нужно взять у преподавателя).
3. Осуществите настройку программы Outlook Express таким образом, чтобы происходила проверка поступившей почты каждые 5 минут.
4. Добавить в адресную книгу адреса трех пользователей группы.
5. Отправить по одному сообщению каждому пользователю из адресной книги, прикрепив к тексту сообщения рисунок (например, сохраненный ранее с помощью браузера в личной папке). Поле **Тема** должно содержать номер компьютера пользователя.
6. Получив сообщение, перенаправьте его двум другим пользователям, включенным в адресную книгу.

7. Из полученных сообщений извлеките вложенные файлы и сохраните их в личной папке (D:\ФИО\ПОЧТА\файлы). После этого нужно ответить адресатам.
8. Создайте свою подпись по шаблону. (Например - *С уважением Фамилия, Имя, Отчество*). Отправьте кому-нибудь письмо с подписью, а в поле **тема** напишите: "Привет". **Копию** сообщения переслать адресату, отправившему Вам письмо первоначально.
9. Создайте папку **ПРИВЕТ**. Создайте правило для автоматического перемещения в папку ПРИВЕТ всех писем, содержащих в качестве темы сообщения слово "Привет". Проверьте, как работает данное правило.
10. Заблокируйте получение почты от некоторых пользователей. Просмотрите список блокируемых отправителей. Проверьте, как работает блокировка.
11. Покажите результат работы преподавателю.
12. Удалите ненужную корреспонденцию, в том числе и из папки **Корзина**.
13. Завершите работу с **Outlook Express**.
14. Создайте собственный почтовый ящик, используя браузер и данные таблицы 1.
15. Зайдите на сайт [www.citycat.ru](http://www.citycat.ru) и подпишитесь на какой-нибудь список рассылки, указав свой web-адрес. Скопируйте часть, полученных писем из рассылки и сохраните в своей папке в виде файла с именем **Рассылка**. Покажите преподавателю.

Ниже приведены некоторые сайты, где можно подписаться на какую-нибудь рассылку:

[www.list.ru](http://www.list.ru) Последние новости на любую тематику.

[subscribe.ru](http://subscribe.ru) Информационный канал, более 35000 бесплатных рассылок.

[www.tomcat.ru](http://www.tomcat.ru) Новости.

[infoart.ru](http://infoart.ru) Разнообразная информация.

[www.citycat.ru](http://www.citycat.ru) Множество списков рассылки на разные темы

[www.russ.ru](http://www.russ.ru) Русский Журнал - сетевое издание о культуре, политике, обществе.

[www.zdnet.ru](http://www.zdnet.ru) Это анонсы новостей компьютерного бизнеса и технологий, опубликованных в день рассылки.

[rsoft.ru](http://rsoft.ru) Появилась возможность оперативно получать курсы валют и всю необходимую информацию по электронной почте.

[dailynews.spb.ru](http://dailynews.spb.ru) Свежие деловые новости.

[anekdot.spb.ru](http://anekdot.spb.ru) Анекдоты почтой.

[www.cnn.com](http://www.cnn.com) Новости со всего мира от известной телевизионной компании.

## Лабораторная работа №9 RSS-технологии.

**Тема работы:** Знакомство с RSS-технологиями, с целью получения информации, касающейся вопросов информатизации общества и информационных технологий.

**Цель работы:** получение RSS-лент с использованием онлайн-агрегаторов и программы-агрегатора GreatNews, подписка на RSS-ленты по информатизации общества и информационным технологиям.

### **Теоретическая информация:**

**RSS** - это семейство форматов XML, предназначенных для описания лент новостей, анонсов статей, изменений в блогах и т.п. Но публиковать можно не только новости. Практически любой материал, который можно разделить на отдельные части, можно публиковать с помощью RSS: например, объявления о последних публикациях на сайте, новых поступлениях в файловые архивы, историю изменений программного продукта и т.п. После того, как информация преобразована в формат RSS, программа, понимающая этот формат, может вытягивать сведения о внесенных изменениях и в зависимости от результата, например, автоматически предпринимать какие-либо действия. RSS ленту иногда называют фид –feed.

Программы, умеющие работать с RSS, называются **агрегаторами**, и они очень популярны среди людей, ведущих сетевые дневники (blog-и). Некоторые программы-дневники даже позволяют другим делать комментарии к записям. И многие дневники умеют публиковать записи в формате RSS. Программа-агрегатор позволяет вам собирать все эти публикации вместе, и вы получаете возможность одновременно следить за появлением новых новостей на всех сайтах сразу и читать их краткое содержание, не посещая каждый сайт в отдельности.

### **Как работает RSS?**

На сайте, который имеет RSS-ленту, есть ссылка или графическая пиктограмма на RSS-ленту. Эта ссылка добавляется в программу агрегатор или онлайн-сервис агрегации. Программа автоматически получает информацию по этой ссылке, обрабатывает ее и показывает в удобочитаемом виде.

### **Итак, как же можно просматривать RSS-ленты?**

Во-первых, в нормальные браузеры (Mozilla firefox, Opera) уже вставлена поддержка этой технологии. В *Internet Explorer7* тоже появилась такая возможность.

Во-вторых, существует множество онлайн-агрегаторов, т.е. сайтов, которые собирают RSS-новости, на которые Вы подписались, на одной страничке. Это например *Яндекс.Лента* <http://lenta.yandex.ru> ; [www.kanban.ru](http://www.kanban.ru) русский интерфейс, здесь поддерживается возможность получения ленты RSS на e-mail; *bloglines* [www.bloglines.com](http://www.bloglines.com) – англоязычный сервис; <http://my.yahoo.com>; <http://my.msn.com> ; <http://www.google.com/reader> .

В-третьих, существует множество удобных программ - агрегаторов. Внешне обычно они похожи на программы электронной почты. Примеры бесплатных программ: *wTicker* [www.wticker.org](http://www.wticker.org); *GreatNews* – разработана китайским программистом в 2005 г. [www.curiostudio.com](http://www.curiostudio.com).

### **Каталоги RSS:**

<http://kanban.ru/catalog.asp> ;

[www.rssroot.com/ru/](http://www.rssroot.com/ru/) ;

[www.retranslator.ru/directiry/rss/](http://www.retranslator.ru/directiry/rss/) старейший (основан в 2003 г.) каталог – более 1000 лент;

[www.2rss.com](http://www.2rss.com) –более 10000 англоязычных лент.

### **Ход работы:**

В каталоге RSS, например <http://kanban.ru> подберите 2-3 ленты, в которых рассматривается тема информатизации общества и информационных технологий. Сохраните адреса этих лент.

В программе-агрегаторе *GreatNews* подпишитесь на эти рассылки лент, посмотрите результат.

В браузере, в онлайн-агрегаторе тоже подпишитесь на эти ленты.

В онлайн-агрегаторе осуществите поиск *RSS*-лент по ключевым словам: информационные технологии, информационная культура, информатизация общества и т.д.

Попытайтесь на каком-нибудь сайте, посвященном информационным технологиям найти ссылку на *RSS*, проверьте ее работоспособность.

В почтовом клиенте подпишитесь на *RSS* рассылку и сохраните результат.

Составьте письменный отчет о проделанной работе, используя кнопку *Prt Scr*, сохраните внешний вид экранов всех Ваших подписок.



## Лабораторная работа №10 Создание HTML-документа.

**Тема работы:** Создание HTML-документа.

**Цель работы:** Научиться создавать файлы, содержащие элементы форматирования, предписанные языком HTML.

**Теоретическая часть:**

HTML-документ — это просто текстовый файл с расширением \*.htm или \*.html.

Элемент разметки или оформления, входящий в формат HTML, называется *тэг*. Любой тэг имеет общий вид <ИМЯ>область действия тэга</ИМЯ> и действует на все, что расположено между треугольными скобками. Почти все тэги (кроме <P>, <BR>, <IMG>) должны закрываться, причем закрытие тэга отличается от открытия только наличием символа '/'. Тэги могут вкладываться друг в друга иерархически, но без пересечений, то есть допустимо вложение вида <ТЭГ1><ТЭГ2></ТЭГ2></ТЭГ1>, но не <ТЭГ1><ТЭГ2></ТЭГ1></ТЭГ2>. Действие вложенных тэгов объединяется, то есть, если внутри тэга, создающего **жирное начертание шрифта, вложен тэг курсива, в результате получится жирный курсив**. Внутри открывающей части тэг может содержать *опции*, уточняющие его действие и имеющие вид имя="значение".

HTML довольно "демократичен", неправильный тэг или неправильное вложение тэгов обычно не приводят к "зависаниям" браузера или каким-либо сообщениям об ошибках, хотя, разумеется, могут вызвать неправильное форматирование документа. Названия тэгов нечувствительны к регистру символов.

HTML-файл имеет следующую общую структуру:

```
<HTML>
<HEAD>
<Мета-тэги>
<Функции скриптов>
<TITLE>Заголовок документа</TITLE>
</HEAD>
<BODY>Основная часть документа</BODY>
</HTML>
```

Обязательным является только открывающий тэг <HTML>.

В тэге <HEAD> помещается заголовочная информация.

Общие сведения о документе размещаются во вложенных тэгах <МЕТА>. Перечислим основные из них:

```
<meta name="Generator" content="Имя программы, сгенерировавшей документ">
```

```
<meta name="Description" content="Краткое описание содержания страницы">
```

```
<meta name="Keywords" content="ключевые слова документа, разделенные запятыми">
```

```
<meta name="owner" content="электронный почтовый адрес владельца страницы">
```

```
<meta name="category" content="категория страницы, например, home site">
```

Мета-тэги Description и Keywords используются поисковыми машинами и настоятельно рекомендуются к использованию на любой web-странице, предназначенной для размещения в Интернет.

Существуют и другие мета-таги, используемые, например, для автоматической переадресации документа, если web-страница "переехала" на другой сервер:

```
<meta http-equiv="Refresh" content="10" URL="новый адрес страницы">
```

Браузеры поймут эту запись как инструкцию ожидать 10 секунд, а затем загрузить новый документ.

Мета-тэг кодировки документа:

**<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">**

В настоящее время широко распространены сразу несколько способов кодирования национальных алфавитов. Обычно используется одна из двух основных русскоязычных кодировок:

- windows-1251 – кодировка Windows от Microsoft, по умолчанию используемая для русского языка во всех приложениях под Windows;
- koi8-r – кодировка KOI8-R, используемая для русского языка в операционной системе Unix.

Для современных браузеров, поддерживающих несколько кодировок, указание тэга кодировки необязательно.

В тэге <TITLE>, также вложенном в тэг <HEAD> помещается текст, отображаемый в заголовке окна браузера. Основной текст документа размещен внутри тэга <BODY>. Тэг <BODY> имеет множество опций, с основными из которых мы познакомимся позже.

### **Примерный ход работы:**

#### Создание простейших файлов HTML.

1. Создайте папку, в которой Вы будете сохранять созданные Web-страницы.
2. Запустите стандартную программу Блокнот (Notepad) и наберите следующий текст с элементами форматирования:

```
<HTML>
<HEAD>
<TITLE> Учебный файл HTML </TITLE>
</HEAD>
<BODY>
Мировые информационные ресурсы
</BODY>
</HTML>
```

3. Сохраните файл под именем *format.html*.
4. Для просмотра созданной Web-страницы загрузите браузер Microsoft Internet Explorer.
5. Откройте в меню браузера Файл (File), Открыть (Open), Обзор (Browse) и найдите созданный ранее файл *format.html*, загрузите его. Убедитесь, что название Web-страницы (*Учебный файл HTML*) отразилось в верхней статусной строке браузера.

#### Управление расположением текста на экране.

1. Внесите изменения в текст, расположив слова "Мировые", "информационные", "ресурсы" на разных строках:

```
<HTML>
<HEAD>
<TITLE> Учебный файл HTML </TITLE>
</HEAD>
<BODY>
Мировые
информационные
ресурсы
</BODY>
</HTML>
```

2. Сохраните внесенные изменения, с помощью команд Файл (File), Сохранить (Save).
3. Просмотрите с помощью браузера Microsoft Internet Explorer новую полученную Web-страницу используя клавишу F5 или с помощью кнопки Обновить (Refresh).

Предыдущие шаги не вызвали никаких видимых изменений в документе html — при отображении гипертекста браузеры автоматически размещают текст на экране, не принимая во внимание встречающиеся в файле переводы строк и идущих подряд пробелов. Существуют специальные команды, выполняющие перевод строки и задающие начало нового абзаца:

- тэг перевода строки <BR> отделяет строку от последующего текста или графики;
- тэг абзаца <P> тоже отделяет строку, но еще добавляет пустую строку, которая зрительно выделяет абзац. Оба тэга являются одноэлементными (не требуют закрывающего тэга), хотя если Вы хотите, чтобы абзацы имели разное форматирование, следует в конце абзаца поставить </p>.

Если Вы хотите, напротив, запретить переносы в какой-то части текста, следует заключить ее в тэг <NOBR>...</NOBR>.

1. Внесите изменения в текст файла HTML:

```
<HTML>
<HEAD>
<TITLE> Учебный файл HTML </TITLE>
</HEAD>
<BODY>
Мировые <P>информационные <BR>ресурсы
</BODY>
</HTML>
```

2. Сохраните внесенные изменения в файле format.html. Просмотрите с помощью браузера обновленную страницу.

Кроме того, существует команда, которая запрещает программе браузера каким-либо образом изменять форматирование текста и позволяет точно воспроизвести на экране заданный фрагмент текстового файла:

```
<PRE> Мировые
                информационные
                                ресурсы </PRE>
```

В окне браузера эта фраза будет выглядеть:

```
Мировые
                информационные
                                ресурсы
```

PRE – означает "предварительно отформатированный" текст, т.е. текст в котором для выравнивания использованы пробелы, знаки табуляции.

Некоторые символы являются для HTML служебными и при "ручном" наборе HTML-документа вместо них следует использовать сочетания символов из таблицы:

Символ	Обозначение в HTML
<	&lt;
>	&gt;

&	&amp;
"	&quot;

Если нужно вставить в текст один или несколько неизменяемых пробелов, для этого используется сочетание символов &nbsp;

### Форматирование фрагментов текста.

Для форматирования текста абзацев используйте тэги, приведенные в таблице :

Тэг	Пояснение	Образец
<H1>...</H1>	Заголовок 1 уровня	<b>1 Заголовок1</b>
<H2>...</H2>	Заголовок 2 уровня	<b>1.1 Заголовок2</b>
<H3>...</H3>	Заголовок 3 уровня	<b>1.1.1 Заголовок3</b>
<H4>...</H4>	Заголовок 4 уровня	<b>1.1.1.1 Заголовок4</b>
<H5>...</H5>	Заголовок 5 уровня	<b>1.1.1.1.1 Заголовок5</b>
<H6>...</H6>	Заголовок 6 уровня	<b>1.1.1.1.1.1 Заголовок6</b>
<BIG>...</BIG>	Большой	Большой
<SMALL>...</SMALL>	Маленький	Маленький
<SUP>...</SUP>	Верхний индекс	Верхний индекс
<SUB>...</SUB>	Нижний индекс	Нижний индекс
<B>...</B>	Жирный	<b>Жирный</b>
<I>...</I>	Курсив	<i>Курсив</i>
<U>...</U>	Подчеркнутый	<u>Подчеркнутый</u>
<S>...</S>	Перечеркнутый	<del>Перечеркнутый</del>

1. Внесите изменения в файл *format.html*:

```
<HTML>
<HEAD>
<TITLE> Учебный файл HTML </TITLE>
```

```
</HEAD>
<BODY>
<B>Мировые</B> <I>информационные</I> <U> ресурсы</U>
</BODY>
</HTML>
```

2. Посмотрите новую полученную Web-страницу.

3. Попробуйте использовать вложение тэгов:

```
<I><B>Мировые</B></I> <I>информационные</I> <U> ресурсы</U>
```

### Изменение размера текста

Существует два способа управления размером текста, отображаемого браузером:

- использование стилей заголовка;
- задание размера текущего шрифта;

1. Внесите изменения в файл *format.html*:

```
<HTML>
<HEAD>
<TITLE> Учебный файл HTML </TITLE>
</HEAD>
<BODY>
<H1>Мировые</H1> <I>информационные</I> <U>ресурсы</U>
</BODY>
</HTML>
```

2. Просмотрите обновления через браузер.

3. Внесите следующие изменения в файл *format.html*:

```
<HTML>
<HEAD>
<TITLE> Учебный файл HTML </TITLE>
</HEAD>
<BODY>
<FONT SIZE="7">Мировые</FONT> информационные ресурсы
</BODY>
</HTML>
```

4. Самостоятельно измените размер шрифта для текста "информационные ресурсы", используя тэг <FONT>.

### Гарнитура и цвет шрифта

SIZE в тэге <BASEFONT> задает базовый (основной) размер шрифта Web-документа. Величина атрибута может лежать в пределах от 1 до 7. По умолчанию используется величина 3. Например:

```
<BASEFONT SIZE = 4>
```

```
<P><I>Мировые информационные ресурсы</I>
```

Тэг **FONT** позволяет задавать размер текущего шрифта в отдельных места текста в диапазоне от 1 до 7, для этого также используется атрибут **SIZE**. Шрифты могут быть заданы относительно базового: size=**+**число size=**-**число.

**Замечание:** сумма базового размера шрифта и размера текущего шрифта должна быть не меньше 1 и не более 7. Например, для основного шрифта, равного 3, размер текущего шрифта может находиться в пределах от -2 до +4.

Например:

```
<BASEFONT SIZE = 4>
<P><Font size=+2><I> Мировые информационные ресурсы </I></Font>
<P>Вопросы к зачету:<BR>
```

Тэг **BIG** выводит текст шрифтом на один размер больше текущего. Тэг **SMALL** выводит текст шрифтом на один размер меньше текущего.

Тэг <FONT> предоставляет возможности управления размером, начертанием и цветом текста. Изменение гарнитуры шрифта выполняется простым добавлением к тэгу <FONT> атрибута FACE. Например, для изображения текста шрифтом Arial необходимо записать <FONT FACE="ARIAL">.

Для изменения цвета шрифта в тэге <FONT> можно использовать атрибут COLOR="X". Вместо X надо подставить либо название цвета, либо его шестнадцатеричное значение (в кавычках и со знаком #). При задании цвета шестнадцатеричным числом можно представить этот цвет разложенным на три составляющие: красную(R), зеленую (G), синюю (B), каждая из которых имеет значение от 0 до FF. Примеры записи цвета в формате RGB приведены в таблице:

Цвет	RRGGBB
black   черный	000000
white   белый	FFFFFF
red   красный	FF0000
green   зеленый	00FF00
blue   синий	0000FF
gray   серый	A0A0A0
purple   фиолетовый	FF00FF
yellow   желтый	FFFF00
brown   коричневый	996633
orange   оранжевый	FF8000
violet   лиловый	8000FF

Полная таблица цветов, названий цветов и их эквиваленты RGB представлены в файле *Таблица HTML кодов цветов.doc*.

1. Внесите изменения в файл *format.html*:

```
<HTML>
<HEAD>
<TITLE> Учебный файл HTML </TITLE>
</HEAD>
<BODY>
<U><I><B><FONT COLOR="#FF0000" FACE="ARIAL"
SIZE="7">Мировые</FONT></B></I></U>информационные ресурсы
</BODY>
</HTML>
```

2. Самостоятельно измените размер, цвет, гарнитуру текста документа и просмотрите обновления.

#### Выравнивание текста по горизонтали

С помощью тэгов HTML можно управлять горизонтальным выравниванием текста. Если не оговаривать способ выравнивания, все элементы в документе будут выравниваться по левому краю и иметь неровное правое поле. Современные браузеры для выравнивания текста используется атрибут `ALIGN=`, который встраивается в тэги абзаца или заголовка.

- `ALIGN=CENTER` — Выравнивание по центру
- `ALIGN=RIGHT` — Выравнивание по правому краю
- `ALIGN=LEFT` — Выравнивание по левому краю

1. Внесите изменения в файл *format.html*:

```
<HTML>
<HEAD>
<TITLE> Учебный файл HTML </TITLE>
</HEAD>
<BODY>
<P ALIGN=CENTER>
<FONT COLOR="#008080" SIZE="7"><B>Мировые</B></FONT><BR>
<FONT SIZE="6"><I>информационные ресурсы</I></FONT>
</P>
</BODY>
</HTML>
```

2. Просмотрите обновления

#### Задание цвета фона и текста

Цвета фона и текста документа устанавливаются в начале файла HTML в тэге `<BODY>`. Атрибут `BGCOLOR=` задает цвет фона страницы, `TEXT=` определяет цвет текста для всей страницы, `LINK=` и `VLINK=` определяют цвета соответственно непросмотренных и просмотренных ссылок.

1. Внесите изменения в файл *format.html*:

```
<HTML>
<HEAD>
<TITLE> Учебный файл HTML </TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFCC" TEXT="#330066">
<P ALIGN=CENTER>
<FONT COLOR="#008080" SIZE="7"><B>Мировые</B></FONT><BR>
<FONT SIZE="6"><I>информационные ресурсы</I></FONT>
</P>
</BODY>
</HTML>
```

2. Просмотрите обновления

Для создания в документе текстового **раздела** используется тэг `<div>...</div>`, с опцией `align`, указывающей способ выравнивания текста в разделе (так же, как в тэге `<p>`). Указанный в тэге `<div>` способ выравнивания текста используется по умолчанию во всех абзацах этого раздела, если в абзаце в явном виде не указан другой способ выравнивания. Для размещения части документа по центру окна браузера можно также использовать тэг `<center>...</center>`

### Горизонтальная черта

Отделить часть текста горизонтальной чертой можно с помощью тэга `<HR>`. Опция `ALIGN` со значением `LEFT`, `RIGHT` или `CENTER` определяет выравнивание черты на странице, опция `SIZE="размер"` - толщину линии в пикселах, а опция `WIDTH="ширина"` - ширину, указанную в пикселах (например, 600) или процентах (100%). Если указана опция `NOSHADER`, линия создается без трехмерных эффектов. Пример:

Эта черта создана тэгом вида `<HR ALIGN="CENTER" SIZE="1" WIDTH="50%">`

### Вставка цитат

Тэг **BLOCKQUOTE** предназначен для обозначения в документе цитаты из другого источника. Текст, обозначенный этим тэгом, отступает от левого края документа на 8 пробелов.

*Пример:*

```
<BLOCKQUOTE>"Я не снимаю фильмы для того, чтобы делать деньги, я делаю деньги, чтобы снимать больше хороших фильмов"</BLOCKQUOTE>
```

### **Задание для самостоятельного выполнения.**

С помощью Блокнота Windows создайте и сохраните в личной папке HTML-файл с именем `first.html` и заголовком, совпадающим с Вашей фамилией. В тексте файла создайте 2 заголовка первого уровня "Мои анкетные данные" и "Мое хобби". Заголовки и тема Вашего файла может быть любой.

В разделах "Мои анкетные данные" и "Мое хобби" добавьте по 2-3 абзаца осмысленного текста, выровненных по ширине окна. Внутри текста используйте выделение слов курсивом, жирным и подчеркиванием.

Добавьте в раздел "Мое хобби" заголовок второго уровня "Программирование" и поместите под ним небольшой листинг программы на изучаемом Вами языке программиро-



вания. Позаботьтесь о выводе листинга программы с сохранением всех отступов, интервалов и переносов строк.

Используйте какой-нибудь фон на странице, добавьте горизонтальную черту, цитату. Добавьте к документу подпись, отформатированную как в примере:

*Студент группы 110 Иванов А.П.*

Добавьте в заголовок документа мета-тэги Description и Keywords.

Просмотрите полученный файл с помощью браузера Internet Explorer. Обратите внимание на изменение форматирования документа при изменениях размера окна и размера шрифтов браузера.

Результат Вашей работы покажите преподавателю.

## Лабораторная работа №11 Размещение графики на Web-странице, списки.

**Тема работы:** Размещение графики на Web-странице, списки.

**Цель работы:** научиться формировать списки на страницах, научиться внедрять в html-документ графические изображения.

**Теоретическая часть:**

### СПИСКИ

Часто для оформления текста используют списки. Списки бывают нумерованные и ненумерованные (маркированные):

**Пронумерованные списки** - используется тэг `<OL>.....</OL>`.

Тэгом **LI** определяется элемент списка.

В пронумерованном списке нумерация происходит автоматически.

Атрибуты тэга `<OL TYPE= вид счетчика>`

Вид счетчика:

- А - большие латинские буквы,
- а - маленькие латинские буквы,
- I - большие римские цифры,
- i - маленькие римские цифры,
- 1 - обычные цифры.

Опция **START="число"** нумерованного списка позволяет начать нумерацию с цифры, отличной от 1 или буквы, отличной от А.

Для нумерованного списка можно изменить нумерацию, указав в тэге `<LI>` опцию **VALUE="число"**.

### **Маркированные списки**

Маркированный список пишется в тэгах `<UL>...</UL>`.

Перед каждым элементом списка ставим тэг **LI**.

Для пометки элемента списка используются маркеры. Для маркированного списка опция **TYPE** указывает вид маркера – circle (кружок), disc (стандартный маркер – маленький закрашенный кружок) или square (квадратик). При введении этих значений обязательно используйте строчные буквы.

Опция **СОМПАКТ** делает список более компактным. Для создания **комбинированных списков** тэги нумерованных и маркированных списков можно вкладывать друг в друга. Слева показан пример списка, а справа его HTML-код:

1. Компьютеры	<code>&lt;ol compact&gt;</code>
• Pentium Pro	<code>&lt;li&gt;Компьютеры</code>
• Pentium MMX	<code>&lt;ul compact</code>
	<code>type=disc&gt;</code>
2. Принтеры	<code>&lt;li&gt;Pentium</code>
▪ матричный	<code>Pro&lt;/li&gt;</code>
▪ струйный	<code>&lt;li&gt;Pentium</code>
▪ лазерный	<code>MMX&lt;/li&gt;</code>
	<code>&lt;/ul&gt;</code>
	<code>&lt;/li&gt;</code>
	<code>&lt;li&gt;Принтеры</code>
	<code>&lt;ul compact</code>
	<code>type=square&gt;</code>

```
<li>матричный</li>
<li>струйный</li>
<li>лазерный</li>
</ul>
</li>
</ol>
```

**Список определений**, содержащий чередующиеся пары вида "термин" - "его описание". Список определений начинается с тэга **<DL>**, термин выделяется тэгом **<DT>**, а описание, которое обычно выводится со сдвигом вправо, тэгом **<DD>**.

Пример:

```
<p align=center><font size="5">Стишок для детей</font>
<dl>
<dt><b>Красный свет - </b></dt><dd>идти опасно</dd>
<dt><b>Желтый свет - </b></dt><dd>погоди</dd>
<dt><b>Зеленый свет - </b></dt><dd>спокойно улицу переходи!</dd>
</dl>
```

## ГРАФИКА

Одна из наиболее привлекательных черт Web — возможность включения ссылок на графические и иные типы данных в HTML-документ. Делается это при помощи тэга **<IMG>**. Использование этого тэга позволяет значительно улучшить внешний вид и функциональность документов.

Один из способов использования графики в HTML-документах— это внедрение графических образов в документ, что позволяет пользователю видеть изображения непосредственно в контексте других элементов документа. Это наиболее используемая техника при проектировании документов, называемая иногда "inline image". Синтаксис тэга:

```
<IMG SRC="URL" ALT="text" HEIGHT=n1 WIDTH=n2
ALIGN=|left|center|right|top|middle|bottom|texttop BORDER=n3 VSPACE=n4
HSPACE=n5>
```

Элементы синтаксиса тэга:

**URL** - обязательный параметр, имеющий такой же синтаксис, как и стандартный URL. Данный URL указывает браузеру где находится рисунок. Рисунок должен храниться в графическом формате, поддерживаемом браузером. На сегодняшний день форматы GIF и JPG поддерживаются большинством браузеров.

**ALT**="text"

Этот необязательный элемент задает текст, который будет отображен браузером, не поддерживающим отображение графики или с отключенной подкачкой изображений. Обычно, это короткое описание изображения, которое пользователь мог бы или сможет увидеть на экране. Если данный параметр отсутствует, то на месте рисунка большинство браузеров выводит пиктограмму (иконку), активизировав которую, пользователь может увидеть изображение.

**HEIGHT**=*n1*

Этот необязательный параметр используется для указания высоты рисунка в пикселях. Если данный параметр не указан, то используется оригинальная высота рисунка. Ширину и высоту изображения можно задать в процентах соответствующих величин окна браузера, например **HEIGHT**=20%.

**WIDTH**=*n2*

Параметр также необязателен, как и предыдущий. Позволяет задать абсолютную ширину рисунка в пикселях.

**ALIGN** - картинку можно расположить слева (LEFT), по центру (CENTER), справа (RIGHT), также можно выровнять картинку: по верхнему краю (TOP), посередине (MIDDLE), по нижнему краю (BOTTOM) текста. Этот параметр используется, чтобы сообщить браузеру, куда поместить следующий блок текста. Это позволяет более строго задать расположение элементов на экране. Если данный параметр не используется, то большинство браузеров располагает изображение в левой части экрана, а текст - справа от него.

**BORDER** - этот параметр позволяет автору определить ширину рамки вокруг рисунка.

**VSPACE** - позволяет установить размер в пикселях пустого пространства над и под рисунком, чтобы текст не наезжал на рисунок. Особенно это важно для динамически формируемых изображений, когда нельзя заранее увидеть документ.

**HSPACE** - то же самое, что и VSPACE, но только по горизонтали.

### Фоновые рисунки

Большинство браузеров позволяет включать в документ фоновый рисунок, который будет отображаться на фоне всего документа. Описание фонового рисунка включается в тэг BODY и выглядит следующим образом:

```
<BODY BACKGROUND="picture.gif">
```

Если Вы хотите, чтоб фоновая картинка не прокручивалась вместе с текстом, то вставьте команду BGPROPERTIES=FIXED. Например:

```
<BODY BACKGROUND="Приложение1\15.jpg" BGPROPERTIES=FIXED>
```

### **Ход работы:**

#### Размещение графики на Web-странице.

Тэг <img> является одиночным, т.е. закрывающий тэг не применяется. Графика в Web, как правило, распространяется в трех форматах: GIF, JPG, PNG. Перед выполнением упражнения поместите файл *picture.jpg* в ту же папку, которая будет использована для хранения создаваемой Web-страницы. Для выполнения этой работы используем файлы, которые были созданы в Лаб.раб 6.

1. Внесите изменения в файл *format.html*:

```
<HTML>
<HEAD>
<TITLE> Учебный файл HTML </TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="#330066">
<P ALIGN=CENTER>
<FONT COLOR="#008080" SIZE="7"><B>Мировые</B></FONT><BR>
<FONT SIZE="6"><I>информационные ресурсы</I></FONT><BR><BR>
<IMG SRC=" picture.jpg">
</P>
</BODY>
</HTML>
```

2. Самостоятельно внесите изменения в файл *format.html*, опробовав использование таких атрибутов графики как ALT, BORDER, HEIGHT, WIDTH. Пример использования атрибутов приведен в таблице ниже:

Атрибут	Формат	Описание
ALT	<IMG SRC="picture.jpg" ALT="картина">	Надпись "картинка" выводится на экран при подведении указателя мыши к изображению.
BORDER	<IMG SRC="picture.jpg" BORDER="3">	Задаёт рамку вокруг изображения толщиной 3 пикселя.
ALIGN	<IMG SRC="picture.jpg" ALIGN=TOP">	Выравнивает изображение относительно текста по верхней границе текста.
HEIGHT	<IMG SRC="picture.jpg" HEIGHT=111">	Вертикальный размер изображения принудительно устанавливается в 111 пикселей.
WIDTH	<IMG SRC="picture.jpg" WIDTH=220">	Горизонтальный размер изображения принудительно устанавливается в 220 пикселей.
VSPACE	<IMG SRC="picture.jpg" VSPACE="8">	Атрибут добавляет верхнее и нижнее пустые поля высотой 8 пикселей.
HSPACE	<IMG SRC="picture.jpg" HSPACE="8">	Добавляет левое и правое пустые поля шириной 8 пикселей.

### Фоновое отображение графики на Web-странице

Поместите файл *fon.jpg* в ту же папку, что и *format.html*. Можно использовать фоновые рисунки из папки FON.

1. Внесите изменения в файл *format.html*:

```

<HTML>
<HEAD>
<TITLE> Учебный файл HTML </TITLE>
</HEAD>
<BODY BACKGROUND="fon.jpg" TEXT="#330066">
<P ALIGN=CENTER>
<FONT COLOR="#008080" SIZE="7"><B>Мировые</B></FONT><BR>
<FONT SIZE="6"> <I>информационные ресурсы</I></FONT><BR><BR>
</P>
</BODY>
</HTML>

```

2. Самостоятельно зафиксируйте фон страницы, чтобы он не двигался вместе с текстом.

### Буквицы

Графические элементы возможно применять в качестве различного вида "украшений". Если Вы придумаете что-нибудь оригинальное, ваша веб-страница будет смотреться необычно и запомнится посетителю.

Можно применить графический элемент в качестве буквицы. Встроить буквицу в текст можно следующим образом:

```
<IMG SRC="b.gif" WIDTH="60" HEIGHT="59" BORDER="0" ALT="B">
```



от пример встроенной в текст буквицы, она добавлена в начало параграфа. На всякий случай в качестве альтернативного текста дается буква "B", чтобы пользователю с отключенной графикой не приходилось строить догадки относительно первой буквы.

1. В слове *Мировые* замените букву М изображением, взять его можно в папке BUKVIZA.
2. Изменяя значение атрибута ALIGN, добейтесь наилучшего расположения буквы на экране.
3. Используя любой графический редактор, создайте свое, альтернативное, изображение буквы *М* и встройте его в документ.

#### Графические маркеры.

Для создания более привлекательных маркированных списков можно использовать графические маркеры. Такой способ удобен при составлении коротких списков.

Сначала надо создать графический файл, изображение которого можно будет использовать в качестве графического маркера. Можно использовать файл *heart.gif*. Внесите изменения в файл *format.html*:

```
<dl>  
<dd><IMG SRC="heart.gif">идти опасно</dd>  
<dd><IMG SRC="heart.gif">погоди</dd>  
<dd><IMG SRC="heart.gif">спокойно улицу переходи!</dd>  
</dl>
```

#### **Задание для самостоятельного выполнения.**

С помощью Блокнота Windows создайте и сохраните в личной папке HTML-файл с именем *graphic.html* и заголовком, совпадающим с темой лабораторной работы.

Можно создать страницу с подробным описанием содержания Вашего сайта, причем содержание оформите в виде комбинированных списков.

Установите фоновый рисунок, используйте буквицу для оформления страниц. Обязательно вставьте несколько графических изображений, с различными атрибутами, создайте список с графическими маркерами.

Добавьте в заголовок документа мета-тэги *Description* и *Keywords*.

Результат Вашей работы покажите преподавателю.

## Лабораторная работа №12 Создание таблиц в HTML-документе.

**Тема работы:** Создание таблиц в HTML-документе.

**Цель работы:** Научиться создавать и редактировать таблицы в терминах языка HTML.

**Теоретическая часть:**

Таблицы представляют собой особую часть HTML-документа. Данные в ней организованы в виде прямоугольной сетки, состоящей из вертикальных столбцов и горизонтальных рядов. Каждая клетка таблицы является ячейкой. Ячейки могут содержать в себе текст, графику или другую таблицу.

Таблица заполняется горизонтальными рядами ячейка за ячейкой слева направо. Заполнение начинается с левого верхнего угла и заканчивается правым нижним. Каждая ячейка должна быть заполнена. Для создания пустых ячеек используются пробелы.

### Теги оформления таблиц

Тег	Форма записи	Примечание
TABLE	<TABLE>текст</TABLE>	Объявление таблиц.
TR	<TR>текст</TR>	Объявление строки.
TD	<TD>текст</TD>	Объявление ячейки в строке.
TH	<TH>текст</TH>	Заголовок столбца. Текст, заключенный между тегами <th> </th>, автоматически выделяется полужирным шрифтом и позиционируется по центру ячейки. Применение необязательно.

### Атрибуты тега <TABLE>

Атрибут	Форма записи	Примечание
BORDER	<TABLE BORDER="X">	Задаёт рамку вокруг таблицы.
WIDTH	<TABLE WIDTH="XX%">	Задаёт ширину таблицы как XX% от ширины страницы или как XX пикселей.
BGCOLOR	<TABLE BGCOLOR="#RRGGBB">	Задаёт цвет фона таблицы.
ALIGN	<TD ALIGN=X>	Устанавливает выравнивание по горизонтали (X=Right, X=Left, X=Center)
CELLSPACING	<TABLE CELLSPACING="X">	Расстояние между ячейками в пикселях X

CELLPADDING	<TABLE CELLPADDING="X">	Размер свободного пространства между границами ячейки и ее содержимым в пикселях X
HSPACE="число"	<TABLE HSPACE="X">	Размер свободного пространства слева и справа от таблицы, в пикселях
VSPACE="число"	<TABLE VSPACE="X">	Размер свободного пространства сверху и снизу от таблицы, в пикселях
BGCOLOR	<TABLE BGCOLOR="#RRGGBB">	Задаёт цвет фона ячейки.
BORDERCOLOR	<TABLE BORDERCOLOR="#RRGGBB">	Цвет рамки таблицы
BACKGROUND	<TABLE BACKGROUND="URL рисунка">	Фоновое изображение таблицы

*Атрибуты тегов <TD> и <TR>*

Атрибут	Форма записи	Примечание
ALIGN	<TD ALIGN=X>	Устанавливает выравнивание по горизонтали (Right, Left, Center)
VALIGN	<TD VALIGN=X>	Устанавливает выравнивание по вертикали (Top – по верхнему краю, Middle – по середине, Bottom – по нижнему краю)
WIDTH	<TD WIDTH="XX% ">	Задаёт ширину ячейки (строки) как XX% от ширины страницы или как XX пикселей.
BGCOLOR	<TD BGCOLOR="#RRGGBB">	Задаёт цвет фона ячейки (строки).
BORDERCOLOR	<TD BORDERCOLOR="#RRGGBB">	Цвет рамки ячейки (строки).
BACKGROUND	<TD BACKGROUND="URL рисунка">	Фоновое изображение ячейки (строки).



Если нужно задать заголовок всей таблицы, используйте тег `<CAPTION>`:

`<CAPTION ALIGN="где" >Текст заглавия</CAPTION>`.

Он должен быть внутри тега `<TABLE>`, но вне описания ячеек. Тег имеет один параметр: `ALIGN` - указывает положение заголовка: он может быть в верхней (TOP) или нижней (BOTTOM) части таблицы.

Таблицы позволяют эффективно размечать Web-страницы. Принимая толщину рамки равной нулю `BORDER="0"`, можно сделать разметку невидимой.

Простейший пример таблицы:

	ГОРОД	СТРАНА
Иванов	Москва	Россия
Смит	Техас	США

```
<html>
<body>
<table align=center width="70%" border=2>
  <tr>
    <th>&nbsp;</th><th>ГОРОД</th><th>СТРАНА</th>
  </tr>
  <tr align=right>
    <td>Иванов</td><td>Москва</td><td>Россия</td>
  </tr>
  <tr align=right>
    <td>Смит</td><td>Техас</td><td>США</td>
  </tr>
</table>
</body>
</html>
```

Для того, чтобы **растянуть** ячейку таблицы на несколько строк или столбцов, в тегах ячейки `<TD>` используются опции `ROWSPAN="число"` и `COLSPAN="число"`, указывающие, сколько строк и сколько столбцов таблицы охватывает данная ячейка. Если ячейку следует оставить пустой, в нее обычно помещают "жесткий" символ пробела `&nbsp;`;

Примеры:

		Характеристики	
		Средний рост, см	Средний вес, кг
Пол	Мужской	177	73
	Женский	166	65

Данная таблица создана с помощью следующего HTML-кода:  
`<table border="1" cellpadding="5" cellspacing="0">`

```

<tr>
  <td colspan="2" rowspan="2">&nbsp;</td>
  <td colspan="2">Характеристики</td>
</tr>
<tr>
  <td>Средний рост, см</td>
  <td>Средний вес, кг</td>
</tr>
<tr align="center">
  <td rowspan="2">Пол</td>
  <td>Мужской</td>
  <td>177</td>
  <td>73</td>
</tr>
<tr align="center">
  <td>Женский</td>
  <td>166</td>
  <td>65</td>
</tr>
</table>

```

Следующая таблица использует 90% ширины окна браузера и различные способы выравнивания данных в ячейках:

Заголовок	
Данные, выровненные по левому краю ячейки	Данные, выровненные по правому краю ячейки
25% ширины таблицы	75% ширины таблицы

```

<table width="90%" align="center" border="5"
cellpadding="2" cellspacing="0">
  <tr valign="center">
    <td align="center" colspan="2"><big>Заголовок</big></td>
  </tr>
  <tr valign="top">
    <td align="left">Данные, выровненные по<br>
      левому краю ячейки</td>
    <td align="right">Данные, выровненные по<br>
      правому краю ячейки</td>
  </tr>
  <tr>
    <td width="25%">25% ширины таблицы</td>
    <td width="75%">75% ширины таблицы</td>
  </tr>
</table>

```

Сложные и красивые эффекты могут быть достигнуты вложением таблиц друг в друга - внутренняя таблица при этом должна быть целиком вложена в таг <td> внешней таблицы.

#### **Задание для самостоятельного выполнения.**

Создайте документ tab.html. Основная часть документа должна быть организована как таблица, состоящая из названия, заголовков, нескольких (не менее 3) столбцов и нескольких строк (не менее 3). Причем в некоторых ячейках должны быть фоновые рисунки (<TD background="url">&nbsp;</TD>), а некоторые ячейки должны сами содержать рисунок (<TD> IMG SRC="url" </TD>).

Для этой web страницы тоже нужен либо фоновый рисунок, либо фоновый цвет. Ширина всей таблицы должна быть 90% от ширины окна браузера. Ячейки таблицы должны иметь обрамление шириной несколько пикселей и определенный цвет.

## Лабораторная работа №13 Создание гиперссылок в HTML-документе.

**Тема работы:** Создание гиперссылок в HTML-документе.

**Цель работы:** Научиться формировать гиперссылки на сайты в Internet, на страницы собственного сайта и на метки в текущем документе.

**Теоретическая часть:**

Важнейшим свойством языка HTML является возможность размещения на странице ссылок на другие документы, на некоторую точку в текущем HTML-документе.

В качестве ссылки можно использовать текст или графику.

Браузер выделяет (обычно цветом и/или подчеркиванием) слова, являющиеся ссылками. Цвет выделения ссылок устанавливается в теге **BODY**:

**LINK** - устанавливает цвет выделения ссылок.

**VLINK** - устанавливает цвет выделения ссылок, на которых уже побывали.

**ALINK** - устанавливает цвет активной ссылки.

Возможны ссылки:

- на удаленный HTML-файл;
- на некоторую точку в этом же документе;
- на любой файл, не являющийся HTML-документом.

Тег **<A>...</A>** создает гиперссылку.

Атрибуты:

**HREF** - имя документа (URL-адрес), к которому осуществляется переход.

**TITLE** задает текст, который выскочит, если задержать курсор мыши на указателе

**TARGET** указывает, куда должна загружаться вызываемая страничка. Если указать **TARGET=\_blank**, то браузер загрузит вызываемую страничку в новое окно. Если этот атрибут опустить, то новая страничка загрузится в текущее окно.

### Ссылка на другой документ.

Описывается ссылка на другой документ следующим образом:

**<A HREF="адрес"> Текст, который будет служить как обращение к другому документу </A>**.

Ссылка, в которой указан полный URL адрес документа, называется **абсолютной**. Абсолютные ссылки используются для связи с внешними ресурсами Интернет, URL которых известен нам и не меняется.

Например:

**<A HREF="http://www.rambler.ru" >Поисковая система RAMBLER </A>**

**Относительная** ссылка ссылается на документ, опуская общую адресную часть. Например, если из документа index.html нужно сослаться на документ test.html, находящийся в той же папке, это можно сделать ссылкой вида

**<a href="test.html">документ test.html</a>**

При использовании относительной ссылки можно сослаться на папки, которые являются как вложенными, так и родительскими по отношению к папке, в которой расположен исходный документ. Например, ссылка на рисунок с именем my.jpg, находящийся во вложенной папке images может иметь вид:

**<a href="images/my.jpg">посмотрите рисунок</a>**

### Ссылки в пределах одного документа.

Такие ссылки называются **внутренними** и требуют наличия двух частей: метки и самой ссылки. Метка определяет точку, к которой происходит переход по ссылке. Ссылка использует имя метки. Ссылки выделяются цветом или подчеркиванием, в зависимости от того, как настроен браузер.

Ссылка должна выглядеть примерно так:

```
<A HREF="#ПН">Понедельник</A>
```

Перед именем метки *ПН*, указывающей куда производится ссылка, ставится символ #. Между символами > и < располагается текст *Понедельник*, на котором производится щелчок для перехода по ссылке.

Метка должна выглядеть примерно так:

```
<A NAME="ПН">Какой-то текст, имеющий отношение к понедельнику</A>
```

### Ссылка на E-mail.

Ссылка на почтовый ящик (*E-mail*) прописывается следующим образом:

```
<A HREF="mailto:pochta@mail.ru">Напишите мне письмо</A>
```

### Гиперссылка-картинка.

В качестве ссылки можно использовать не только текст, но и графику.

Для того, чтобы использовать в качестве гиперссылки картинку, достаточно заключить тег `IMG` в тег `A`.

Например, `<a href="File.htm"></a>`

При разработке собственного web-сайта для перехода между его страницами используются, как правило, относительные ссылки, что позволяет просматривать сайт на локальной машине, не внося в него изменений, а также при необходимости позволяет легко переместить сайт на другой сервер.

### **Задание для самостоятельного выполнения.**

Создайте документ *links.html* с заголовком "Мои ссылки" и сохраните его в той же папке, что и документ *first.html* из Л.р. № 10.

Добавьте в *links.html* перечень ссылок на различные ресурсы Интернет, например: Почтовая служба *http://www.mail.ru/*

Поисковые серверы *Yandex* и *Rambler* и т.д. - всего 6-8 ссылок.

При щелчке по ним ссылки должны открываться в **новом** окне.

Добавьте в верхнюю часть документа *links.html* ссылку вида

**на главную страницу**

возвращающую к документу *first.html*.

Добавьте в документ *first.html* строку вида "Мои ссылки", адресующую документ *links.html*. Документ *links.html* должен открываться в текущем окне.

Сделайте подпись из файла *first.html* ссылкой на Ваш адрес электронной почты.

Добавьте в нижнюю часть файла *first.html* ссылку "в начало страницы", адресующую первую строку документа, сделайте в этом файле еще несколько внутренних ссылок.

В таблице файла *tab.html* несколько ячеек, содержащих графический файл, сделайте гиперссылками, чтобы они открывались в отдельном окне. Например в таблице расположена маленькая фотография (пиктограмма, размер ее около 30 Kb), а при нажатии на нее эта фотография будет расположена в отдельном окне, размер фотографии должен быть около 80% окна. Размер увеличенного фото должен быть не более 300 Kb.

## Лабораторная работа №14 Создание фреймов в HTML-документе.

**Тема работы:** Создание фреймов в HTML-документе.

**Цель работы:** Научиться формировать фреймы для более удобной и наглядной работы с сайтом.

**Теоретическая часть:**

В HTML существует возможность разделить основное окно, в котором отображаются Web-странички, на несколько частей - фреймов. Фрейм, по сути, представляет собой окно, в которое загружается отдельная страничка. В результате - в окне браузера отображается сразу несколько страничек. Фреймы удобны при создании страниц, которые должны иметь как динамическое, так статическое содержимое. Например, узкий левый фрейм может содержать оглавление сайта, а широкий правый будет предназначен для вывода информации. Возможны также любые другие конфигурации.

**Фрейм - подокно, в которое загружается отдельная страничка.**

Разбиение окна браузера на фреймы реализуется следующим образом :

- создаются отдельные html странички для каждого фрейма.
- создается HTML-файл (обычно это первая страничка сервера по имени index.htm):

```
<HTML>
<HEAD>
  Страничка с фреймами
</HEAD>
</HTML>
```

Обратите внимание на то, что тег BODY отсутствует, вместо него используется тег FRAMESET, содержащий описание внутренних фреймов:

**Структура HTML-страницы с фреймами:**

```
<HTML>
  <HEAD>
    описание заголовка
  </HEAD>
  <FRAMESET>
    <FRAME>первый фрейм</FRAME>
    <FRAME>второй фрейм</FRAME>
  </FRAMESET>
</HTML>
```

- В документе *index.htm* указывается сколько документов откроется одновременно в окне браузера, сколько места будет занимать каждый, каким образом они будут располагаться: можно разделить экран на несколько вертикальных или несколько горизонтальных фреймов.

Например, в документе *index.htm* документы *name.htm*, *menu.htm*, *content.htm* могут быть расположены:



Атрибуты тега **FRAMESET**:

**COLS** - подразделяет экран на определенное количество вертикальных колонок.

**ROWS** - подразделяет экран на определенное количество горизонтальных колонок.

Размеры колонок перечисляются через запятую (сколько колонок, столько и размеров) и могут быть выражены:

- в **пикселях**,
- в **процентах** относительно размера окна,
- в **относительных величинах**.

Например, тэг `<FRAMESET ROWS="100,240,140">` создает сетку из 3 кадров, **высота** которых равна 100, 240 и 140 пикселей соответственно, `<FRAMESET COLS="25%,75%">` создает 2 кадра с **шириной** 25 и 75 процентов от ширины окна браузера, а `<FRAMESET COLS="*,2*">` - 2 кадра с шириной 1/3 и 2/3 ширины окна браузера.

Эти способы можно использовать и **совместно** - например, `<FRAMESET COLS="128,*">` указывает оставить 128 пикселей слева под первый фрейм, а все остальное пространство - под второй. При определении обоих атрибутов rows и cols, например, `<FRAMESET ROWS="*,2*" cols="*,2*">` получается сетка кадров.

**BORDER="ширина"** - указывает **ширину обрамления всех рамок** для всех кадров, в пикселях;

**FRAMEBORDER=yes** или **FRAMEBORDER=no** - включает или выключает **отображение обрамления** кадров. В случае yes рамка имеет трехмерную форму, иначе она невидима, то есть имеет цвет фона окна по умолчанию. Некоторые браузеры "понимают" эту опцию только в виде `frameborder=1` или `frameborder=0`, поэтому обычно указывают оба способа;

**FRAMESPACING="ширина"** - указать **ширину промежутка** между смежными кадрами в пикселях. Не действует в Netscape Navigator.

Внутри тэга `<FRAMESET>` находятся **описания отдельных кадров**, каждое в собственном тэге `<FRAME>`, закрывать который не нужно. Число тэгов `<FRAME>` должно быть равно числу кадров, определенных в тэге `<FRAMESET>`, при этом считается, что кадры описываются слева направо и сверху вниз.

Атрибуты тэга `<FRAME>`:

**SRC="url"** указывает **URL исходного документа** для данного кадра;

Если этот атрибут отсутствует, то отображается пустой фрейм.

Пример. Создадим первый вариант:

```
<HTML>
<HEAD>
<TITLE>Страничка с фрей-
мами</TITLE>
</HEAD>
<FRAMESET
ROWS="25%,*,25%">
<FRAME SRC="name.htm">
<FRAME SRC="menu.htm">
<FRAME SRC="content.htm">
</FRAMESET>
</HTML>
```



Пример. Создадим второй вариант:

```

<HTML>
<HEAD>
<TITLE>Страничка с
фреймами</TITLE>
</HEAD>
<FRAMESET
COLS="100*,50%">
<FRAME SRC="name.htm">
<FRAME SRC="menu.htm">
<FRAME SRC="content.htm">
</FRAMESET>
</HTML>

```



**NAME**="строка" указывает **имя кадра**. Это необходимо сделать, если предполагается ссылаться из одних кадров на другие;

**SCROLLING**="значение" управляет **линейками прокрутки** кадра. Значение может быть задано в виде yes (линейки есть всегда), no (никогда) или auto (если необходимо);

**NORESIZE** запрещает изменять **размеры кадра**. Опция NORESIZE, указанная для данного кадра, влияет также и на все кадры, смежные с ним;

**FRAMEBORDER** может указываться также внутри тэга <FRAME>, со всеми замечаниями, которые сделаны относительно нее. Указание этой опции в тэге <FRAME> отменяет указание, сделанное в тэге <FRAMESET> для данного кадра и всех, смежных с ним;

**MARGINHEIGHT**="ширина" и **MARGINWIDTH**="ширина" задают **размещение** по верхней, нижней и боковым сторонам кадра **областей свободного пространства**, ширина которых указывается в пикселях.

После того, как тэг <FRAMESET> закрыт, можно использовать тэг <noframes>...</noframes>, определяющий содержимое, которое будет выводиться браузерами, не поддерживающими кадры.

Для создания более сложных конфигураций кадров тэги <FRAMESET> могут **вкладываться** друг в друга - внутренний тэг <FRAMESET> может быть вложен вместо любого из тэгов <FRAME>.

Пример. Создадим третий вариант: сначала разделим окно на колонки, вторая колонка будет содержать в себе документ content.htm, а первую колонку мы разобьем на два ряда, и поместим в них документы name.htm и menu.htm.

```

<HTML>
<HEAD>
<TITLE>Страничка с
фреймами</TITLE>
</HEAD>
<FRAMESET COLS="*,50%">
<FRAMESET ROWS="100,*">
<FRAME SRC="name.htm">
<FRAME SRC="menu.htm">
</FRAMESET>
<FRAME SRC="content.htm">
</FRAMESET>
</HTML>

```



Пример документа с фреймами:

```

<html>
<head></head>

```



```

<FRAMESET ROWS="40,*" border="0" frameborder="0" frameborder="no">
  <frame src="reclama.html" name="top" scrolling="no" noresize>
  <FRAMESET COLS="128,*" border="0" frameborder="0" framebor-
  der="no">
    <frame src="menu.html" name="menu" scrolling="auto">
    <frame src="content.html" name="main" scrolling="yes">
  </frameset>
</frameset>
<noframes>
  Извините, Ваш браузер не поддерживает кадры!
</noframes>
</html>

```

Здесь верхний кадр высотой 40 пикселей может служить, например, для вывода рекламы и связан с файлом reclama.html. Остальная часть окна разбита на 2 колонки. Левая колонка имеет ширину 128 пикселей и может быть предназначена для вывода меню (документ menu.html). Третий кадр занимает основную часть окна и предназначен для вывода информации (файл content.html).

Для создания **ссылки из одного кадра в другой** достаточно указать в тэге ссылки опцию вида **target="имя кадра"**. Например, для ссылки из кадра menu в кадр main достаточно написать в документе menu.html ссылку вида `<a href="URL нового документа" target="main">...</a>`. При щелчке по этой ссылке содержимое правого кадра, то есть, документ content.html, будет заменен на новый документ.

Для атрибута **target** существуют следующие значения:

**\_self** - загрузка новой страницы в тот же кадр, откуда делается ссылка,  
**\_parent** - загрузить в родительский для данного кадр; если такого нет - результат действия аналогичен **\_self**,

**\_top** - загрузка новой страницы в полное окно,

**\_blank** - загрузка новой страницы в новое окно браузера,

Можно задать атрибут **target** по умолчанию. Это удобно, если в фрейме содержится много ссылок, т.е. чтобы для каждой не добавлять атрибут **target**, нужно добавить тэг

**<BASE TARGET="имя кадра" >** между тэгами **<HEAD>** и **</HEAD>**.

Например, для вышеописанного примера в файле menu.html в области **<HEAD>** и **</HEAD>** нужно вставить тэг **<BASE TARGET="main">**. Тогда все ссылки организованные в кадре **menu** будут открываться в кадре **main**.

#### **Задание для самостоятельного выполнения.**

По результатам предыдущих лабораторных работ Вы должны иметь:

1. Страницу о себе, где используются возможности форматирования HTML – first.html;
2. Страницу с графикой и списками – graphic.html;
3. Страницу с таблицей – tab.html;
4. Страницу с содержанием всего сайта, в которой должны быть установлены ссылки на соответствующие документы – links.html.

Создайте документ index.html, содержащий 2 вертикально расположенных фрейма. Левый фрейм (узкий) предназначен для вывода меню, правый (широкий) - для вывода содержимого документов, на которые Вы ссылаетесь из меню. Размеры фреймов должны быть изменяемы. Оставьте небольшое свободное пространство между фреймами.

Создайте в левом фрейме меню для навигации по всем страницам Вашего сайта. В левом фрейме должен быть загружен файл links.html, в правом файл first.html. Ссылки из левого фрейма должны открываться в правом фрейме. Проверьте работу документа с фреймами и покажите результат преподавателю.

## Лабораторная работа №15 Знакомство со средой Boson Network Designer.

**Тема работы:** знакомство со средой Boson Network Designer.

**Цель работы:** знакомство со средой Boson Network Designer, создание топологии, назначение компьютерам адресов, пингование компьютеров.

**Теоретическая часть:**

Программные продукты фирмы Boson дают возможность создавать сетевые топологии из широкого спектра маршрутизаторов и коммутаторов компании Cisco, рабочих станций и сетевых соединений типа Ethernet, Serial, ISDN, Frame Relay. Эта функция может быть выполнена как для обучения, так и для работы. Например, чтобы сделать настройку сети ещё на этапе планирования или чтобы создать копию рабочей сети с целью устранения неисправности.

### **Boson Network Designer.**

Знакомство с программными продуктами фирмы Boson начнём с программы Boson Network Designer. Именно в ней создаётся топология сети, которая затем программируется (конфигурируется) в программе Boson NetSim.

Для запуска Boson Network Designer необходимо вызвать исполняемый файл, Net Designer.exe. Общий вид программы можно увидеть на рис.12.

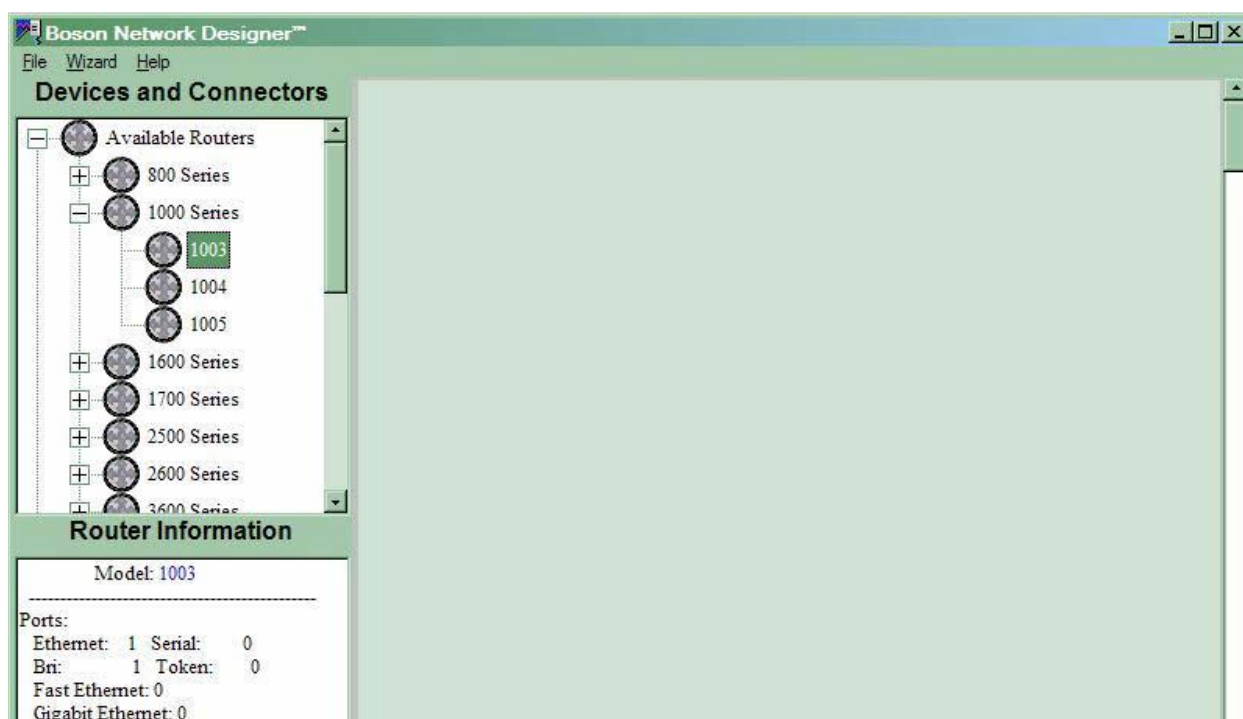


Рисунок 12. Общий вид программы Boson Network Designer.

В левой части главного окна вы видите два раздела:

В верхнем разделе – “Devices and Connectors” – находится дерево, которое содержит устройства, из которых создаётся топология будущей сети. Available Routers – доступные в среде маршрутизаторы.

Available Switches – доступные в среде коммутаторы.

Available Connectors – соединения.

Other Devices – другие устройства (компьютеры PC)

В нижнем разделе – “..... Information” высвечиваются параметры выбранного устройства.

Для построения топологии следует, используя метод Drag-and-Drop (выбираем объект, удерживаем левую кнопку мыши, и перетаскиваем в нужное окно), добавить не-

обходимые элементы из раздела Devices and Connectors в правое поле главного окна программы (см. рис. 13).

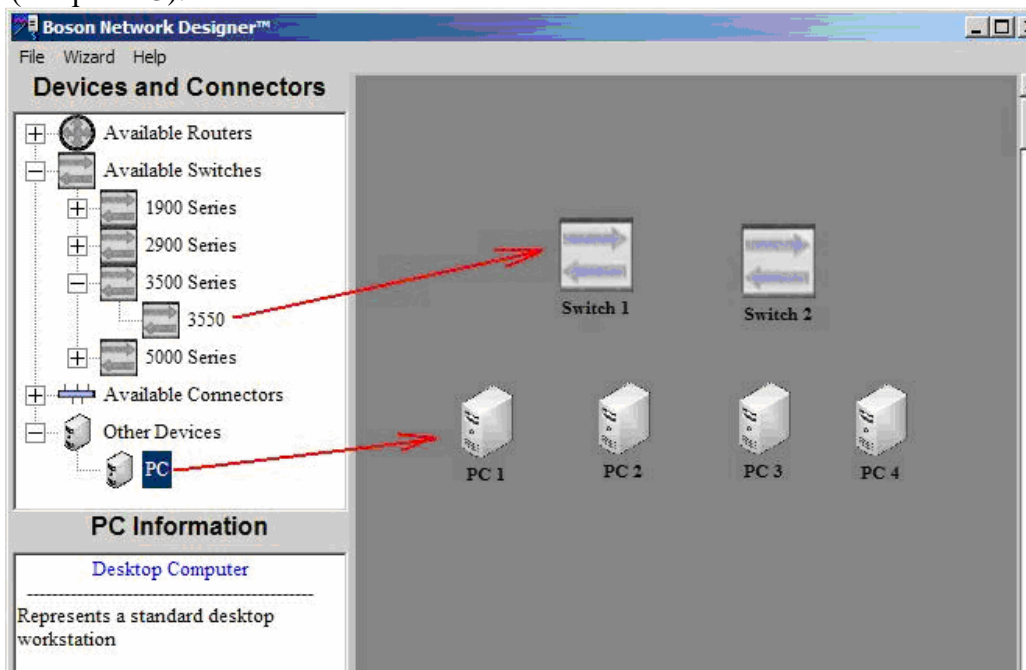


Рисунок 13. Добавление элементов сети.

При добавлении каждого элемента вы можете дать ему имя и установить параметры. Это можно сделать в окне, которое появляется при добавлении устройства. Двойной щелчок на устройстве приведёт к появлению диалогового окна, в котором можно увидеть имя и модель устройства, интерфейсы устройства и к каким другим устройствам они присоединены. Здесь можно осуществить подсоединение свободных интерфейсов устройства. При нажатии правой кнопки мыши выпадает контекстное меню в котором можно осуществить подсоединение свободных интерфейсов устройства либо уничтожить существующие соединения. Можно также удалить устройство.

Добавлять устройства можно также с помощью мастера (меню Wizard-> Add Device Wizard). С этим пунктом разберитесь самостоятельно.

Добавив элементы, мы связываем их с помощью соединяющих связей. Это можно сделать несколькими способами:

Первый - если перетащить элемент соединения из раздела “Available Connectors” в правое рабочее поле, то появится мастер, в котором за два шага можно осуществить нужное соединение. В начале следует выбрать нужное устройство и в нём желаемый интерфейс (занятые интерфейсы помечены звёздочкой). Нажать кнопку next. Выбрать второе устройство и интерфейс в нём нажать кнопку – Finish.

Второй – если дважды нажать мышкой либо выбрать правой кнопкой контекстное меню на элементе, к которому или от которого мы создаём соединение. Вначале следует выбрать свободный интерфейс устройства, далее из списка выбрать второе устройство и интерфейс в нём и нажать кнопку – Finish. Добавлять соединения можно также с помощью мастера (меню Wizard->Make Connection Wizard). С этим пунктом разберитесь самостоятельно.

Каждое соединение имеет свой цвет. Ethernet имеет синий цвет, а последовательное соединение – чёрный цвет. Создание последовательных интерфейсов имеет некоторые особенности, рассмотренные в следующих работах. Поэтому для начала используйте только Ethernet соединения.

После создания сети её следует сохранить, выбрав пункт меню File -> Save. Файл, в котором сохраняется топология, имеет расширение \*.top . Это текстовый файл с понятным содержанием. Откройте его в любом текстовом редакторе (notepad) и изучите его содержание.

После сохранения вашей сети вы можете сразу открыть готовую сеть в симуляторе Boson NetSim для программирования. Для этого следует выбрать в меню File → Load NetMap Into the Simulator. Заметим, что симулятор сам автоматически при этом не стартует и должен быть предварительно запущен.

### Симулятор Boson Netsim.

Запустите программу Boson\_NetSim.exe.

Программа Boson Netsim симулирует работу с интерфейсом командной строки (ИКС) операционной системы IOS, установленной на всех коммутаторах и маршрутизаторах компании Cisco. Существует всего 4 метода загрузки ИКС для соединения с устройством eDevice (в симулятор должна быть предварительно загружена топология, созданная в дизайнере).



Рисунок 14. Кнопка NetMap.

1. На Панели Инструментов есть кнопка NetMap. (см. рис.14). Она запускает утилиту NetMap Viewer, с помощью которой можно увидеть сетевую топологию, загруженную в симулятор. Непосредственно внутри утилиты NetMap Viewer щелкните правой кнопкой мыши на любом eDevice, выберите в контекстном меню пункт Configure и подключитесь к выбранному устройству

2. На Панели Инструментов есть кнопки eRouters, eSwitches и eStaions для получения списка устройств в текущей топологии сети. Например, когда вы выберете Router 1, то вы подключитесь к Router 1 (рис. 15)

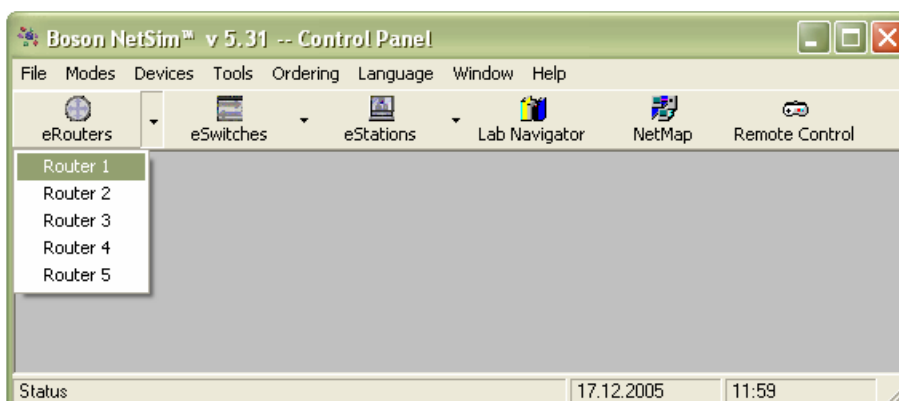


Рисунок 15. Список роутеров.

3. В главном меню программы есть пункт Devices и далее пункты для выбора типа устройства и устройства (рис. 16).

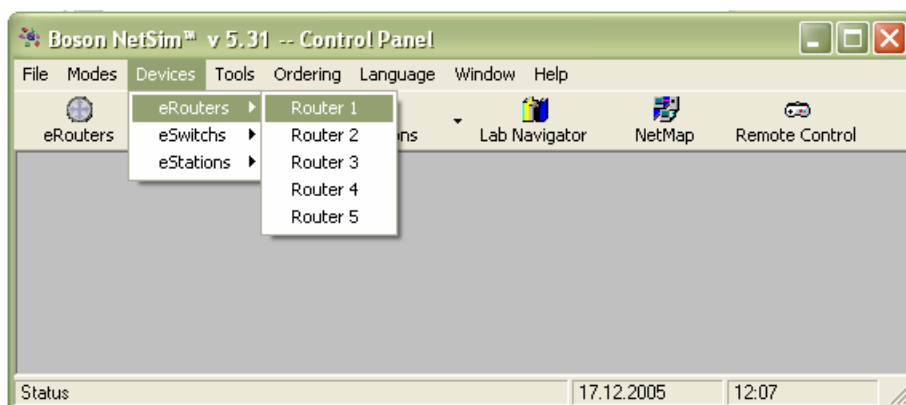


Рисунок 16. Выбор устройства

4. На Панели Инструментов есть кнопка Remote Control для запуска диалогового окна Remote Control. В окне есть пункт “Telnet to ...” для выбора устройства. Подключившись к устройству, вы можете работать с ним так, как за консолью реального устройства. Симулятор обеспечивает поддержку практически всех команд, доступных на реальных устройствах. (См. Пункт меню tools->available commands для ознакомления со списком поддерживаемых команд). ИКС работает в двух режимах, переключаемых в меню Modes: для начинающих Beginner Mode (WiW) и опытных Advanced Mode(Telnet) пользователей. Beginner Mode (WiW) – режим новичка. Режим WiW (Window-in- Window Interface) открывает главное окно Boson NetSim™ и отобразит экран ИКС для выбранного устройства. Четыре способа переключения между различными устройствами рассмотрены выше. Вы можете также использовать F-клавиши, чтобы переключаться между устройствами. F1 - для Устройства 1, F2 - для Устройство 2 и так далее.

Advanced Mode(Telnet) – запускает программу стандартного Telnet – клиента из вашей операционной системы, которая скроет главное окно и откроет диалоговое окно Remote Control. Вы можете выключить Remote Control, нажимая кнопку Close, или используя подпункт Toolbars -> Remote Control пункта Modes главного меню. Этот режим запускает разные окна Telnet для каждого устройства, которое Вы хотели бы конфигурировать. Причём возможен запуск нескольких Telnet для одного устройства. Когда Вы закончите конфигурирование устройства, Вы может закрыть окно Telnet либо, закрыв окно, либо, перейдя в режим телнета (Ctrl+J), дать команду quit. Режим также имеет встроенный метод переключения между устройствами в одном и том же окне Telnet. Когда Вы готовы соединиться с другим устройством, Вы должны использовать сочетание клавиш CTRL-Q (зажмите одновременно клавиши CTRL и Q). Это отобразит листинг всех доступных устройств в текущем окне telnet.

В этом режиме существует ещё три способа переключения между различными устройствами, которые рассмотрены выше: кнопка NetMap, кнопки eRouters, eSwitches и eStaions и пункт Devices главного меню.

Интересно отметить, что при выборе режима Advanced Mode симулятор стартует свой телнет сервер. Поэтому конфигурировать устройства в симуляторе можно извне и даже удалённо из другой машины, набрав в интерпретаторе командной строки cmd.exe команду telnet hostname.

#### **Работа с файлами в симуляторе.**

В меню File есть такие пункты:

Load Single Device Config (merge) – загрузить из файла конфигурацию одного устройства и добавить её в конфигурацию устройства.

Load Single Device Config (overwrite) – загрузить из файла конфигурацию одного устройства топологии, заменив существующую конфигурацию устройства.

Load Multi Devices Configs – загрузить из файла конфигурацию всех устройств.

Save Single Device Config – сохранить конфигурацию текущего устройства, которые вы ввели в ИКС. Файл будет иметь расширение \*.rtr.

Save Multi Devices Configs – сохранить конфигурацию всех устройств.

Файл будет иметь расширение \*.nws. Программа для каждого устройства создаст отдельный файл с расширением .rtr. Если при сохранении конфигурации вы дали имя папке и в вашей сети, например 10 устройств, то будут созданы файлы конфигурации для каждого устройства name1.rtr, name2.rtr, ... name10.rtr. Интересно отметить, что при хорошем знании команд IOS и структуры rtr файлов, можно модифицировать rtr файлы в любом текстовом редакторе и не использовать командную строку.

### Практическая часть.

1. В программе Network Designer добавим на правое рабочее поле программы из ветви Available Switches два коммутатора 3550 серии 3500. Примем имена по умолчанию. Имеем – Switch1 и Switch2.

2. Добавим на рабочее поле из ветки Other Devices, четыре компьютера. Примем имена по умолчанию. Имеем – PC1, PC2, PC3 и PC4.

3. Соединим устройства в Ethernet сеть, как указано на рис.17.

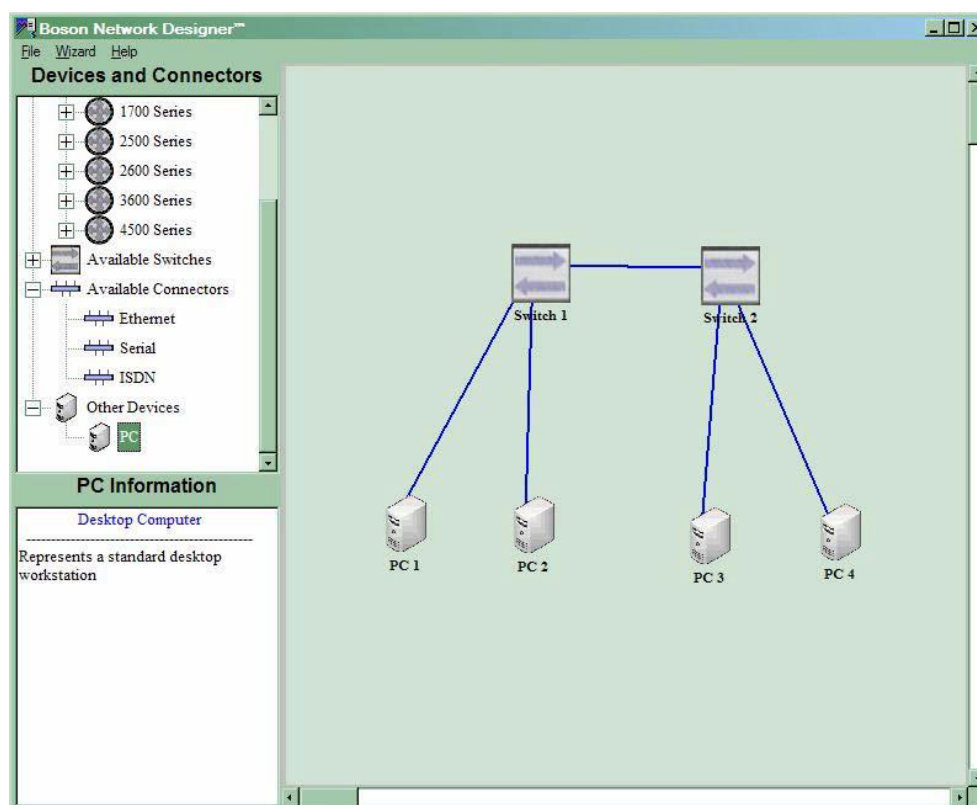


Рисунок 17. Топология сети.

Сохраним созданную топологию, выбрав пункт меню File-> Save.

4. Откроем симулятор (NetSim) и с помощью пункта меню File->Load NetMap загрузим в него созданную.

5. Проверьте загруженную топологию, нажав кнопку Netmap на панели инструментов.

6. На панели инструментов выбираем eStations -> PC1. Появится окно компьютера PC1.

7. Для конфигурирования компьютера используем команду *winipcfg*. Наберём её в ИКР и нажмём Enter. Появится окно (см. рис. 18)



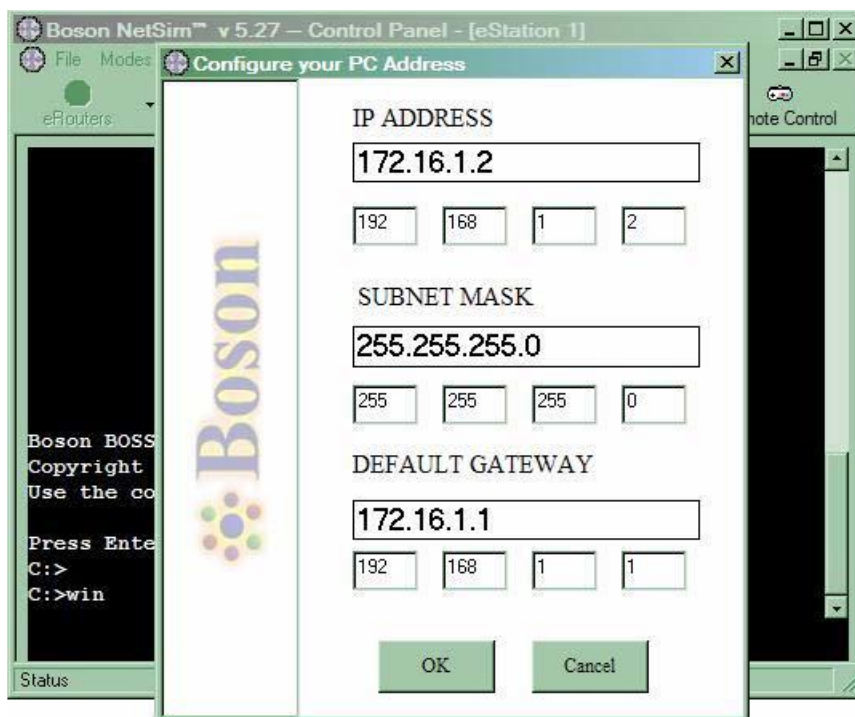


Рисунок 18. Конфигурация компьютера.

8. В четырёх полях *IP ADDRESS* – вводим *IP* адрес компьютера 192.168.1.2. Поле *SUBNET MASK* –оставим без изменения 255.255.255.0. Поле *DEFAULT GATEWAY* –адреса шлюза не важно, так как создаваемая сеть не требует маршрутизации. Адрес можно назначить и из командной строки командой *ipconfig*, например *ipconfig /ip 192.168.1.2 255.255.255.0*

Таким же образом настроим каждый компьютер.

Устройство	IP ADDRESS	SUBNET MASK
PC1	192.168.1.2	255.255.255.0
PC2	192.168.1.3	255.255.255.0
PC3	192.168.1.4	255.255.255.0
PC4	192.168.1.5	255.255.255.0

9. На каждом компьютере посмотрите назначенные адреса командой *ipconfig* без параметров.

10. Если сделано всё правильно вы сможете пропинговать любой компьютер из любого компьютера. Например, зайдите на компьютер *PC4* и пропингуйте компьютер *PC1*. Вы должны увидеть то, что изображено на рисунке 19.

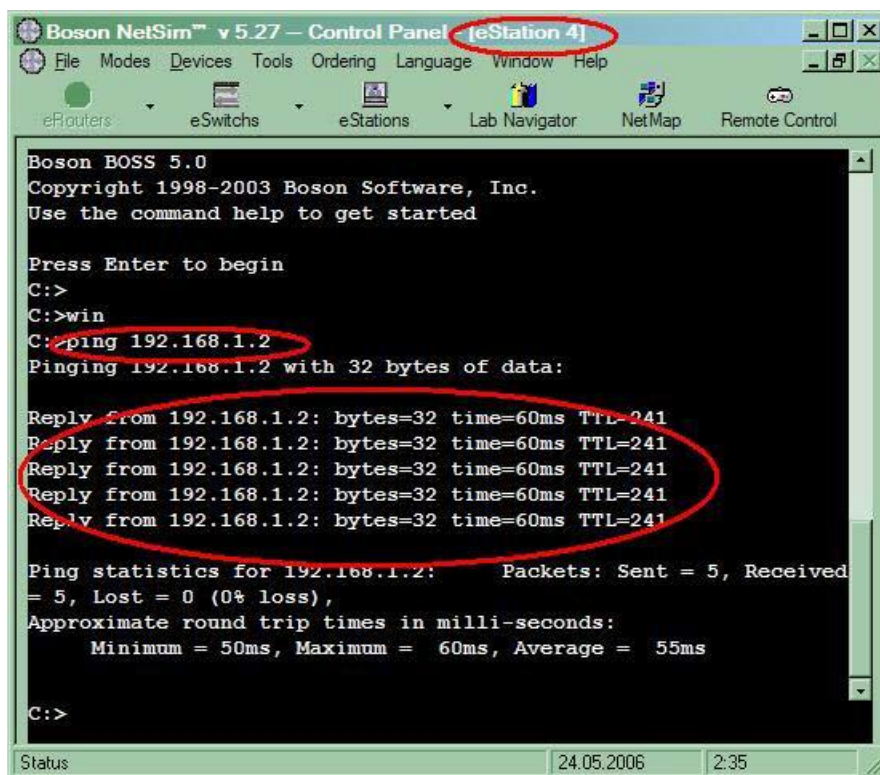


Рисунок 19. Удачное «пингование» PC1.

11. Сохраним в отдельной папке с помощью команды *File->Save Multi Devices Configs*. Посмотрите содержимое *rtr* файлов.

12. Воспользуемся для каждого устройства командой *Save Single Device Config* и дадим *rtr* файлам конфигураций имена, совпадающие с именами устройств. Например, для коммутатора *Switch1* возьмём *Switch1.rtr*.

13. Откроем в *Boson Network Designer* нашу топологию и добавим туда ещё один компьютер *PC5* и соединим его к коммутатору *Switch1*. Сохраним топологию в новом *top* файле и загрузим её в *Boson NetSim*.

14. На панели управления пошагово выбираем каждое устройство. Далее в меню *File* выбираем *Load Single Device Config(overwrite)*, для загрузки в устройство конфигурации из ранее сохранённых *rtr* файлов. Удобно пользоваться файлами из пункта 12. Назначьте компьютеру *PC5* адрес *192.168.14.6*, маска *255.255.255.0*.

15. Если сделано всё правильно вы сможете пропинговать любой компьютер из любого компьютера.

### Контрольные вопросы

1. Какое максимальное количество устройств в сети поддерживает *Boson NetSim*?
2. Из каких программ состоит *Boson NetSim*?
3. Какие типы сетевых устройств и соединений можно использовать в *Boson Network Designer*?
4. Какими четырьмя способами можно перейти к интерфейсу командной строки устройства.
5. Как конфигурировать устройства из другого компьютера?
6. Как изменить имя компьютера в интерфейсе командной строки?
7. Как добавить в топологию и настроить новое устройство?
8. Как закомментировать команду в *rtr* файле.



## Порядок выполнения работы

1. Изучить теоретическую и практическую часть.
2. Сдать преподавателю теорию работы путём ответа на контрольные вопросы.
3. Выполнить в Boson практическую часть.
4. Получите вариант (1-12) и выполните в Boson задание для самостоятельной работы
5. Предъявите преподавателю результат выполнения задания для самостоятельной работы. Покажите ему свои rtr, pncs и top файлы. Продемонстрируйте ему, что любой компьютер пингуется из любого компьютера.
6. Оформите отчёт.

## Задание для самостоятельной работы

1. Создайте топологию на рис.20.
2. Назначьте компьютерам адреса, согласно варианту (v=1-12)

Устройство	IP ADDRESS	SUBNET MASK
PC1	v.1.1.1	255.255.255.0
PC2	v.1.1.2	255.255.255.0
PC3	v.1.1.3	255.255.255.0
PC4	v.1.1.4	255.255.255.0
PC5	v.1.1.5	255.255.255.0
PC6	v.1.1.6	255.255.255.0
PC7	v.1.1.7	255.255.255.0

Например, для варианта 7 (v=7) и компьютера PC1 имеем IP ADDRESS 7.1.1.1

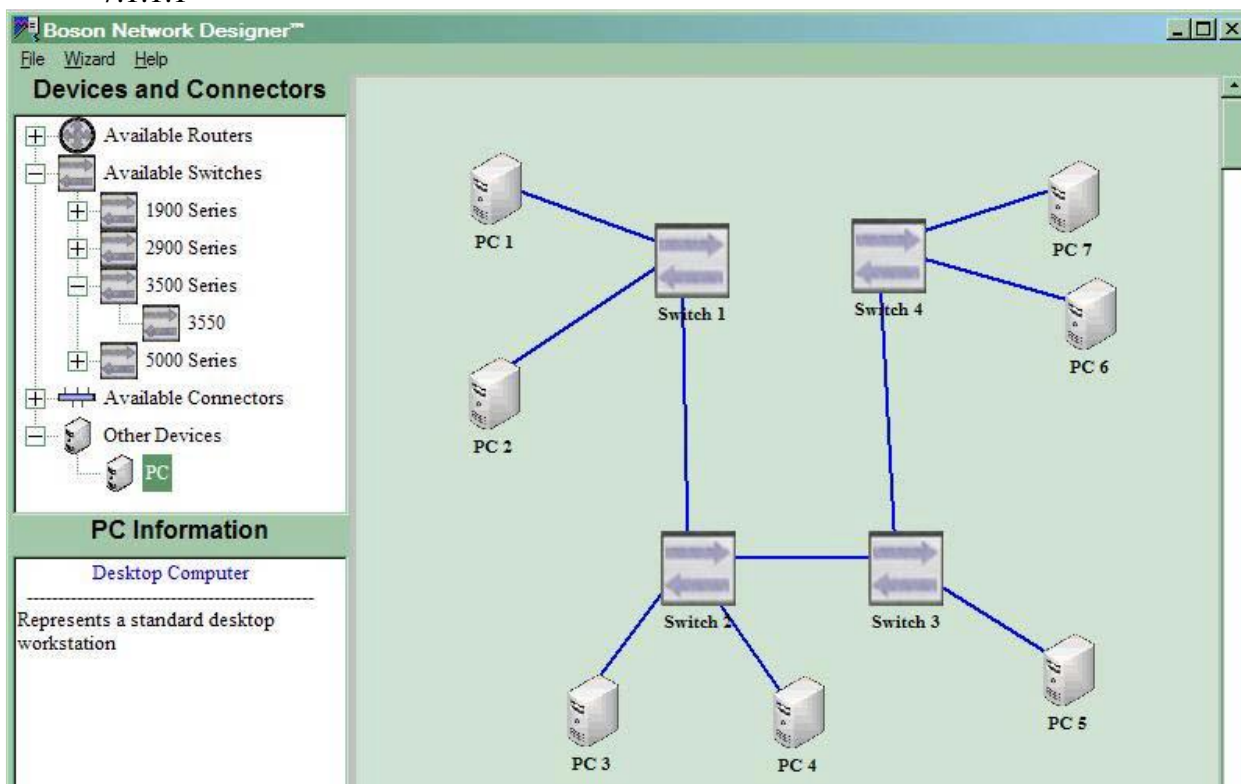


Рисунок 20. Топология сети.

3. Назначьте компьютерам разные имена в командной строке.
4. Если сделано всё правильно вы сможете пропинговать любой компьютер из любого компьютера.

## Лабораторная работа №16 Введение в межсетевую операционную систему IOS компании Cisco.

**Тема работы:** Введение в межсетевую операционную систему IOS компании Cisco.

**Цель работы:** Ознакомление с сетевым устройством Cisco, конфигурация интерфейсов, настройка IP адресов интерфейсов, применение команды Telnet.

**Теоретическая часть:**

При первом входе в сетевое устройство пользователь видит командную строку пользовательского режима вида: `Switch>` Команды, доступные на пользовательском уровне являются подмножеством команд, доступных в привилегированном режиме. Эти команды позволяют выводить на экран информацию без смены установок сетевого устройства. Чтобы получить доступ к полному набору команд, необходимо сначала активизировать привилегированный режим. Press ENTER to start.

```
Switch>
Switch> enable
Switch#
Switch# disable
Switch>
```

Здесь и далее вывод сетевого устройства будет даваться обычным шрифтом, а ввод пользователя **жирным** шрифтом. О переходе в этот режим будет свидетельствовать появление в командной строке приглашения в виде знака #. Из привилегированного уровня можно получать информацию о настройках системы и получить доступ к режиму глобального конфигурирования и других специальных режимов конфигурирования, включая режимы конфигурирования интерфейса, подинтерфейса, линии, сетевого устройства, карты маршрутов и т.п.

Для выхода из системы IOS необходимо набрать на клавиатуре команду *exit* (выход). `Switch> exit`

Независимо от того, как обращаются к сетевому устройству: через консоль терминальной программы, подсоединённой через ноль-модем к COM-порту сетевого устройства, либо в рамках сеанса протокола *Telnet*, устройство можно перевести в один из режимов.

Нас интересуют следующие режимы.

Пользовательский режим — это режим просмотра, в котором пользователь может только просматривать определённую информацию о сетевом устройстве, но не может ничего менять. В этом режиме приглашение имеет вид типа *Switch>*.

Привилегированный режим — поддерживает команды настройки и тестирования, детальную проверку сетевого устройства, манипуляцию с конфигурационными файлами и доступ в режим конфигурирования. В этом режиме приглашение имеет вид типа *Switch#*.

Режим глобального конфигурирования — реализует мощные однострочные команды, которые решают задачи конфигурирования. В этом режиме приглашение имеет вид типа *Switch (config) #*.

Команды в любом режиме IOS распознаёт по первым уникальным символам. При нажатии табуляции IOS сам дополнит команду до полного имени. При вводе в командной строке любого режима имени команды и знака вопроса (?) на экран выводятся комментарии к команде.

При вводе одного знака результатом будет список всех команд режима. На экран может выводиться много экранов строк, поэтому иногда внизу экрана будет появляться подсказка - More -. Для продолжения следует нажать enter или пробел. Команды режима глобального конфигурирования определяют поведение системы в целом.

Кроме этого, команды режима глобального конфигурирования включают команды перехода в другие режимы конфигурирования, которые используются для создания конфигураций, требующих многострочных команд. Для входа в режим глобального конфигурирования используется команда привилегированного режима **configure**.

При вводе этой команды следует указать источник команд конфигурирования: **terminal** (терминал), **memory** (энергонезависимая память или файл), **network** (сервер tftp (Trivial ftp -упрощённый ftp) в сети). По умолчанию команды вводятся с терминала консоли. Например

```
Switch# configure terminal
Switch(config)#(commands)
Switch(config)# exit
Switch#
```

Команды для активизации частного вида конфигурации должны предваряться командами глобального конфигурирования. Так для конфигурации интерфейса, на возможность которой указывает приглашение Switch( configif)#, сначала вводится глобальная команда для определения типа интерфейса и номера его порта:

```
Switch# conf t
Switch(config)# interface type port
Switch( config-if)# (commands)
Switch( config-if)# exit
Switch(config)# exit
```

Для ограничения доступа к системе используются пароли. Команда **line console** устанавливает пароль на вход на терминал консоли:

```
Switch (config)# line console 0
Switch ( config-line)# login
Switch ( config-line)# password Cisco
```

Команда **line vty 0 4** устанавливает парольную защиту на вход по протоколу Telnet:

```
Switch (config)# line vty 0 4
Switch (config-line)# login
Switch (config-line)# password cisco
```

Команда **enable password** ограничивает доступ к привилегированному режиму:

```
Switch#conf t
Switch(config)# enable password пароль
Далее Ctrl-Z
Switch#ex
```

```
...
Press RETURN to get started
```

```
Switch>en
Password: пароль
Switch#
```

Здесь пароль **пароль** – последовательность латинских символов.

Для установки на сетевом интерфейсе IP адреса используется команда:

```
Router( config-if)#ip address [ip-address] [subnet-mask],
```

Важно иметь возможность контроля правильности функционирования и состояния сетевого устройства в любой момент времени. Для этого служат команды, представленные в таблице 3.

Таблица 3.

Команда	Описание
show version	Выводит на экран данные о конфигурации аппаратной части системы, версии программного обеспечения, именах и источниках конфигурационных файлов и загруженных образах
show running-config	Показывает содержание активной конфигурации
show interfaces	Показывает данные обо всех интерфейсах на устройстве
show protocols	Выводит данные о протоколах третьего сетевого уровня.

### Cisco Discovery Protocol (CDP)

CDP позволяет устройствам обмениваться основной конфигурационной информацией. CDP будет работать без настройки какого ни будь протокола. По умолчанию, CDP включен на всех интерфейсах.

CDP работает на втором (канальном) уровне модели OSI. Поэтому CDP не является маршрутизируемым 14 протоколом и работает только с непосредственно подключенными устройствами. Протокол CDP связывает физическую среду передачи данных более низкого уровня с протоколами более высокого сетевого уровня. Поэтому устройства, поддерживающие разные протоколы третьего уровня, могут узнавать друг друга.

При запуске устройства протокол CDP запускается автоматически. После этого он может автоматически определить соседние устройства, на которых также выполняется протокол CDP. Среди найденных устройств будут не только те, которые работают с протоколом IP. CDP позволяет администраторам иметь доступ к данным о другом сетевом устройстве, к которому есть непосредственное соединение.

Для вывода информации о соседних устройствах, обнаруженных по протоколу CDP, используется семейство команд **show cdp**. Оно выводит следующие данные по каждому порту и каждому подсоединённому к нему устройству: Идентификаторы устройства, список адресов, идентификатор порта, перечень функциональных возможностей, аппаратная платформа устройства.

### Команды ping и traceroute.

Для диагностики возможности установления связи в сетях используются протоколы тип запрос-ответ или протокол эхо-пакетов. Результаты работы такого протокола могут помочь в оценке надёжности пути к другому устройству, величин задержек в целом и между промежуточными устройствами. Для того чтобы такая команда работала, необходимо, чтобы не только локальное сетевое устройство знало, как попасть в пункт назначения, но и чтобы устройство в пункте назначения знало, как добраться до источника.

Команда ping посылает ICMP (Internet Control Message Protocol) эхо-пакеты для верификации соединения. В приведённом ниже примере время прохождения одного эхо-пакета превысило заданное, о чём свидетельствует точка (.) в выведенной информации, а четыре пакета прошли успешно, о чём говорит восклицательный знак (!).

```
Switch> ping 172.16.101.1
```

```
Type escape sequence to abort.
```

```
Sending 5 100-byte ICMP echoes to 172.16.10.1 timeout is 2 seconds:
```

```
.!!!!
```

```
Success rate is 80 percent, round-trip min/avg/max = 6/6/6
```

```
Ms
```

Таблица 4. Результаты команды ping

Символ	Значение
!	Успешный приём эхо-ответа
.	Превышено время ожидания
U	Пункт назначения недостижим
C	Перегрузка сети
I	Выполнение команды прервано администратором
?	Неизвестный тип пакета
&	Пакет превысил значение параметра времени жизни TTL пакета

Команды *traceroute* показывает адреса промежуточных интерфейсов (хопов) на пути пакетов в пункт назначения.

```
Switch> traceroute 172.16.101.1
```

Расширенная версия команды *ping* поддерживается только в привилегированном режиме. Для того, чтобы войти в расширенный режим, необходимо в строке подсказки *Extended commands* ввести букву "y"(Yes) Команда в режиме диалога опрашивает значения параметров. Важно отметить, что эта команда позволяет, находясь на одном устройстве, проверять связь между сетевыми интерфейсами на других устройствах.

```
Router# ping
Protocol [ip]:
Target IP address: 2.2.2.2
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address:1.1.1.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose [none]:
Sweep range of sizes [n]:
```

### Команда *telnet*.

Протокол виртуального терминала *telnet*, входящий в состав протоколов TCP/IP, позволяет установить соединение между сетевым устройством *telnet* клиента и сетевым устройством *telnet* сервера, что обеспечивает возможность работы в режиме виртуального терминала. *Telnet* используется для удалённого управления сетевым устройством либо для проверки связи на уровне приложений. Успешное установление *Telnet*-соединения позволяет вам управлять удалённым устройством так, как будто вы находитесь за его консолью.

Сетевые устройства *Cisco* способны поддерживать одновременно до пяти входных сеансов протокола *Telnet*.

### Практическая часть.

#### Соединение с сетевым устройством Cisco.

Создайте в *Boson network designer* топологию, изображённую на рисунке с использованием модели маршрутизатора 805 из серии 800. Назовите устройства 16 так, как вы видите на рисунке 21: для двух первых устройств, примите имена по умолчанию, а третье назовите *Router 4*.

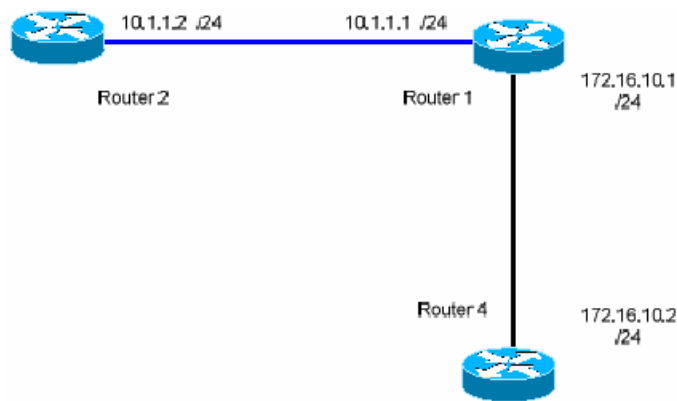


Рисунок 21. Топология сети.

Синяя линия означает Ethernet соединение. Чёрная означает последовательное соединение. При создании последовательного соединения *designer* спросит какой тип соединения вы хотите установить: последовательное соединение **точка-точка** (*serial cable*) или **точка-многоточка** (*frame relay*). Выбираем *serial cable*.

Выбираем второе устройство. Определяемся, какой маршрутизатор будет выполнять функции *DCE* устройства. Это устройство задаёт синхронизацию. В симуляторе для него будет необходимо определить частоту синхронизации. Сохраните топологию и загрузите её в симулятор.

Проверьте загрузку с помощью команды *NetMap* на панели инструментов.

#### Знакомство с сетевым устройством Cisco.

1. Для выбора сетевого устройства *Router1* нажмите кнопку *eRouters* в верхней части экрана и выберите *Router1*. Откроется окно сетевого устройства 1 и появится текст "*Press Enter to Start*".

2. Нажмите мышкой в середине экрана Сетевого устройства *Router1* и вы увидите *Press Enter to Start*

Нажмите *<Enter>*.

Вы увидите

*Router>*

Теперь вы подключены к сетевому устройству и находитесь в командной строке режима пользователя. Здесь "*Router*" – это имя Сетевого устройства, а ">" означает, что вы находитесь в режиме пользователя.

3. Теперь введите команду *enable*, чтобы попасть в привилегированный режим.

*Router>enable*

*Router#*

4. Чтобы вернуться в режим пользователя, просто напечатайте *disable*. Из режима пользователя введите *logout* или *exit*, чтобы покинуть сетевое устройство.

*Router#disable*

*Router>*

*Router>exit*

*Router con0 is now available*

*Press Enter to Start*

#### Основные команды сетевого устройства

1. Войдите сетевое устройство *Router1*

*Router>*

2. Мы хотим увидеть список всех доступных команд в этом режиме. Введите команду, которая используется для просмотра всех доступных команд:

*Router>?*

Клавишу *Enter* нажимать не надо.

3. Теперь войдите в у привилегированный режим

```
Router>enable
```

```
Router#
```

4. Просмотрите список доступных команд в привилегированном режиме

```
Router#?
```

5. Перейдём в режим конфигурации

```
Router#config terminal
```

```
Router(config)#
```

6. *Имя хоста* сетевого устройства используется для локальной идентификации. Когда вы входите в сетевое устройство, вы видите *Имя хоста* перед символом режима (">" или "#"). Это имя может быть использовано для определения места нахождения. Установите "Router1" как имя вашего сетевого устройства.

```
Router(config)#hostname Router1
```

```
Router1(config)#
```

7. Пароль доступа позволяет вам контролировать доступ в привилегированный режим. Это очень важный пароль, потому что в привилегированном режиме можно вносить конфигурационные изменения.

Установите пароль доступу "boson".

```
Router1(config)#enable password boson
```

8. Давайте испытаем этот пароль. Выйдите из сетевого устройства и попытайтесь зайти в привилегированный режим.

```
Router1>en
```

```
Password:*****
```

```
Router1#
```

Здесь знаки: \*\*\*\*\* - это ваш ввод пароля. Эти знаки на экране не видны.

#### Основные Show команды.

Перейдите в пользовательский режим командой *disable*. Введите команду для просмотра всех доступных *show* команд.

```
Router1>show ?
```

1. Команда **show version** используется для получения типа платформы сетевого устройства, версии операционной системы, имени файла образа операционной системы, время работы системы, объём памяти, количество интерфейсов и конфигурационный регистр.

2. Можно увидеть часы

```
Router1>show clock
```

3. Во флеш-памяти сетевого устройства сохраняется файл-образ операционной системы *Cisco IOS*. В отличие от оперативной памяти, в реальных устройствах флеш память сохраняет файл-образ даже при сбое питания.

```
Router1>show flash
```

4. ИКС сетевого устройства по умолчанию сохраняет 10 последних введенных команд

```
Router1>show history
```

5. Две команды позволят вам вернуться к командам, введенным ранее. Нажмите на стрелку вверх или <ctrl> P.

6. Две команды позволят вам перейти к следующей команде, сохранённой в буфере. Нажмите на стрелку вниз или <ctrl> N.

7. Можно увидеть список хостов и IP-Адреса всех их интерфейсов **Router1>show hosts**.

8. Следующая команда выведет детальную информацию о каждом интерфейсе

```
Router1>show interfaces
```

9. Команда

*Router1>show sessions* выведет информацию о каждой *telnet* сессии.

10. Команда

*Router1>show terminal* показывает конфигурационные параметры терминала.

11. Можно увидеть список всех пользователей, подсоединённых к устройству по терминальным линиям

*Router1>show users*

12. Команда

*Router1>show controllers* показывает состояние контроллеров интерфейсов.

13. Перейдём в привилегированный режим. *Router1>en*

14. Введите команду для просмотра всех доступных *show* команд. *Router1#show ?*

Привилегированный режим включает в себя все *show* команды пользовательского режима и ряд новых.

15. Посмотрим активную конфигурацию в памяти сетевого устройства. Необходим привилегированный режим. Активная конфигурация автоматически сохраняется и будет потеряна в случае сбоя электропитания.

*Router1#show running-config*

В строке *more*, нажмите на клавишу пробел для просмотра следующей страницы информации.

16. Следующая команда позволит вам увидеть текущее состояние протоколов третьего уровня. *Router#show protocols*

#### Введение в конфигурацию интерфейсов.

Постараемся понять, как включать (поднимать) интерфейсы сетевого устройства и что надо, чтобы перевести интерфейс в состояние *UP*.

1. На сетевом устройстве *Router1* войдём в режим конфигурации

*Router1#conf t*

*Router1( config)#*

2. Теперь мы хотим настроить *Ethernet* интерфейс. Для этого мы должны зайти в режим конфигурации интерфейса.

*Router1( config)#interface Ethernet 0*

*Router1( config-if)#*

3. Посмотрим все доступные в этом режиме команды

*Router1( config-if)#?*

Для выхода в режим глобальной конфигурации наберите *exit*. Снова войдите в режим конфигурации интерфейса

*Router1( config)#interface e0*

Мы использовали сокращенное имя интерфейса.

4. Для каждой команды мы можем выполнить противоположную команду, поставив перед ней слово *no*. Так команда *Router1( config-if)#no shutdown* включает этот интерфейс.

5. Добавим к интерфейсу описание

*Router1( config-if)#description Ethernet interface on Router 1*

Чтобы увидеть описание этого интерфейса, перейдите в привилегированный режим и выполните команду *show interface* .

*Router1( config-if)#end*

*Router1#show interface*

6. Теперь присоединитесь к сетевому устройству *Router 2* и поменяйте имя его хоста на *Router2*

*Router#conf t*

*Router( config)#hostname Router2*

Войдём на интерфейс *Ethernet 0*.

*Router2( config)#interface e0*



Включите интерфейс.

```
Router2( config-if)#no shutdown
```

Теперь, когда интерфейсы на двух концах нашего *Ethernet* соединения включены на экране появится сообщение о смене состояния интерфейса на активное.

7. Перейдём к конфигурации последовательных интерфейсов. Зайдём на *Router1*. Проверим, каким устройством выступает наш маршрутизатор для последовательной линии связи: окончательным устройством *DTE (data terminal equipment)* либо устройством связи *DCE (data circuit)*

```
Router1#show controllers S0
```

Если видим - ..... *DCE cable*.....- ,то наш маршрутизатор является устройством связи и он должен задавать частоту синхронизации тактовых импульсов, используемых при передаче данных. Частота берётся из определённого ряда частот.

```
Router1#conf t
```

```
Router1(config)#int s0
```

```
Router1( config-if)#clock rate ?
```

Выберем частоту 64000

```
Router1( config-if)#clock rate 64000
```

и поднимаем интерфейс

```
Router1( config-if)#no shut
```

8. Переходим к маршрутизатору *router4* и дадим одноимённое имя.

Поднимаем на нём интерфейс *serial0*. Теперь, когда интерфейсы на двух концах нашего последовательного соединения включены на экране появится сообщение о смене состояния интерфейса на активное.

9. Проверим на каждом устройстве, что сконфигурированные нами интерфейсы находятся в состоянии *UP*.

```
Router1#sh int s0
```

```
Router1#sh int e0
```

```
Router2#sh int e0
```

```
Router4#sh int s0
```

## CDP

1. На маршрутизаторе *router1*, введём команду для вывода состояния всех интерфейсов, на которых работает *CDP*.

```
router1#show cdp interface
```

Мы должны увидеть, что оба интерфейса подняты и посылают *CDP* пакеты.

```
Serial0 is up, line protocol is up
encapsulation HDLC
Sending CDP packets every 60 seconds
Holdtime is 180 seconds
Ethernet0 is up, line protocol is up
encapsulation ARPA
Sending CDP packets every 60 seconds
Holdtime is 180 seconds
```

2. Убедившись, что сетевое устройство посылает и принимает *CDP*- обновления, мы можем использовать *CDP* для получения информации о непосредственно подключённых устройствах. Введите команду *router1#show cdp neighbors*

```
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
S -Switch, H - Host, i - IGMP, r - Repeater
Device ID          Local Intrfce    Holdtme    Capability  Platform  Port ID
router2            Eth0             174        R           805       Eth 0
router4            Ser0             174        R           805       Ser 0
```

Мы сделали всё правильно. Видим, что наш *Ser0*. маршрутизатор *router1* соединён с интерфейсом *Eth 0 (Port ID)* маршрутизатора (*Capability*) *router2 (Device ID)* серии 805

(Platform) через интерфейс *Eth 0 (Local Intrfce)* и с интерфейсом *Ser 0* маршрутизатора *router4* серии 805 через интерфейс

3. На *router1*, введите команду для более детальной информации о соседях *router1#show cdp neighbors detail*.

Эта команда показывает по одному устройству за раз. Она используется для отображения адресной информации сетевого уровня.

На данный момент этот уровень у нас не настроен, поэтому поле *Entry address(es)* пустое. Команда также выводит информацию о версии *IOS*.

4. На *router1*, введите команду, чтобы узнать информацию об устройстве "*router4*" *router1#show cdp entry router4* Эта команда даёт ту же информацию, как и *show cdp neighbor detail*, но для одного конкретного устройства. Помните, что имена хостов чувствительны к регистру.

5. На устройстве *router1* введите команду, чтобы увидеть, как часто *router1* посылает соседям обновления *CDP* и как долго у соседей они должны храниться.

```
router1#show cdp
```

```
Global CDP information:
```

```
  Sending CDP packets every 60 seconds
```

```
  Sending a holdtime value of 180 seconds
```

```
  Sending CDPv2 advertisements is enabled
```

6. На устройстве *router1* введите команды для установки времени обновления *CDP* 45 секунд и для установки времени сохранения *CDP* 60секунд.

```
router1 (config)#cdp timer 45
```

```
router1 (config)#cdp holdtime 60
```

```
router1 (config)#Ctrl-Z
```

```
router1#show cdp
```

Для экономии полосы пропускания низкоскоростных устройств *CDP* можно отключить

```
router1 (config)#no cdp run
```

и снова включить для всего устройства *router1 (config)#cdp run*

7. Иногда необходимо отключить *CDP* для определённого интерфейса, например при его узкой полосе пропускания или в целях безопасности. На устройстве *router1*, отключите *CDP* на интерфейсе *Ethernet 0*.

```
router1 (config)#interface Ethernet 0
```

```
router1 (config-if)#no cdp enable
```

```
router1 (config)#Ctrl-Z
```

```
router1#show cdp interface
```

В полученном выводе вы не увидите сведений об *Ethernet 0*.

### Настройка IP адресов интерфейсов

1. Подключимся к устройству *Router1* и установим *IP* адрес *Ethernet* интерфейса

```
Router1(config)#interface ethernet 0
```

```
Router1( config-if)#ip address 10.1.1.1 255.255.255.0
```

2. Теперь назначим интерфейсу *S0* *IP* адрес *172.16.10.1 255.255.255.0*, не выходя из конфигурации интерфейса

```
Router1( config-if)#in s0
```

```
Router1( config-if)#ip ad 172.16.10.1 255.255.255.0
```

Отметим, что на последовательное соединение точка-точка всегда выделяется целая подсеть.

3. Переключимся к устройству *Router2* и назначим интерфейсу *Ethernet 0*

*IP* адрес *10.1.1.2 255.255.255.0*

```
Router2(config)#interface Ethernet 0
```

```
Router2( config-if)#ip address 10.1.1.2 255.255.255.0
```

4. Подключимся к устройству *Router4* и установим *IP* адрес *Ethernet* интерфейса *Serial 0*.

```
Router4( config-if)#ip address 172.16.10.2 255.255.255.0
```

5. На каждом устройстве посмотрите вашу активную конфигурацию и убедитесь, что там появились назначенные *IP* адреса.

```
Router1#show running-config
```

```
Router2#show running-config
```

```
Router3#show running-config
```

6. Посмотрите детальную *IP* информацию о каждом интерфейсе и убедитесь, что отконфигурованные интерфейсы перешли в состояние *UP*

```
Router1#show ip interface
```

```
Router2#show ip interface
```

```
Router4#show ip interface
```

7. Краткую информацию можно получить командой *show ip interface brief*, например

```
Router1#show ip in bri
```

Interface	IP-Address	OK?	Method	Status	Protocol
Serial0	172.16.10.1	YES	unset	up	up
Ethernet0	10.1.1.1	YES	unset	up	up

```
Router2#show ip in bri
```

Interface	IP-Address	OK?	Method	Status	Protocol
Serial0	unassigned	YES	unset	administratively down	down
Ethernet0	10.1.1.2	YES	unset	up	up

```
Router4#show ip in bri
```

Interface	IP-Address	OK?	Method	Status	Protocol
Serial0	172.16.10.2	YES	unset	up	up
Ethernet0	unassigned	YES	unset	administratively down	down

8. Подключимся к устройству *Router1*. Вы должны успешно пропинговать непосредственно подсоединённый *Ethernet 0* интерфейс на устройстве *Router2*.

```
Router1#ping 10.1.1.2
```

```
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
```

```
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

Попробуем пропинговать интерфейс *Serial 0* на устройстве *Router4*.

```
Router1#ping 172.16.10.2
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 172.16.10.2, timeout is 2 seconds:
```

```
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

Успешно.

9. Вернёмся на *Router2*. Вы должны успешно пропинговать адрес *10.1.1.1* непосредственно подсоединённого *Ethernet 0* интерфейса на устройстве *Router1*. Вернёмся на *Router4*. Вы должны успешно пропинговать адрес *172.16.10.1* непосредственно подсоединённого интерфейса *Serial 0* на устройстве *Router1*. Попробуем пропинговать интерфейс *Ethernet 0* на устройстве *Router1*.

```
Router4#ping 10.1.1.1
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
```

```
.....
```

```
Success rate is 0 percent (0/5), round-trip min/avg/max = 1/2/4 ms
```

Неудача.

Попробуем пропинговать адрес *10.1.1.2 ethernet 0* интерфейса на устройстве *Router2*. Неудача.

10. Вернёмся на *Router2*. Попробуем пропинговать адрес *172.16.10.1* интерфейса *Serial 0* на устройстве *Router1*. Неудача.

Попробуем пропинговать адрес *172.16.10.2* интерфейса *Serial 0* на устройстве *Router4*. Неудача.

Неудачи наступили потому, что мы не настроили на маршрутизаторах маршрутизацию.

11. Зайдите на устройстве *Router1*. Определите пути прохождения пакетов на *Router2* *Router1#traceroute 10.1.1.2* и *Router4*

```
Router1# traceroute 172.16.10.2
```

Вы должны увидеть по одному хопу.

12. Выполните команду расширенного пинга от адреса *10.1.1.2* к адресу *172.16.10.2*

```
Router1#ping
```

```
...
```

```
Target IP address: 172.16.10.2
```

```
...
```

```
Extended commands [n]: y
```

```
Source address: 10.1.1.2
```

### Telnet

Будьте внимательны: симулятор имеет ограниченную поддержку *telnet*.

1. Войдите на устройство *Router1*. Нам необходимо, чтобы сетевое устройство принимало *telnet*-сессии и было защищено паролем. Каждая так называемая линия в сетевом устройстве потенциально представляет активную *telnet*-сессию, которую устройство может поддерживать. Наши сетевые устройства поддерживают до 5 линий, назначенные на виртуальные терминалы *vtu*. Мы используем все 5 линий

```
Router1(config)#line vty 0 4
```

```
Router1( config-line)#
```

2. Теперь сообщим сетевому устройству, что нам понадобится пароль входу в систему.

```
Router1( config-line)#login
```

```
Router1( config-line)#password boson
```

3. Войдите на устройство *Router2* и установим *telnet*-соединение с устройством *Router1*. Для этого мы используем *IP* адресу его интерфейсу

```
Ethernet 0
```

```
Router2#telnet 10.1.1.1
```

4. Мы увидим просьбу ввести пароль. Введите пароль *boson* и нажмите *<enter>*. Заметьте, что имя сетевого устройства поменялось на "*Router1*", потому, что мы установили *telnet*-соединение с *Router1*.

Команда *Router1>show user*

```
* 1 vty 1
```

```
idle
```

```
10.1.1.2
```

покажет, что соединение осуществлено от адреса *10.1.1.2* устройства *router2*.

5. Теперь на секунду нажмите одновременно клавиши *control-shift-6*, потом отпустите и сразу нажмите клавишу *x*. Заметьте, что имя сетевого устройства поменялось назад на "*Router2*". Теперь вы опять устройстве *Router2*.

```
Router1><Control> + <Shift> + <6> nomom <x>
```

```
Router2#
```

6. Введите команду *show sessions*. Это позволит вам увидеть все активные *telnet*-сессии. Чтобы возобновить *telnet*-сессию, определите номер сессии, которую вы хотите возобновить (в нашем случае есть только одна с номером 1) и введите команду *resume 1*.

```
Router2#Show sessions
```

Conn	Host	Address	Byte	Idle	Conn Name
*	1 10.1.1.1	10.1.1.1		0	9 10.1.1.1

Router2#resume 1

Router1#

7. Теперь имя хоста снова поменялось на *Router1*. Нажмите комбинацию *control-shift-6* и клавишу *x*, чтобы вернуться назад на *Router2*.

Router1#<Control> + <Shift> + <6> *nomom* <x>

Router2#

8. Закройте сессию

Router2#disconnect 1

**Closing connection to 10.1.1.1**

Router1#disconnect 1

**Closing connection to 10.1.1.1**

Сохраните проект в целом и конфигурацию каждого устройства в отдельности.

### Контрольные вопросы.

1. Какие есть режимы ввода команд в командной строке?
2. Как переключаться между режимами ввода команд в командной строке?
3. Какую роль играет клавиша табуляции при вводе команд?
4. Как войти в режимы глобальной конфигурации, активизировать частный вид конфигурации и выйти из этих режимов?
5. Как ориентироваться в ранее введенных командах и повторять их?
6. Что такое *CDP*, для чего он служит и как им пользоваться?
7. Какую информацию возвращает команда *ping*?
8. Можно ли находясь на одном устройстве попарно пропинговать все устройства в сети?
9. Для чего служит команда *traceroute*?
10. Для чего служит команда протокол *telnet*?
11. Как задать имя хоста?
12. Какую информацию можно посмотреть командами *show* в пользовательском режиме?
13. Какую информацию можно посмотреть командами *show* в привилегированном режиме, но нельзя посмотреть в пользовательском режиме?
14. Каким устройством может выступать маршрутизатор для последовательной линии связи?
15. На каком устройстве при последовательном соединении можно устанавливать частоту синхронизации?
16. Как поднять интерфейс и определить его состояние?
17. Как назначить IP адрес на интерфейс и убедиться, что он назначен?
18. Почему могут не проходить пинги между устройствами?
19. Как приостановить и возобновить *telnet*-сессию?
20. Как закрыть *telnet* соединение?

### Порядок выполнения и сдачи работы

1. Изучить теоретическую и практическую часть.
2. Сдать преподавателю теорию работы путём ответа на контрольные вопросы.
3. Выполнить в *Boson* практическую часть.
4. Получите вариант и выполните в *Boson* задание для самостоятельной работы
5. Предъявите преподавателю результат выполнения задания для самостоятельной работы. Покажите и объясните ему свои *rtr*, *nwc* и *top* файлы. Продемонстрируйте ему, что компьютеры пингуются согласно таблице 5.

6. Продемонстрируйте работу *telnet*.
7. Оформите отчёт. Содержание отчёта смотри ниже.
8. Защитите отчёт.

### Задание для самостоятельной работы.

1. Получите свой вариант

Таблица 5.

Вариант	i11-i31	i12-i21	i22-i32
1, 9	serial	Serial	serial
2, 10	serial	Serial	ethernet
3, 11	serial	Ethernet	serial
4, 12	serial	Ethernet	ethernet
5, 13	ethernet	Serial	serial
6, 14	ethernet	Serial	ethernet
7, 15	ethernet	Ethernet	serial
8, 16	ethernet	Ethernet	ethernet

Выберите в дизайнера подходящие устройства и создайте топологию, изображённую на рис. 22. Сами назначьте устройствам имена. Поднимите на каждом устройстве используемые интерфейсы. Проверьте их состояния. На каждом устройстве, используя команды *CDP show cdp neighbors*, получите информацию о соседних устройствах. Сохраните скриншоты команд CDP.

2. Назначьте интерфейсам адреса, согласно варианту (v=1-16) из таблицы 7.
3. Все маски равны 255.255.255.0. Например, для варианта 7 (v=7) имеем адреса из таблицы 6.

Таблица 6.

Устройство	Интерфейс	Адрес
Router1	i11	7.1.1.2
Router3	i31	7.1.1.2
Router1	i12	7.1.2.1
Router2	i21	7.1.2.2
Router2	i22	7.1.3.1
Router3	i32	7.1.3.2

Таблица 7.

Вариант	i11, i31	i12, i21	i22, i32
1	v.1.1.1, v.1.1.2	v.1.2.1, v.1.2.2	v.1.3.1, v.1.3.2
2	v.1.1.1, v.1.1.2	v.1.2.1, v.1.2.2	v.1.3.1, v.1.3.2
3	v.1.1.1, v.1.1.2	v.1.2.1, v.1.2.2	v.1.3.1, v.1.3.2
4	v.1.1.1, v.1.1.2	v.1.2.1, v.1.2.2	v.1.3.1, v.1.3.2
5	v.1.1.1, v.1.1.2	v.1.2.1, v.1.2.2	v.1.3.1, v.1.3.2
6	v.1.1.1, v.1.1.2	v.1.2.1, v.1.2.2	v.1.3.1, v.1.3.2

7	v.1.1.1, v.1.1.2	v.1.2.1, v.1.2.2	v.1.3.1, v.1.3.2
8	v.1.1.1, v.1.1.2	v.1.2.1, v.1.2.2	v.1.3.1, v.1.3.2
9	v.1.1.1, v.1.1.2	v.1.2.1, v.1.2.2	v.1.2.1, v.1.2.2
10	v.1.1.1, v.1.1.2	v.1.2.1, v.1.2.2	v.1.3.1, v.1.3.2
11	v.1.1.1, v.1.1.2	v.1.2.1, v.1.2.2	v.1.3.1, v.1.3.2
12	v.1.1.1, v.1.1.2	v.1.2.1, v.1.2.2	v.1.3.1, v.1.3.2
13	v.1.1.1, v.1.1.2	v.1.2.1, v.1.2.2	v.1.3.1, v.1.3.2
14	v.1.1.1, v.1.1.2	v.1.2.1, v.1.2.2	v.1.3.1, v.1.3.2
15	v.1.1.1, v.1.1.2	v.1.2.1, v.1.2.2	v.1.3.1, v.1.3.2
16	v.1.1.1, v.1.1.2	v.1.2.1, v.1.2.2	v.1.3.1, v.1.3.2

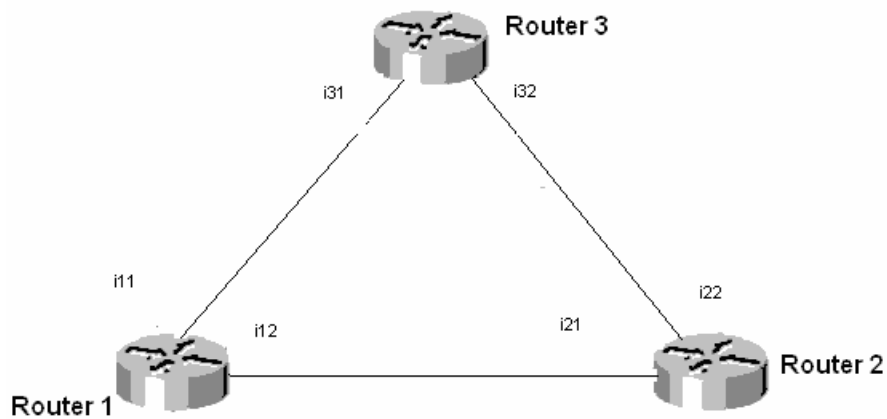


Рисунок 22. Топология сети.

4. Проверьте, что адреса назначены. На каждом устройстве выполните команду *show ip interface brief*. Сохраните скриншоты.

5. Если сделано всё правильно вы сможете пропинговать из любого компьютера определённые (но не все) адреса интерфейсов других компьютеров.

Таблица 8.

Из\На	I11	I12	I21	I22	I31	I32
Router1	Да	Да	Да *	Нет	Да *	Нет
Router2	Нет	Да	Да	Да	Нет	Да *
Router3	Да	Нет	Нет	Да	Да	Да

Сделайте это. Сохраните скриншоты для пингов соединений, отмеченных в таблице 8 знаком \*.

6. Выполните на Router1 расширенный пинг. Сохраните 3 скриншота для пингов: от *i12* к *i21*, от *i11* к *i31* и от *i22* к *i32*.

7. Настройте на Router1 Telnet. Задайте пароль.

8. Перейдите на Router2. Зайдите по Telnet на Router1. Выполните команду *show user*. Приостановите сессию. Возобновите сессию. Убейте сессии. Сохраните скриншот консоли Router2.

9. Сохраните топологию и конфигурацию.

## Лабораторная работа №17 Статическая маршрутизация.

**Тема работы:** статическая маршрутизация.

**Цель работы:** маршрутизация, понятие статического маршрута, настройка маршрутизации, таблица маршрутизации.

**Теоретическая часть:**

### ARP (Address Resolution Protocol).

Когда отправитель определил *IP* адрес приёмника, он смотрит в свою *ARP* таблицу чтобы узнать *MAC* адрес приёмника. Если источник обнаруживает, что *MAC* и *IP* адреса приёмника присутствуют в *ARP* таблице, он устанавливает между ними соответствие и использует его в ходе инкапсуляции *IP* пакетов во фреймы канального уровня. *MAC* адреса фреймов канального уровня берутся из *ARP* таблиц. После этого фрейм по физическому каналу отправляется от отправителя к адресату.

Если отправитель имеет *IP* пакет для получателя с *IP*-адресом АДР и этот адрес отсутствует в *ARP* таблице, то отправитель отправляет по сети широковещательный *ARP* запрос следующего содержания: сообщите *MAC* адрес сетевого интерфейса с *IP*-адресом АДР. Запрос принимают все сетевые устройства в сегменте сети, и только устройство, имеющее *IP*-адрес АДР, реагирует на него, посылая отправителю информацию о *MAC* адресе своего сетевого интерфейса с *IP* адресом АДР. Отправитель записывает пару <*MAC* адрес, *IP*-адрес АДР > в свою *ARP* таблицу.

### Маршрутизация.

Протоколы маршрутизации - это правила, по которым осуществляется обмен информации о путях передачи пакетов между маршрутизаторами. Протоколы характеризуются временем сходимости, потерями и масштабируемостью. В настоящее время используется несколько протоколов маршрутизации. Каждый протокол имеет сильные и слабые стороны.

Одна из главных задач маршрутизатора состоит в определении наилучшего пути к заданному адресату. Маршрутизатор определяет пути (маршруты) к адресатам или из статической конфигурации, введённой администратором, или динамически на основании маршрутной информации, полученной от других маршрутизаторов. Маршрутизаторы обмениваются маршрутной информацией с помощью протоколов маршрутизации. Маршрутизатор хранит таблицы маршрутов в оперативной памяти. Таблица маршрутов это список наилучших известных доступных маршрутов. Маршрутизатор использует эту таблицу для принятия решения куда направлять пакет. Для просмотра таблицы маршрутов следует использовать команду *show ip route*. Даже, если на некотором маршрутизаторе *X* не задавались никакие команды маршрутизации, тогда он всё равно строит таблицу маршрутов для непосредственно подсоединённых к нему сетей, например:

```
C    192.168.4.0/24 is directly connected, Ethernet0
10.0.0.0/16 is subnetted, 3 subnets
C    10.3.0.0 is directly connected, Serial0
C    10.4.0.0 is directly connected, Serial1
C    10.5.0.0 is directly connected, Ethernet1
```

Маршрут на непосредственно подсоединённые сети отображается на интерфейс маршрутизатора, к которому они присоединены. Здесь /24 обозначает маску 255.255.255.0, а /16 - 255.255.0.0.

Таблица маршрутов отображает сетевые префиксы (адреса сетей) на выходные интерфейсы. Когда *X* получает пакет, предназначенный для 192.168.4.46, он ищет префикс 192.168.4.0/24 в таблице маршрутов. Согласно таблице пакет будет направлен на интерфейс *Ethernet0*. Если *X* получит пакет для 10.3.21.5, он направит его на *Serial0*.

Эта таблица показывает четыре маршрута для непосредственно подсоединённых сетей. Они имеют метку *C*. Маршрутизатор *X* отбрасывает все пакеты, направляемые к



сетям, не указанным в таблице маршрутов. Для направления пакетам к другим адресатам необходимо в таблицу включить дополнительные маршруты. Новые маршруты могут быть добавлены двумя методами:

**Статическая маршрутизация** – администратор вручную определяет маршруты к сетям назначения.

**Динамическая маршрутизация** – маршрутизаторы следуют правилам, определяемым протоколами маршрутизации для обмена информацией о маршрутах и выбора лучшего пути. Статические маршруты не меняются самим маршрутизатором. Динамические маршруты изменяются самим маршрутизатором автоматически при получении информации о смене маршрутов от соседних маршрутизаторов. Статическая маршрутизация потребляет мало вычислительных ресурсов и полезна в сетях, которые не имеют нескольких путей к адресату назначения. Если от маршрутизатора к маршрутизатору есть только один путь, то часто используют статическую маршрутизацию.

Для конфигурации статической маршрутизации в маршрутизаторах Cisco используют две версии команды `ip route`

*Первая версия*

`ip route АдресСетиНазначения МаскаСетиНазначения Интерфейс`

Команда указывает маршрутизатору, что все пакеты, предназначенные для АдресСетиНазначения-МаскаСетиНазначения следует направлять на свой интерфейс Интерфейс. Если интерфейс Интерфейс - типа Ethernet, то физические (MAC) адреса исходящих пакетов будут ширококестельными.

*Вторая версия*

`ip route АдресСетиНазначения МаскаСетиНазначения Адрес`

Команда указывает маршрутизатору, что все пакеты, предназначенные для АдресСетиНазначения-МаскаСетиНазначения, следует направлять на тот свой интерфейс, из которого достижим IP адрес Адрес. Как правило, Адрес это адрес следующего хопа по пути к АдресСетиНазначения. Выходной интерфейс и физические адреса исходящих пакетов определяются маршрутизатором по своим ARP таблицам на основании IP адреса Адрес.

Например

`ip route 10.6.0.0 255.255.0.0 Serial1 (1)`

`ip route 10.7.0.0 255.255.0.0 10.4.0.2 (2)`

Первый пример отображает сетевой префикс 10.6.0.0/16 на локальный интерфейс маршрутизатора Serial1. Следующий пример отображает сетевой префикс 10.7.0.0/16 на IP адрес 10.4.0.2 следующего хопа по пути к 10.7.0.0/16. Обе эти команды добавят статические маршруты в таблицу маршрутизации (метка S):

`S 10.6.0.0 via Serial1`

`S 10.7.0.0 [1/0] via 10.4.0.2`

Когда интерфейс падает, все статические маршруты, отображаемые на этот интерфейс, удаляются из таблицы маршрутов. Если маршрутизатор не может больше найти адрес следующего хопа по пути к адресу, указанному в статическом маршруте, то маршрут исключается из таблицы.

Заметим, что для сетей типа *Ethernet* рекомендуется всегда использовать форму (2) команды `ip route`. *Ethernet* интерфейс на маршрутизаторе, как правило, соединён с несколькими *Ethernet* интерфейсами других устройств в сети. Указание в команде `ip route` IP адреса позволит маршрутизатору правильно сформировать физический адрес выходного пакета по своим ARP таблицам.

#### Маршрутизация по умолчанию.

Совсем не обязательно, чтобы каждый маршрутизатор обслуживал маршруты ко всем возможным сетям назначения. Вместо этого маршрутизатор хранит маршрут по умолчанию или шлюз последнего пристанища (*last resort*). Маршруты по умолчанию ис-

пользуются, когда маршрутизатор не может поставить в соответствие сети назначения строку в таблице маршрутов. Маршрутизатор должен использовать маршрут по умолчанию для отсылки пакетов другому маршрутизатору. Следующий маршрутизатор будет иметь маршрут к этой сети назначения или иметь свой маршрут по умолчанию к третьему маршрутизатору и т.д. В конечном счёте, пакет будет маршрутизирован на маршрутизатор, имеющий маршрут к сети назначения.

Маршрут по умолчанию может быть статически введен администратором или динамически получен из протокола маршрутизации.

Так как все IP адреса принадлежат сети 0.0.0.0 с маской 0.0.0.0, то в простейшем случае надо использовать команду

```
ip route 0.0.0.0 0.0.0.0 [адрес следующего хоста | выходной интерфейс]
```

Ручное задание маршрута по умолчанию на каждом маршрутизаторе подходит для простых сетей. В сложных сетях необходимо организовать динамический обмен маршрутами по умолчанию.

### Интерфейс петля.

На сетевых устройствах можно создавать сетевые интерфейсы не связанные с реальными каналами для передачи данных и назначать на них IP адреса с масками. Такие интерфейсы называют петлями (*loopback*). Петли полезны при поэтапном проектировании сетей. Если к какому-то реальному сетевому интерфейсу маршрутизатора в дальнейшем будет подсоединена 32 подсеть, то в начале на маршрутизаторе создаётся *loopback*, настраивается в плане взаимодействия с остальными участками сети и лишь затем заменяется на реальный интерфейс. Интерфейс петля появляется после команды *interface loopbackN* или сокращённо *int IN*, где *N* целое неотрицательное число – номер петли. Например

```
Router(conf)>int l0 1.1.1.1 255.0.0.0
```

### Команда trace.

Команда *trace* является идеальным способом для выяснения того, куда отправляются данные в сети. Эта команда использует ту же технологию протокола *ICMP*, что и команда *ping*, только вместо проверки сквозной связи между отправителем и получателем, она проверяет каждый шаг на пути. Команда *trace* использует способность маршрутизаторов генерировать сообщения об ошибке при превышении пакетом своего установленного времени жизни (Time To Live, TTL). Эта команда посылает несколько пакетов и выводит на экран данные про время прохождения туда и назад для каждого из них. Преимущество команды *trace* заключается в том, что она показывает очередной достигнутый маршрутизатор на пути к пункту назначения. Это очень мощное средство для локализации отказов на пути от отправителя к получателю. Варианты ответов утилиты *trace* представлены в таблице 9.

Таблица 9.

Символ	Значение
!H	Зондирующий пакет был принят маршрутизатором, но не переадресован, что обычно бывает из-за списка доступа
P	Протокол недостижим
N	Сеть недостижима
U	Порт недостижим
*	Превышение границы ожидания

### **Практическая часть.**

Загрузите топологию и конфигурацию из практической части предыдущей работы.  
**ARP**

1. Присоединитесь к маршрутизатору *Router1* с и посмотрите его *ARP* таблицу

```
Router1#show arp
```

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	10.1.1.1	-	000C.3997.1200	ARPA	Ethernet0

Она содержит только одну строку о *MAC* адресе своего *Ethernet* интерфейса с *IP* адресом *10.1.1.1*.

2. Присоединитесь к маршрутизатору *Router2* и посмотрите его *ARP* таблицу. Она содержит только одну строку о *MAC* адресе своего *Ethernet* интерфейса с *IP* адресом *10.1.1.2*.

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	10.1.1.2	-	000C.4361.1005	ARPA	Ethernet0

3. Пропингуйте *Ethernet* интерфейс маршрутизатора *Router1*

```
Router2#ping 10.1.1.1.
```

4. Снова посмотрите вашу *ARP* таблицу. Она содержит уже две строки. Появилась запись о *MAC* адресе *Ethernet* интерфейса *Router1* с *IP* адресом *10.1.1.1*.

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	10.1.1.2	-	000C.4361.1005	ARPA	Ethernet0
Internet	10.1.1.1	1	000C.3997.1200	ARPA	Ethernet0

5. Присоединитесь к маршрутизатору *router1* и посмотрите его *ARP* таблицу. Она содержит уже две строки

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	10.1.1.1	-	000C.3997.1200	ARPA	Ethernet0
Internet	10.1.1.2	1	000C.4361.1005	ARPA	Ethernet0

Появилась запись о *MAC* адресе *Ethernet* интерфейса маршрутизатора *Router2* с *IP* адресом *10.1.1.2*. Почему, ведь мы не слали от *Router1* никаких *IP* пакетов? Потому, что *Router1* для ответа на пинг от *Router2* должен был знать о *MAC* адресе *Ethernet* интерфейса *Router2* с *IP* адресом *10.1.1.2*, и он сформировал *ARP* пакет для его получения.

## Статические маршруты

В прошлой работе мы не могли из маршрутизаторов *Router2* и *Router4* пропинговать некоторые интерфейсы из-за отсутствия маршрутизации.

Исправим положение.

1. Присоединитесь к маршрутизатору *Router2*. Мы не могли пинговать адреса *172.16.10.1* и *172.16.10.2*. Посмотрите таблицу маршрутов

```
Router2# show ip route
```

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
U - per-user static route
```

```
Gateway of last resort is not set
```

```
10.0.0.0/24 is subnetted, 1 subnets
```

```
C 10.1.1.0 is directly connected, Ethernet0
```

Видим непосредственно присоединённые сети. Нет маршрута к сети *172.16.10.0/24*. Добавим маршрут к сети *172.16.10.0/24* через адрес *10.1.1.1* ближайшего хоста на пути к этой сети:

```
Router2(config)#ip route 172.16.10.0 255.255.255.0 10.1.1.1
```

Здесь и далее *172.16.10.0/24* – это сокращённая запись – определение подсети *172.16.10.0* с маской *255.255.255.0*. В маске *255.255.255.0* содержится *24* единицы, что и обозначается */24*.

2. Успешно пропингуем *serial* интерфейс *Router1*

```
Router2#ping 172.16.10.1
```

Снова посмотрите таблицу маршрутов

```

172.16.0.0/24 is subnetted, 1 subnets
S    172.16.10.0 [1/0] via 10.1.1.1
10.0.0.0/24 is subnetted, 1 subnets
C    10.1.1.0 is directly connected, Ethernet0

```

3. Но нам не удастся пропинговать *serial* интерфейс *Router4*.

*Router2#ping 172.16.10.2*

Почему? Потому, что *ICMP* пакеты пингов не знают, как им вернуться обратно от *Router4*, так как на *Router4* не прописаны маршруты.

4. Присоединитесь к маршрутизатору *Router4*. Посмотрите таблицу маршрутов

```

Router4# show ip route
172.16.0.0/24 is subnetted, 1 subnets
C    172.16.10.0 is directly connected, Serial0

```

Нет маршрута к сети *10.1.1.0/24*. Добавим маршрут к сети *10.1.1.0/24* через адрес *172.16.10.1* ближайшего хоста на пути к этой сети:

*Router4(config)#ip route 10.1.1.0 255.255.255.0 172.16.10.1*

Снова посмотрите таблицу маршрутов.

```

10.0.0.0/24 is subnetted, 1 subnets
S    10.1.1.0 [1/0] via 172.16.10.1
172.16.0.0/24 is subnetted, 1 subnets
C    172.16.10.0 is directly connected, Serial0

```

5. Теперь все сетевые интерфейсы в сети пингуются из каждого сетевого устройства. Проверьте это.

### Маршрутизация по умолчанию.

Сетевые устройства *Router2* и *Router4* имеют только по одному выход во внешний мир: через интерфейсы с адресами *10.1.1.1* и *172.16.10.1*, соответственно. Поэтому, можно не определять на какие подсети мы маршрутизируем пакеты и использовать маршрутизацию по умолчанию.

1. Вначале удалим старые маршруты.

*Router2(config)#no ip route 172.16.10.0 255.255.255.0 10.1.1.1*

*Router4(config)#no ip route 10.1.1.0 255.255.255.0 172.16.10.1*

2. И назначим маршруты по умолчанию.

*Router2(config)#ip route 0.0.0.0 0.0.0.0 10.1.1.1*

*Router4(config)#ip route 0.0.0.0 0.0.0.0 172.16.10.1*

3. Посмотрите таблицу маршрутов на всех устройствах.

*Router2#sh ip route*

```

Gateway of last resort is to network 0.0.0.0

S*   0.0.0.0 [1/0] via 10.1.1.1
     10.0.0.0/24 is subnetted, 1 subnets
C    10.1.1.0 is directly connected, Ethernet0

```

*Router4#sh ip route*

```

Gateway of last resort is to network 0.0.0.0

S*   0.0.0.0 [1/0] via 172.16.10.1
     172.16.0.0/24 is subnetted, 1 subnets
C    172.16.10.0 is directly connected, Serial0

```

4. Все сетевые интерфейсы в сети пингуются из каждого сетевого устройства. Проверьте это.

### Loopback

1. Определим интерфейс петлю на устройстве *Router4*

```
Router4(config)#int loopback 0
```

```
Router4(config-if)#ip address 1.1.1.1 255.255.255.0
```

2. Пропишем на устройстве *Router1* маршрут на сеть петли

```
Router1(conf)# ip route 1.1.1.0 255.255.255.0 172.16.10.2
```

3. Присоединимся к устройству *Router2* и пропингуем созданную петлю

```
Router2#ping 1.1.1.1
```

Сохраните проект в целом и конфигурацию каждого устройства в отдельности.

### Контрольные вопросы.

1. Как отправитель узнаёт *MAC* адрес получателя?
2. Как посмотреть *ARP* таблицу?
3. Когда в *ARP* таблице появляются новые строки?
4. Что такое таблица маршрутов?
5. Если администратор не настраивал никаких маршрутов, то что она будет содержать?
6. Чем статическая маршрутизация отличается от динамической?
7. Какие две формы задания статической маршрутизации вы знаете?
8. Как в команде маршрутизации определяется сеть назначения?
9. Объясните значения полей в командах маршрутизации.
10. Почему в качестве поля Адрес рекомендуют использовать адрес следующего хопа по пути к сети назначения.
11. Когда используется маршрутизация по умолчанию?
12. Когда используют интерфейс петля?
13. Как работает команда трасировки?

### Задание для самостоятельной работы.

1. Построить в *Boson Network Designer* топологию, представленную на рисунке 23. Использовать маршрутизаторы модели 2501.

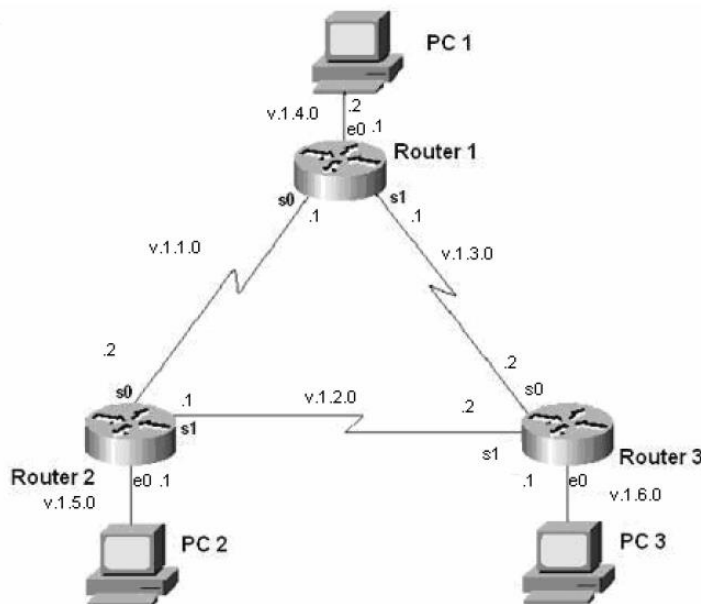


Рисунок 23. Топология сети.

В нашей сети шесть подсетей. Вы видите, что каждый маршрутизатор подключён к трём подсетям.

2. На каждом маршрутизаторе поднять используемые интерфейсы и посмотреть соседей командой *show cdp neighbors*. Сделать скриншоты.

3. Назначить интерфейсам сети адреса согласно рисунку 23 и таблице 10, в которых *v* – это номер варианта. Все маски *255.255.255.0*. Не забудьте назначить шлюзы по умолчанию для компьютеров согласно таблице 1.

Таблица 10.

	v.1.1.0	v.1.2.0	v.1.3.0	v.1.4.0	v.1.5.0	v.1.6.0
Router1	S0:v.1.1.1		S1:v.1.3.1	E0:v.1.4.1		
Router2	S0:v.1.1.2	S1:v.1.2.1			E0:v.1.5.1	
Router3		S0:v.1.2.2	S1:v.1.3.2			E0:v.1.6.1
PC1				E0:v.1.4.2		
PC2					E0:v.1.5.2	
PC3						E0:v.1.6.2

4. Проверьте факт назначения адресов путём выполнения на каждом маршрутизаторе команд *show running-config* и *show ip interface brief*. Для компьютеров используйте команду *ipconfig*.

5. Проверьте правильность назначения адресов путём выполнения на каждом маршрутизаторе команд *ping* к непосредственным соседям. Например, на маршрутизаторе *Router1* выполните

```
Router1#ping v.1.1.2
```

```
Router1#ping v.1.3.2
```

```
Router1#ping v.1.4.2
```

6. Поставим перед собой задачу связать между собой компьютеры *PC1*, *PC2* и *PC3*. Для этого осуществим на маршрутизаторах настройку статической маршрутизации. В каждом маршрутизаторе пропишем маршруты на удалённые *Ethernet* сети. Для решения поставленной задачи маршрутизировать пакеты на удалённые сети последовательных соединений не надо. У каждого маршрутизатора есть по две на удалённые *Ethernet* сети. Всего надо прописать шесть статических маршрутов.

Чтобы из маршрутизатора *router1* достичь удалённую *Ethernet* сеть *v.1.5.0/24*, пакеты можно направить на IP адрес *v.1.1.2* ближайшего внешнего интерфейса на пути в эту сеть. Это делает команда

```
router1(config)#ip route v.1.5.0 255.255.255.0 v.1.1.2
```

Задайте остальные пять команд маршрутизации.

7. На каждом маршрутизаторе посмотреть таблицу маршрутизации командой *show ip route*. Сделать скриншоты.

8. На каждом маршрутизаторе сделайте скриншоты расширенных пингов

а) на маршрутизаторе *router1* от *PC2* к *PC3*

б) на маршрутизаторе *router2* от *PC1* к *PC3*

в) на маршрутизаторе *router3* от *PC1* к *PC2*

Например, результат расширенного пинга на маршрутизаторе *router1* от *PC2* к *PC3* для для варианта 1 ( $v=1$ ) имеет вид

```
router1#ping
Protocol [ip]:
Target IP address: 1.1.6.2
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]:y
Source address or interface:1.1.5.2
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose [none]:
Sweep range of sizes [n]:

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.6.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

9. На каждом компьютере сделайте о скриншоты выполнения команд трасировки *tracert* других компьютеров. Всего шесть скриншотов. Например, трасировка из *PC1* на *PC2* для варианта 1 ( $v=1$ )

```
PC1:#tracert 1.1.5.2
```

```
"Type escape sequence to abort."
```

```
Tracing the route to 1.1.5.2
```

```
 1 1.1.4.1 0 msec 16 msec 0 msec  
 2 1.1.1.2 20 msec 16 msec 16 msec  
 3 1.1.5.2 20 msec 16 msec *
```

10. Сохраните проект.

## Лабораторная работа №18 Динамическая маршрутизация.

**Тема работы:** динамическая маршрутизация.

**Цель работы:** понятие динамической маршрутизации, конфигурирование динамической маршрутизации, использование протоколов *RIP*, *IGRP*, *OSPF*.

**Теоретическая часть:**

Статическая маршрутизация не подходит для больших, сложных сетей потому, что обычно сети включают избыточные связи, многие протоколы и смешанные топологии. Маршрутизаторы в сложных сетях должны быстро адаптироваться к изменениям топологии и выбирать лучший маршрут из многих кандидатов.

*IP* сети имеют иерархическую структуру. С точки зрения маршрутизации сеть рассматривается как совокупность автономных систем. В автономных подсистемах больших сетей для маршрутизации на остальные автономные системы широко используются маршруты по умолчанию.

Динамическая маршрутизация может быть осуществлена с использованием одного и более протоколов. Эти протоколы часто группируются согласно того, где они используются. Протоколы для работы внутри автономных систем называют внутренними протоколами шлюзов (*interior gateway protocols (IGP)*), а протоколы для работы между автономными системами называют внешними протоколами шлюзов (*exterior gateway protocols (EGP)*). К протоколам *IGP* относятся *RIP*, *RIP v2*, *IGRP*, *EIGRP*, *OSPF* и *IS-IS*. Протоколы *EGP3* и *BGP4* относятся к *EGP*. Все эти протоколы могут быть разделены на два класса: дистанционно-векторные протоколы и протоколы состояния связи.

Маршрутизаторы используют метрики для оценки или измерения маршрутов. Когда от маршрутизатора к сети назначения существует много маршрутов, и все они используют один протокол маршрутизации, то маршрут с наименьшей метрикой рассматривается как лучший. Если используются разные протоколы маршрутизации, то для выбора маршрута используются административные расстояния, которые назначаются маршрутам операционной системой маршрутизатора.

*RIP* использует в качестве метрики количество переходов (хопов). *EIGRP* использует сложную комбинацию факторов, включающую полосу пропускания канала и его надёжность.

Результаты работы маршрутизирующих протоколов заносятся в таблицу маршрутов, которая постоянно изменяется при смене ситуации в сети.

Рассмотрим типичную строку в таблице маршрутов, относящуюся к динамической маршрутизации *R 192.168.14.0/24 [120/3] via 10.3.0.1 00:00:06 Serial0*.

Здесь *R* определяет протокол маршрутизации. Так *R* означает *RIP*, а *O* – *OSPF* и т.д. Запись *[120/3]* означает, этот маршрут имеет административное расстояние 120 и метрику 3. Эти числа маршрутизатор использует для выбора маршрута. Элемент *00:00:06* определяет время, когда обновилась данная строка. *Serial0* это локальный интерфейс, через который маршрутизатор будет направлять пакеты к сети *192.168.14.0/24* через адрес *10.3.0.1*.

Для того чтобы динамические протоколы маршрутизации обменивались информацией о статических маршрутах, следует осуществлять дополнительное конфигурирование.

### Дистанционно-векторная маршрутизация

Эта маршрутизация базируется на алгоритме Белмана-Форда. Через определённые моменты времени маршрутизатор передаёт соседним маршрутизаторам всю свою таблицу маршрутизации. Такие простые протоколы как *RIP* и *IGRP* просто распространяют информацию о таблицах маршрутов через все интерфейсы маршрутизатора в широковещательном режиме без уточнения точного адреса конкретного соседнего маршрутизатора.

Соседний маршрутизатор, получая широковещание, сравнивает информацию со своей текущей таблицей маршрутов. В неё добавляются маршруты к новым сетям или маршруты к известным сетям с лучшей метрикой. Происходит удаление несуществующих



маршрутов. Маршрутизатор добавляет свои собственные значения к метрикам полученных маршрутов. Новая таблица маршрутизации снова распространяется по соседним маршрутизаторам (см. рис. 24).

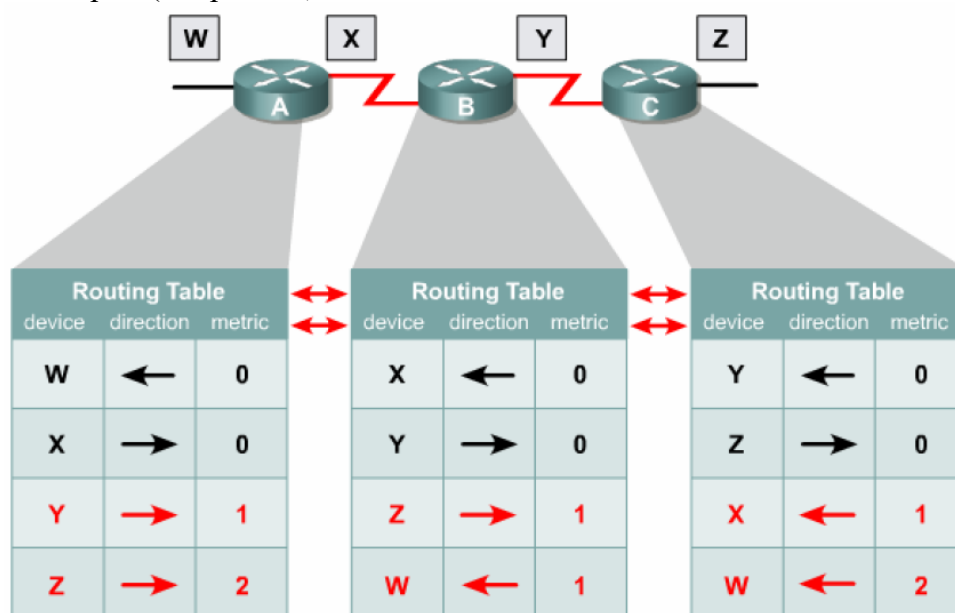


Рисунок 24. Дистанционно-векторная маршрутизация.

### Протоколы состояния связи

Эти протоколы предлагают лучшую масштабируемость и сходимость по сравнению с дистанционно-векторными протоколами. Протокол базируется на алгоритме Дейкстры, который часто называют алгоритмом «кратчайший путь – первым» (*shortest path first (SPF)*). Наиболее типичным представителем является протокол *OSPF (Open Shortest Path First)*.

Маршрутизатор берёт в рассмотрение состояние связи интерфейсов других маршрутизаторов в сети. Маршрутизатор строит полную базу данных всех состояний связи в своей области, то есть имеет достаточно информации для создания своего отображения сети. Каждый маршрутизатор затем самостоятельно выполняет SPF-алгоритм на своём собственном отображении сети или базе данных состояний связи для определения лучшего пути, который заносится в таблицу маршрутов. Эти пути к другим сетям формируют дерево с вершиной в виде локального маршрутизатора.

Маршрутизаторы извещают о состоянии своих связей всем маршрутизаторам в области. Такое извещение называют *LSA (link-state advertisements)*.

В отличие от дистанционно-векторных маршрутизаторов, маршрутизаторы состояния связи могут формировать специальные отношения со своими соседями.

Имеет место начальный наплыв *LSA* пакетов для построения базы данных состояний связи. Далее обновление маршрутов производится только при смене состояний связи или, если состояние не изменилось в течение определённого интервала времени. Если состояние связи изменилось, то частичное обновление пересылается немедленно. Оно содержит только состояния связей, которые изменились, а не всю таблицу маршрутов.

Администратор, заботящийся об использовании линий связи, находит эти частичные и редкие обновления эффективной альтернативой дистанционно-векторной маршрутизации, которая передаёт всю таблицу маршрутов через регулярные промежутки времени.

Протоколы состояния связи имеют более быструю сходимость и лучшее использование полосы пропускания по сравнению с дистанционно-векторными протоколами. Они превосходят дистанционно-векторные протоколы для сетей любых размеров, однако име-

ют два главных недостатка: повышенные требования к вычислительной мощности маршрутизаторов и сложное администрирование.

#### **Сходимость.**

Этот процесс одновременно и совместный и индивидуальный. Маршрутизаторы разделяют между собой информацию, но самостоятельно пересчитывают свои таблицы маршрутизации. Для того чтобы индивидуальные таблицы маршрутизации были точными, все маршрутизаторы должны иметь одинаковое представление о топологии сети. Если маршрутизаторы договорились о топологии сети, то имеет место их сходимость. Быстрая сходимость означает быстрое восстановление после обрыва связей и других изменений в сети. О протоколах маршрутизации и о качестве проектирования сети судят главным образом по сходимости.

Когда маршрутизаторы находятся в процессе сходимости, сеть восприимчива к проблемам маршрутизации. Если некоторые маршрутизаторы определили, что некоторая связь отсутствует, то другие ошибочно считают эту связь присутствующей. Если это случится, то отдельная таблица маршрутов будет противоречива, что может привести к отбрасыванию пакетов и петлям маршрутизации.

Невозможно, чтобы все маршрутизаторы в сети одновременно обнаружили изменения в топологии. В зависимости от использованного протокола, может пройти много времени пока все процессы маршрутизации в сети сойдутся. На это влияют следующие факторы:

- расстояние в хопах до точки изменения топологии;
- число маршрутизаторов, использующих динамические протоколы;
- полоса пропускания и загрузка каналов связи;
- загрузка маршрутизаторов;
- эффект некоторых факторов может быть уменьшен при тщательном проектировании сети.

#### **Конфигурирование динамической маршрутизации**

Для конфигурирования динамической маршрутизации используются две основные команды: *router i network*. Команда *router* запускает процесс маршрутизации и имеет форму:

```
Router(config)# router protocol [keyword],
```

где *Protocol* - любой из протоколов маршрутизации: *RIP, IGRP, OSPF* и т.п.,

*keyword* –дополнительные параметры.

Затем необходимы команды *network*:

```
Router ( config-router)# network network-number [keyword],
```

где *network-number* - идентифицирует непосредственно подключенную сеть, добавляемую в процесс маршрутизации, *keyword* –дополнительные параметры. *network-number* позволяет процессу маршрутизации определить интерфейсы, которые будут брать участие в отсылке и приёме пакетов актуализации маршрутной информации.

Для просмотра информации о протоколах маршрутизации используется команда *show ip protocol*, которая выводит значения таймеров процессов маршрутизации и сетевую информацию, имеющую отношение к маршрутизации. Эта информация может использоваться для идентификации маршрутизатора, подозреваемого в поставке плохой маршрутной информации.

Содержимое таблицы IP маршрутизации выводится командой *show ip route*. Она содержит записи про все известные маршрутизатору сети и подсети и указывает на способ получения этой информации.

#### **Протокол RIP**

Ключевые характеристики протокола RIP:

- маршрутизация на основании вектора расстояния;
- метрика при выборе пути в виде количества переходов (хопов);
- максимально допустимое количества хопов- 15;

- по умолчанию пакеты актуализации маршрутной информации посылаются в режиме широковещания каждые 30 секунд.

Выбор протокола RIP как протокола маршрутизации осуществляется командой:

*Router(config)# router rip*

Команда *network* назначает IP адрес сети с которой маршрутизатор имеет непосредственное соединение.

*Router (config-router)# network network-number*

Процесс маршрутизации связывает интерфейсы с соответствующими адресами и начинает обработку пакетов в заданных сетях.

В показанном на рис.25 примере команды *network 1.0.0.0* и *network 2.0.0.0* задают непосредственно подключенные к маршрутизатора Cisco A сети.

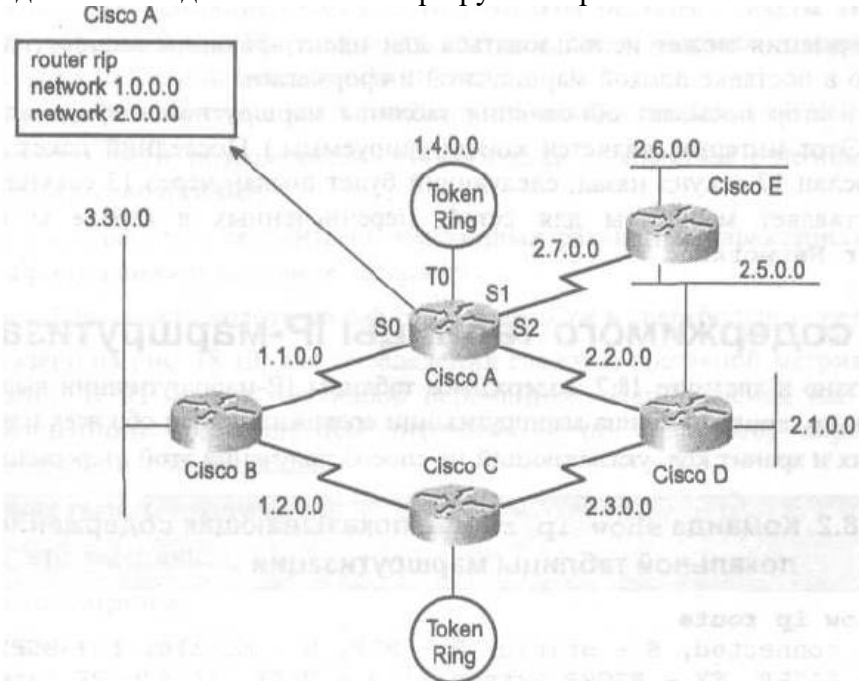


Рисунок 25. Пример команды *network*.

Команда *debug ip rip* выводит содержание пакетов актуализации маршрутной информации протокола *RIP* в том виде, в котором эти данные посылаются и принимаются.

### Протокол *IGRP*

*IGRP* представляет собой протокол маршрутизации по вектору расстояния разработанный компанией *Cisco*. Этот протокол посылает пакеты актуализации маршрутной информации с 90-секундным интервалом, в которых содержатся сведения о сетях для конкретной автономной системы. Этот протокол характеризует универсальность, позволяющий автоматически справляться со сложными топологиями и гибкость в работе с сегментами, имеющими разные характеристики по полосе пропускания и величины задержки. Используемая им метрика не имеет свойственных протоколу *RIP* ограничений по количеству переходов. Она включает следующие составляющие: ширина полосы пропускания; величина задержки; уровень загрузки; надёжность канала; размер максимального блока передачи в канале.

Выбор протокола *IGRP* в качестве протокола маршрутизации осуществляется с помощью команды:

*Router (config)# router igrp autonomous-system*

где параметр *Autonomous-system* называют номером автономной системы и он идентифицирует вычислительный процесс *IGRP*- маршрутизации. Процессы в маршрутизаторах сети с одинаковым номером *autonomous-system* будут коллективно использовать маршрутную информацию.

Команда *network* задаёт непосредственно присоединённые сети, которые подлежат включению в данный процесс маршрутизации:

***Router(config-router)#network network-number***

В показанном на рис.26 примере на маршрутизаторе *Cisco A* запущен маршрутизирующий процесс, организующий *IGRP* маршрутизацию в автономной системе с номером *109*. В маршрутизации будут участвовать сети *1.0.0.0* и *2.0.0.0*.

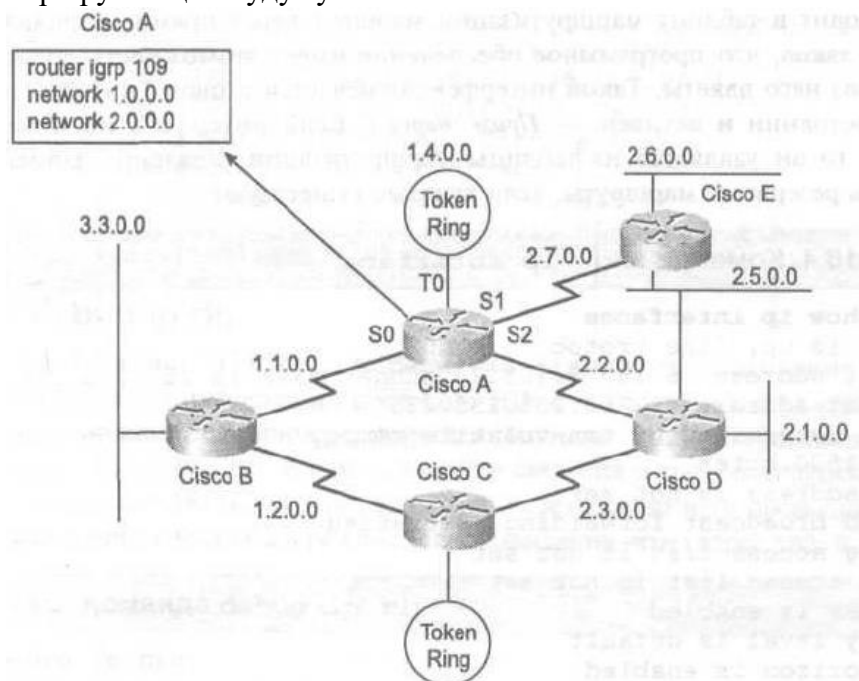


Рисунок 26. Процесс, организующий *IGRP* маршрутизацию.

Команда *debug ip igrp transactions* и *debug ip igrp events* выводят содержание пакетов актуализации маршрутной информации протокола *IGRP* в том виде, в котором эти данные посылаются и принимаются

### **Протокол *OSPF***

*OSPF* это динамический, иерархический протокол состояния связи, используемый для маршрутизации внутри автономных систем. Он базируется на открытых стандартах и был спроектирован как замена протоколу *RIP*. Он является развитием ранних версий протокола маршрутизации *IS-IS*. *OSPF* - устойчивый протокол, поддерживающий маршрутизацию с наименьшим весом и балансировку загрузки. Кратчайший путь в сети вычисляется по алгоритму Дейкстры. *Cisco* поддерживает свою версию стандарта *OSPF*.

Как только маршрутизатор настроен на работу с *OSPF*, он начинает процесс изучения окружения, проходя несколько фаз инициализации. В начале маршрутизатор использует *Hello* для определения своих соседей и создания отношений для обмена обновлением маршрутной информацией с ними. Затем маршрутизатор начинает фазу *ExStart* начального обмена между базами маршрутов. Следующей является фаза обмена, в которой назначенный маршрутизатор отправляет маршрутную информацию и получает подтверждения от нашего нового маршрутизатора. В течение стадии загрузки, новый маршрутизатор компилирует таблицу маршрутов. По окончании вычислений маршрутизатор переходит в полное состояние, в котором он является активным членом сети.

Для запуска *OSPF* маршрутизации служит команда *Router(config)#router ospf N*, где *N*-номер вычислительного процесса *OSPF*. В отличие от *IGRP* он может быть различным для разных маршрутизаторов автономной системы. *OSPF* область *Area* организуется командой

***Router(config-router)# network network-number area Area***

и определяет автономную систему.

В *OSPF network-number* имеет особый формат. Для подключаемой в процесс маршрутизации сети используется инверсная маска. Так, чтобы сеть *212.34.0.0 255.255.0.0* поместить в область *7 OSPF* маршрутизации следует дать команду

```
Router(config-router)# network 212.34.0.0 0.0.255.255 area 7
```

Команда `show ip ospf interface` для каждого интерфейса выводит всю *OSPF* информацию: *IP* адрес, область, номер процесса, идентификатор маршрутизатора, стоимость, приоритет, тип сети, интервалы таймера.

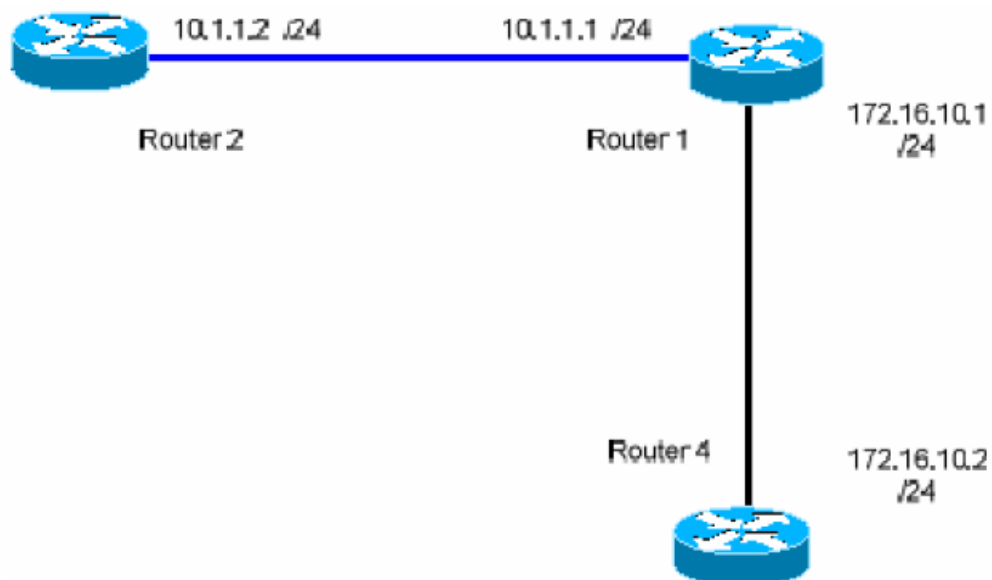
Команда `show ip ospf neighbor` показывает важную информацию, касающуюся состояния соседей.

### Практическая часть

1. Загрузите в симулятор топологию и конфигурацию, использованную в практической части лабораторной работы №16.

2. Если сделано всё правильно вы сможете пропинговать из любого маршрутизатора адреса непосредственно соединённых интерфейсов других маршрутизаторов. На каждом устройстве, используя команды *CDP* `show cdp neighbors detail`, получите *IP* адреса соседних устройств и пропингуйте их.

3. В лабораторной работе №16 мы не смогли пинговать дальние интерфейсы. Из *Router2* была недоступна сеть `172.16.10.0/24`, а из *Router4* была недоступна сеть `10.1.1.0/24`. В лабораторной работе №17 с помощью статической маршрутизации мы решили проблему. В этой работе для решения проблемы используем разные формы динамической маршрутизации.



4. Посмотрим таблицы маршрутов `Router2# sh ip route`

Нет маршрута на сеть `172.16.10.0/24`.

Поэтому в эту сеть из *Router2* не идут пинги.

```
Router4# sh ip route
```

Нет маршрута на сеть `10.1.1.0/24`.

Поэтому в эту сеть из *Router4* не идут пинги.

### RIP

1. Включим *RIP* на всех маршрутизаторах

```
Router1(config)# router rip
```

```
Router1(config-router)# network 172.16.10.0
```

```
Router1(config-router)# network 10.1.1.0
```

```
Router2(config)# router rip
```

```
Router2(config-router)# network 10.1.1.0
```

```
Router4(config)# router rip
```

```
Router4(config-router)# network 172.16.10.0
```

2. На каждом роутере командой *show running-config* посмотрим как маршрутизаторы поняли наши команды. Видим, что сеть *10.1.1.0/24* воспринята как сеть *10.0.0.0/8*, а сеть *172.16.10.0/24* воспринята как сеть *172.16.10.0/16*. Это связано с классами IP адресов.

3. Командой *show ip protocols* посмотрим с какими параметрами работает протокол RIP. Например, для *Router1* имеем

```
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 26 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  Redistributing:  rip
  Default version control:  send version 1, receive any version
    Interface          Send  Recv  Key-chain
    Serial0             1    1  2
    Ethernet0           1    1  2
  Routing for Networks:
    172.16.0.0
    10.0.0.0
  Routing Information Sources:
  Distance: (default is 120)
```

4. Посмотрим таблицы маршрутов

```
Router2# sh ip route
      10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, Ethernet0
R       172.16.0.0 [120/1] via 10.1.1.1, 00:08:37, Ethernet0
```

Есть маршрут на сеть *172.16.10.0/24* через интерфейс *Ethernet* на адрес *10.1.1.1*. Пинги в эту сеть из *Router2* пойдут. Проверьте

```
Router2#ping 172.16.10.1
Router2# ping 172.16.10.2
```

5. Перейдём на другой маршрутизатор

```
Router4# sh ip route
      172.16.0.0/24 is subnetted, 1 subnets
C       172.16.10.0 is directly connected, Serial0
R       10.0.0.0 [120/1] via 172.16.10.1, 00:03:15, Serial0
```

Есть маршрут на сеть *10.1.1.0/24* через интерфейс *Serial* на адрес *172.16.10.1*.

Пинги в эту сеть из *Router4* пойдут. Проверьте

```
Router4#ping 10.1.1.1
Router4# ping 10.1.1.2.
```

6. Командой *debug ip rip* посмотрим как маршрутизаторы обмениваются маршрутной информацией. Например, для *Router1* имеем повторяющиеся каждые 30 секунд сообщения

```
Router1# debug ip rip
RIP: sending update to 255.255.255.255 via Serial0 (172.16.10.1)
      subnet 10.1.1.0, metric 1

RIP: sending update to 255.255.255.255 via Ethernet0 (10.1.1.1)
      subnet 172.16.10.0, metric 1

RIP: received update from 172.16.10.2 on Serial0

RIP: received update from 10.1.1.2 on Ethernet0
```

Выключим трасировку  
*Router1# no debug ip rip*

Сохраните конфигурацию.

## IGRP

Остановим на всех маршрутизаторах *RIP* командой *Router(config)#no router rip*.

1. Включим *IGRP* на всех маршрутизаторах, образуя автономную систему с номером 100

```
Router1(config)# router igrp 100
Router1(config-router)# network 172.16.10.0
Router1(config-router)# network 10.1.1.0
Router2(config)# router igrp 100
Router2 (config-router)# network 10.1.1.0
Router4(config)# router igrp 100
Router4 (config-router)# network 172.16.10.0
```

2. На каждом маршрутизаторе командой *show running-config* посмотрим, как маршрутизаторы поняли наши команды. Видим, что сеть *10.1.1.0/24* воспринята как сеть *10.0.0.0/8*, а сеть *172.16.10.0/24* воспринята как сеть *172.16.10.0/16*. Это связано с классами *IP* адресов.

3. Командой *show ip protocols* посмотрим с какими параметрами работает протокол *IGRP*. Например, для *Router1* имеем

```
Routing Protocol is "igrp 100"
  Sending updates every 90 seconds, next due in 50 seconds
  Invalid after 270 seconds, hold down 280, flushed after 630
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  IGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  IGRP maximum hopcount 100
  IGRP maximum metric variance 1
  Redistributing: igrp 100
  Routing for Networks:
    172.16.0.0
    10.0.0.0
  Routing Information Sources:
  Distance: (default is 100)
```

Переведите сообщение.

4. Посмотрим таблицы маршрутов *Router2# sh ip route*

```
10.0.0.0/24 is subnetted, 1 subnets
C    10.1.1.0 is directly connected, Ethernet0
I    172.16.0.0 [100/273] via 10.1.1.1, 00:04:32, Ethernet0
```

Есть маршрут на сеть *172.16.10.0/24* через интерфейс *Ethernet* на адрес *10.1.1.1*. Пинги в эту сеть из *Router2* пойдут. Проверьте *Router2#ping 172.16.10.1*

```
Router2# ping 172.16.10.2
```

5. Перейдём на другой маршрутизатор

```
Router4# sh ip route
```

```
172.16.0.0/24 is subnetted, 1 subnets
C    172.16.10.0 is directly connected, Serial0
I    10.0.0.0 [100/651] via 172.16.10.1, 00:04:17, Serial0
```

Есть маршрут на сеть *10.1.1.0/24* через интерфейс *Serial* на адрес *2.16.10.1*. Пинги в эту сеть из *Router4* пойдут. Проверьте

```
Router4#ping 10.1.1.1
```

```
Router4# ping 10.1.1.2.
```

6. Командами *debug ip igrp transactions* и *debug ip igrp events* посмотрите как маршрутизаторы обмениваются маршрутной информацией.

Сохраните конфигурацию.

## OSPF

Остановим на всех маршрутизаторах IGRP командой *Router(config)#no router igrp 100*

1. Включим OSPF на всех маршрутизаторах. Дадим процессу OSPF номер 100. Образуем OSPF область 0

```
Router1(config)#router ospf 100
Router1(config-router)# network 172.16.10.0 0.0.0.255 area 0
Router1(config-router)# network 10.1.1.0 0.0.0.255 area 0
Router2(config)#router ospf 100
Router2(config-router)# network 10.1.1.0 0.0.0.255 area 0
Router4(config)# router ospf 100
Router4(config-router)# network 172.16.10.0 0.0.0.255 area 0
```

2. Команда *show running-config* показывает, что маршрутизаторы поняли наши команды в том же виде, как мы их и задавали.

3. Командой *show ip protocols* посмотрим с какими параметрами работает протокол OSPF. Например, для *Router1* имеем

```
Routing Protocol is "ospf 100"
  Sending updates every 90 seconds, next due in 10 seconds
  Invalid after 30 seconds, hold down 0, flushed after 60
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  Redistributing: ospf 100
  Routing for Networks:
    10.1.1.0 0.0.0.255 area 0
    172.16.10.0 0.0.0.255 area 0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.1.1.1         110          00:00:06
    172.16.10.1     110          00:00:06
  Distance: (default is 110)
```

Переведите сообщение.

4. Посмотрим таблицы маршрутов *Router2# sh ip route*

```
10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, Ethernet0
       172.16.0.0/24 is subnetted, 1 subnets
O       172.16.10.0 [110/10] via 10.1.1.1, 00:00:36, Ethernet0
```

Есть маршрут на сеть *172.16.10.0/24* через интерфейс *Ethernet* на адрес *10.1.1.1*. Пинги в эту сеть из *Router2* пойдут. Проверьте

```
Router2#ping 172.16.10.1
Router2# ping 172.16.10.2
```

5. Перейдём на другой маршрутизатор

```
Router4# sh ip route
```

```
172.16.0.0/24 is subnetted, 1 subnets
C       172.16.10.0 is directly connected, Serial0
       10.0.0.0/24 is subnetted, 1 subnets
O       10.1.1.0 [110/74] via 172.16.10.1, 00:00:09, Serial0
```

Есть маршрут на сеть *10.1.1.0/24* через интерфейс *Serial* на адрес *172.16.10.1*. Пинги в эту сеть из *Router4* пойдут. Проверьте

```
Router4#ping 10.1.1.1
Router4# ping 10.1.1.2.
```

6. Команды *show ip ospf interface*, *show ip ospf database* и *debug ip igrp neighbor* покажут вам всю информацию о параметрах протокола OSPF.

Сохраните конфигурацию.



### Контрольные вопросы.

1. Что такое автономная система?
2. Что такое метрика?
3. Какие существуют два класса протоколов динамической маршрутизации.
4. Объясните работу дистанционно-векторных протоколов.
5. Объясните работу протоколов состояния связи.
6. Как узнать, какие протоколы маршрутизации запущены на маршрутизаторе?
7. В чём преимущества и недостатки дистанционно-векторных протоколов и протоколов состояния связи?
8. Что такое сходимость протоколов маршрутизации?
9. Какие параметры влияют на скорость сходимости?
10. Как на маршрутизаторе запустить и настроить протокол маршрутизации *RIP*?
11. Как на маршрутизаторе запустить и настроить протокол маршрутизации *IGRP*?
12. Как на маршрутизаторе запустить и настроить протокол маршрутизации *OSPF*?
13. Как посмотреть содержание пакетов актуализации маршрутной информации протокола *RIP*?
14. Как посмотреть содержание пакетов актуализации маршрутной информации протокола *IGRP*?
15. Уточните, что в *IGRP* понимается под автономной системой?
16. В чём различие в формате команды *router* для *IGRP* и *OSPF*?
17. В чём различие в формате команд *network* для *RIP*, *IGRP* и *OSPF*?

### Задание для самостоятельной работы.

1. Используйте топологию своего варианта из предыдущей лабораторной работы, представленную на рисунке 27.

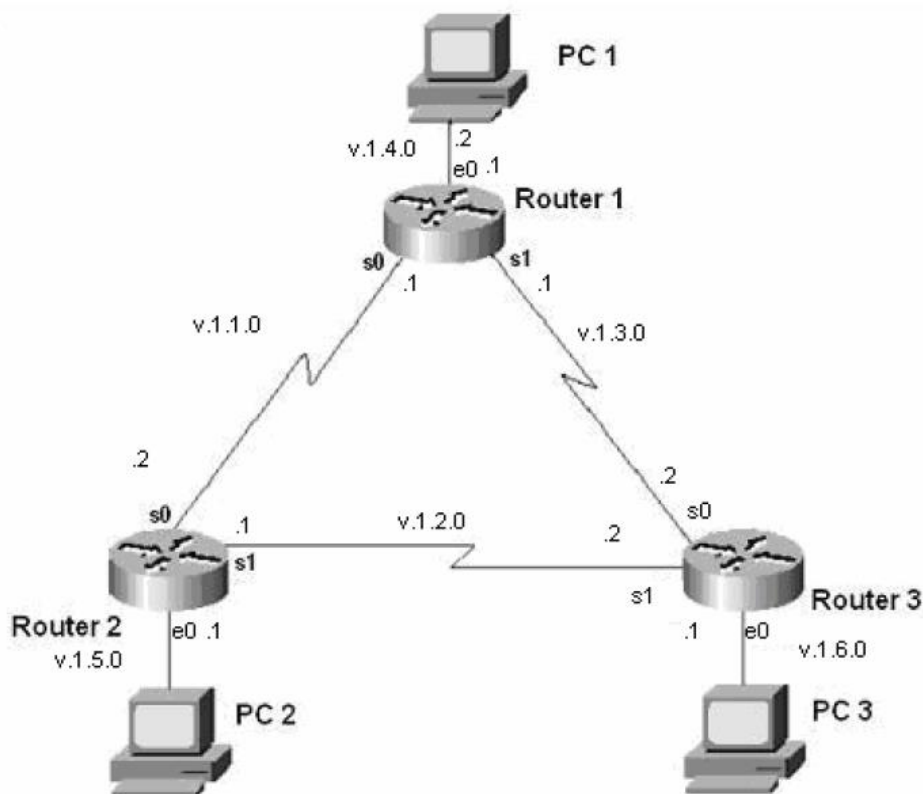


Рисунок 27. Топология сети.

В нашей сети шесть подсетей. Вы видите, что каждый маршрутизатор подключён к трём подсетям.

2. Отредактируйте вручную сохранённую конфигурацию предыдущей успешно выполненной лабораторной работы: уберите в *rtr* файлах маршрутизаторов команды статической маршрутизации.

3. Загрузите отредактированную конфигурацию в симулятор.

4. На каждом маршрутизаторе проверьте правильность загрузки конфигурации командами *show cdp neighbors* и *show ip interface brief*.

Если последовательный интерфейс не поднялся, проверьте командой *show running-config*, что на интерфейсе *DCE* стороны последовательной связи задана команда *clock rate!* Если не задана, то задайте её.

5. Настройте на каждом маршрутизаторе динамическую маршрутизацию по протоколу *RIP*.

6. На каждом маршрутизаторе посмотреть таблицу маршрутизации командой *show ip route*. Сделать скриншоты. Например, для маршрутизатора *router1* для варианта 1 ( $v=1$ ) имеем

```
C      1.1.4.0/24 is directly connected, Ethernet0
C      1.1.1.0/24 is directly connected, Serial0
C      1.1.3.0/24 is directly connected, Serial1
R      1.1.6.0/24 [120/1] via 1.1.3.2, 00:01:43, Serial1
R      1.1.5.0/24 [120/1] via 1.1.1.2, 00:06:30, Serial0
R      1.1.2.0/24 [120/1] via 1.1.1.2, 00:03:35, Serial0
```

7. На каждом компьютере выполните команды трассировки *tracert* других компьютеров. Сделайте скриншоты. Всего шесть скриншотов. Например, трассировка из *PC1* на *PC2* для варианта 1 ( $v=1$ ) имеет вид

```
PC1:#tracert 1.1.5.2

>Type escape sequence to abort.
Tracing the route to 1.1.5.2

 0 1.1.4.1 0 msec 16 msec 0 msec
 1 1.1.1.2 20 msec 16 msec 16 msec
 2 1.1.5.2 20 msec 16 msec *
```

Видим, что путь для пакетов из *PC1* на *PC2* (*1.1.5.2*) лежит последовательно через *Router1* (*Ethernet 1.1.4.1*) и затем через *Router2* (*serial0 1.1.1.2*).

8. Отключим на маршрутизаторе *router1* последовательный интерфейс *serial 0*

```
Router1(config)#interface serial 0
```

```
Router1(config-if)#shutdown
```

9. Через пару минут, когда в сети пройдут обновления маршрутной информации, на каждом маршрутизаторе посмотреть таблицу маршрутизации командой *show ip route*. Сделать скриншоты. Например, для маршрутизатора *router1* для варианта 1 ( $v=1$ ) имеем

```
C      1.1.4.0/24 is directly connected, Ethernet0
C      1.1.3.0/24 is directly connected, Serial1
R      1.1.2.0/24 [120/1] via 1.1.3.2, 00:07:26, Serial1
R      1.1.6.0/24 [120/1] via 1.1.3.2, 00:08:41, Serial1
R      1.1.5.0/24 [120/2] via 1.1.3.2, 00:07:44, Serial1
```

Видим, что таблица изменилась: пропала сеть *1.1.1.0/24* и все пакеты теперь маршрутизируются на адрес *1.1.3.2* через интерфейс *Serial1*.

10. На каждом компьютере выполните команды трассировки *tracert* других компьютеров. Сделайте скриншоты. Всего шесть скриншотов. Например, трассировка из *PC1* на *PC2* для варианта 1 ( $v=1$ ) имеет вид

```
PC1:#tracert 1.1.5.2
```

```
"Type escape sequence to abort."  
Tracing the route to 1.1.5.2
```

```
 1 1.1.4.1 0 msec 16 msec 0 msec  
 2 1.1.3.2 20 msec 16 msec 16 msec  
 3 1.1.2.1 20 msec 16 msec 16 msec  
 4 1.1.5.2 20 msec 16 msec *
```

Видим, что теперь путь для пакетов из *PC1* на *PC2* (1.1.5.2) лежит последовательно через *Router1* (*Ethernet 1.1.4.1*), затем через *Router3* (*serial0 1.1.3.2*) и затем через *Router2* (*serial1 1.1.2.1*).

Сохраните конфигурацию.

11. Отключите *RIP* и настройте на каждом маршрутизаторе динамическую маршрутизацию по протоколу *IGRP*.

12. На каждом маршрутизаторе посмотреть таблицу маршрутизации командой *show ip route*. Сделать скриншоты. Например, для маршрутизатора *router1* для варианта 1 (*v=1*) имеем

```
C      1.1.4.0/24 is directly connected, Ethernet0  
C      1.1.1.0/24 is directly connected, Serial0  
C      1.1.3.0/24 is directly connected, Serial1  
I      1.1.6.0/24 [100/651] via 1.1.3.2, 00:02:38, Serial1  
I      1.1.5.0/24 [100/651] via 1.1.1.2, 00:04:26, Serial0  
I      1.1.2.0/24 [100/651] via 1.1.1.2, 00:08:28, Serial0
```

13. Проверьте, что вы всё сделали правильно. На каждом компьютере выполните команд трассировки *tracert* других компьютеров. Сохраните конфигурацию.

14. Отключите *IGRP* и настройте на каждом маршрутизаторе динамическую маршрутизацию по протоколу *OSPF*.

15. На каждом маршрутизаторе посмотреть таблицу маршрутизации командой *show ip route*. Сделать скриншоты. Например, для маршрутизатора *router1* для варианта 1 (*v=1*) имеем

```
C      1.1.4.0/24 is directly connected, Ethernet0  
C      1.1.1.0/24 is directly connected, Serial0  
C      1.1.3.0/24 is directly connected, Serial1  
O      1.1.5.0/24 [110/65] via 1.1.1.2, 00:00:39, Serial0  
O      1.1.2.0/24 [110/65] via 1.1.3.2, 00:00:22, Serial1  
O      1.1.6.0/24 [110/65] via 1.1.3.2, 00:00:26, Serial1
```

16. Проверьте, что вы всё сделали правильно. На каждом компьютере выполните команд трассировки *tracert* других компьютеров. Сохраните конфигурацию.

## Список литературы и информационных источников.

1. Олифер, В. Г. Компьютерные сети. Принципы, технологии, протоколы / В. Г. Олифер, Н. А. Олифер. – 4-е изд. – СПб.: Питер, 2010. -944 стр.
2. Уилсон, Э. Мониторинг и анализ сетей. Методы выявления неисправностей / Э. Уилсон. – М.: ЛОРИ, 2012. - 386 стр.
3. Филимонов, А. Протоколы Интернета / А. Филимонов. – СПб.: БХВ-Петербург, 2003. – 528 стр.
4. Золотов, С. Протоколы Internet / С. Золотов. – СПб.: БХВ-Петербург, 1998. – 301 стр.
5. Семенов, Ю. А. Telecommunication technologies – телекоммуникационные технологии [Электронный ресурс]. Режим доступа: [https://www.opennet.ru/docs/RUS/inet\\_book/](https://www.opennet.ru/docs/RUS/inet_book/), свободный. – Загл. с экрана. – Рус.
6. Коннов Н. Н. Анализ сетевых протоколов. Лабораторный практикум по курсу «Сети ЭВМ и телекоммуникации», Часть 1/ Н. Н. Коннов, В. Б. Механов. – Пенза: Издательство ПГУ, 2010. - 69 стр.
7. Костров Б. Телекоммуникационные и вычислительные сети. Архитектура, стандарты и технологии./ Б. Костров, В. Ручкин, Т. Калинкина. - СПб.: БХВ-Петербург, 2010 г. - 288 стр.
8. Бройдо В. Вычислительные системы, сети и телекоммуникации: учебник для вузов (4-е издание). / В. Бройдо, О. Ильина. - СПб.:Питер, 2010 г. -560 стр.
9. Таненбаум Э. Компьютерные сети. / Э.Таненбаум. – СПб.: Питер, 2012. – 960 с.
10. Ручкин В.Н. Архитектура компьютерных сетей. / В.Н. Ручкин, В.А.Фулин. – М.: Диалог-МИФИ, 2008 г. -240 стр.
11. Попов И. Компьютерные сети./ И. Попов, Н. Максимов. – М.: Форум, 2010 г. - 464 стр.