

Лабораторная работа №1. Знакомство со средой

Теоретическая часть

Программные продукты фирмы Boson дают возможность создавать сетевые топологии (до 200 устройств) 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.1.1.

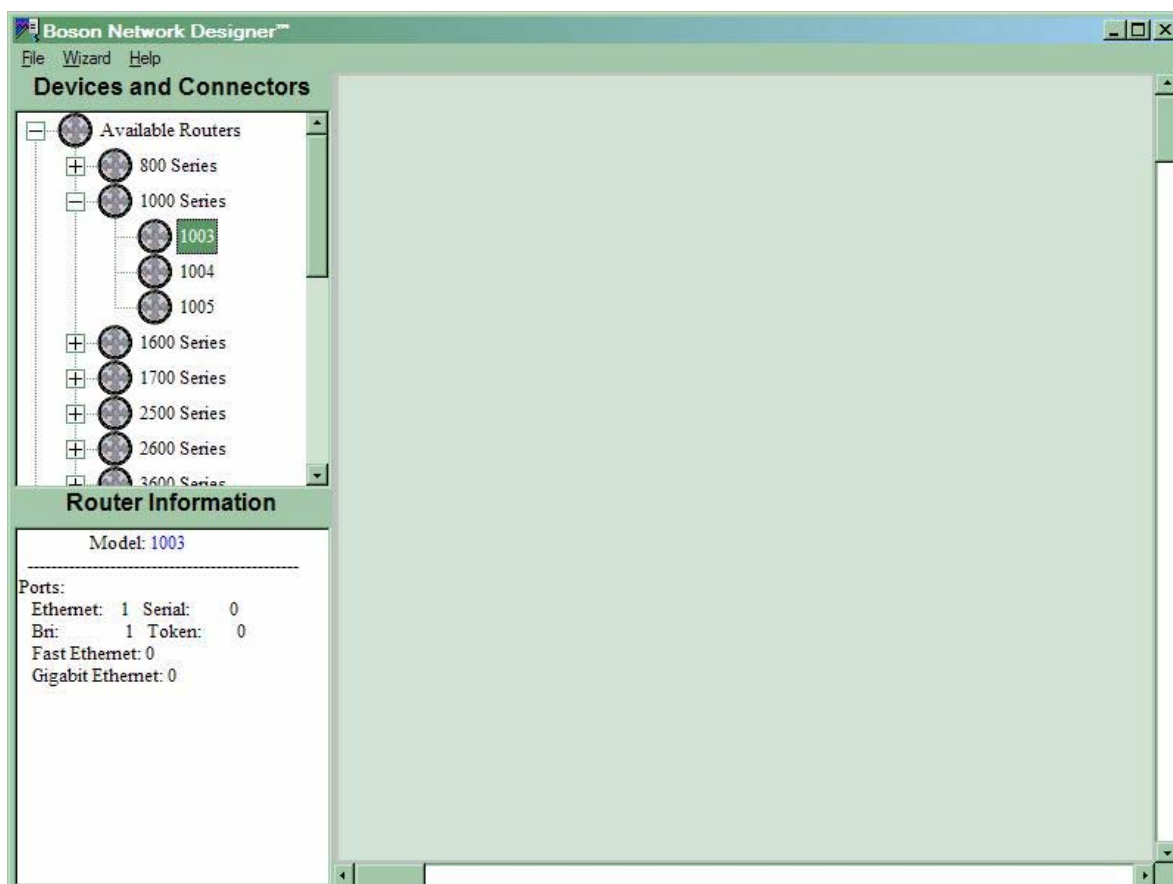


Рис.1.1. Общий вид программы Boson Network Designer.

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-n-Drop (выбираем объект, удерживаем левую кнопку мыши, и перетаскиваем в нужное окно), добавить необходимые элементы из раздела Devices and Connectors в правое поле главного окна программы, (см. рис. 1.2).

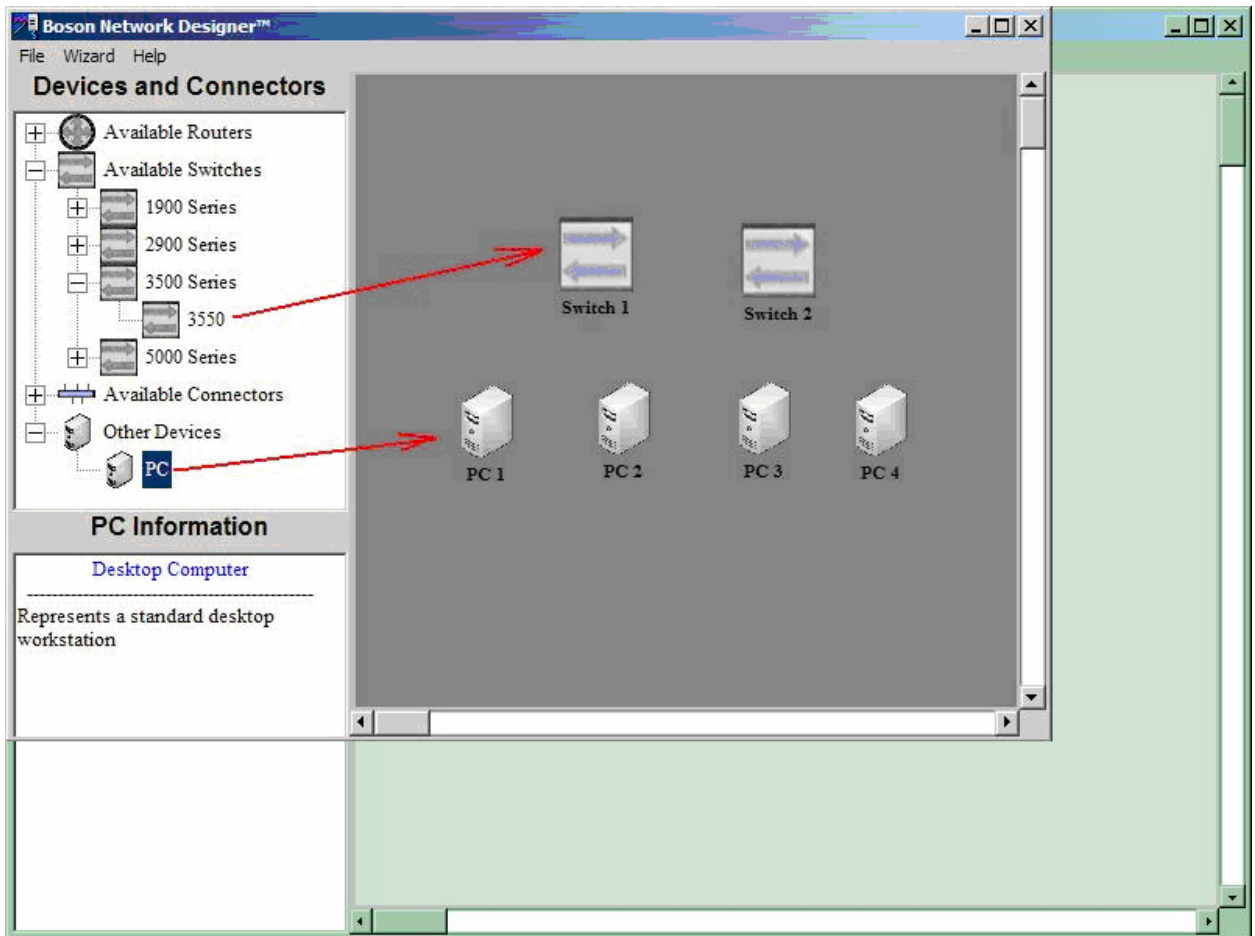


Рис.1.2. Добавление элементов сети.

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 тремя 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 содержимое.

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. Если появится Lab Navigator (лаб навигатор), то закройте его.

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).

1. На Панели Инструментов есть кнопка **NetMap**. (см. рис. 1.3)

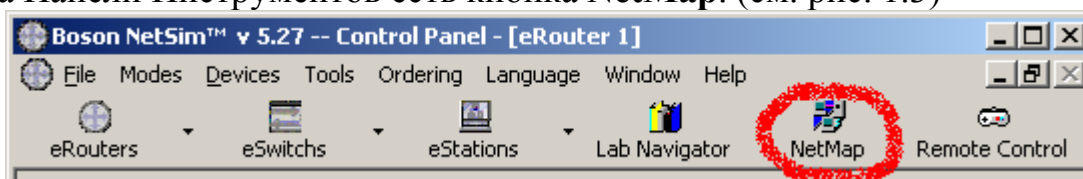


Рис.1.3. Кнопка NetMap.

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 **eStations** для получения списка устройств в текущей топологии сети. Например, когда вы выберете Router 1, то вы подключитесь к Router 1 (см. рис. 1.4)

3. В главном меню программы есть пункт **Devices** и далее пункты для выбора типа устройства и устройства (см. рис. 1.5).

4. На Панели Инструментов есть кнопка **Remote Control** (см. рис. 1.5) 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)** откроет 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.

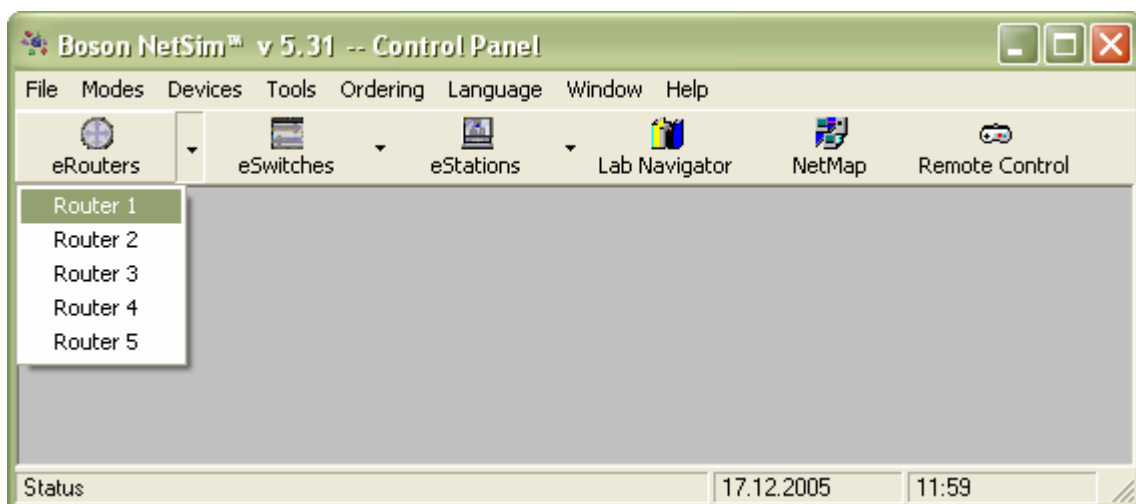


Рис.1.4.

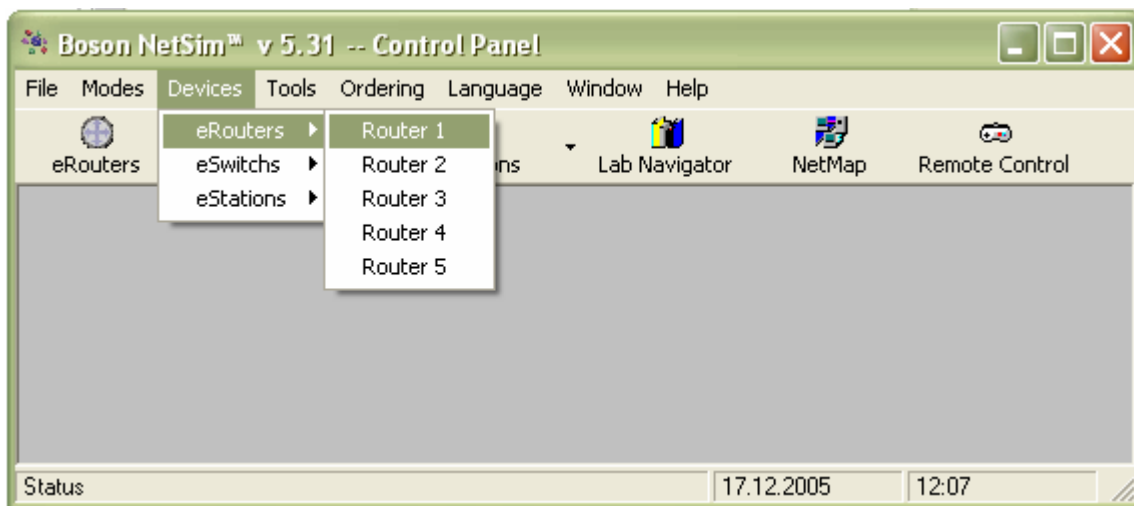


Рис. 1.5

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 можете закрыть окно Telnet либо, закрыв окно, либо, перейдя в режим телнета (Ctrl+]), 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 **eStations** и пункт **Devices** главного меню.

Интересно отметить, что при выборе режима **Advanced Mode** симулятор запускает 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 name 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 ветки **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.1.6. 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 eStatons -> PC1. Появится окно ИКР компьютера PC1.

7. Список команд получим, нажав ?. Для конфигурирования компьютера используем команду **winipcfg**. Наберём её в ИКР и нажмём **Enter**. Появится окно (см. рис. 1.7)

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

Табл.1

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

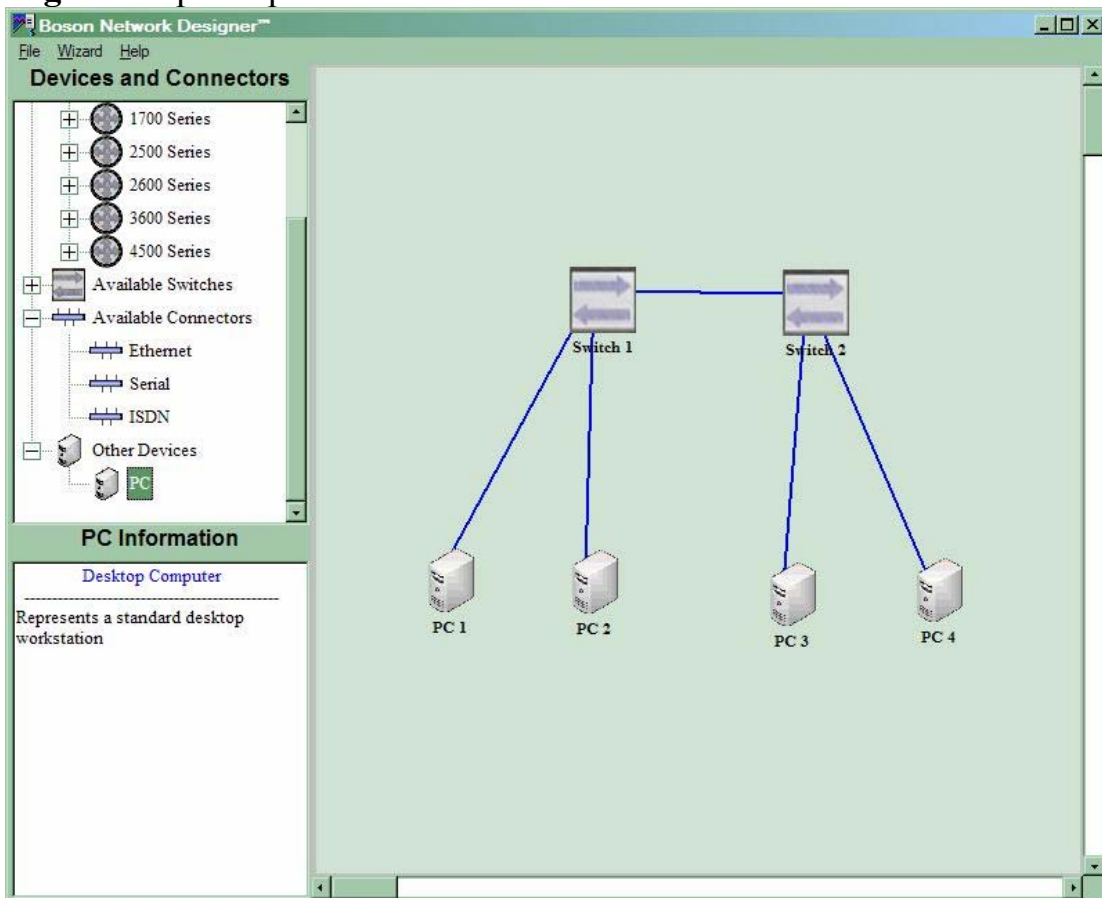


Рис. 1.6.

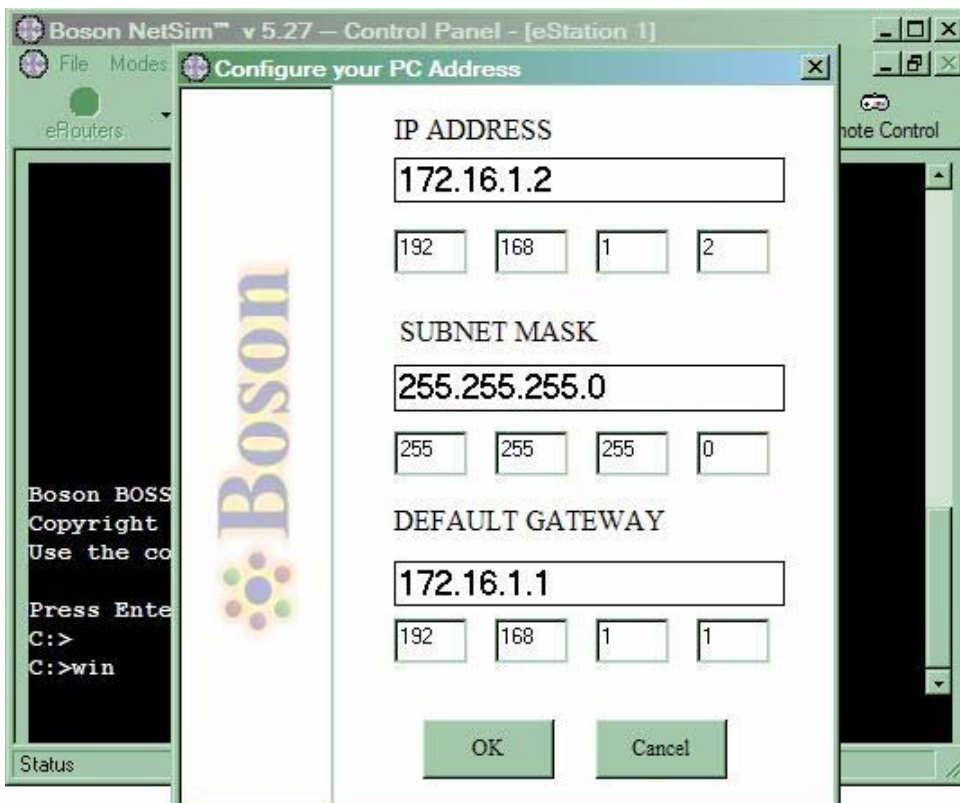


Рис. 1.7.

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 1.8.

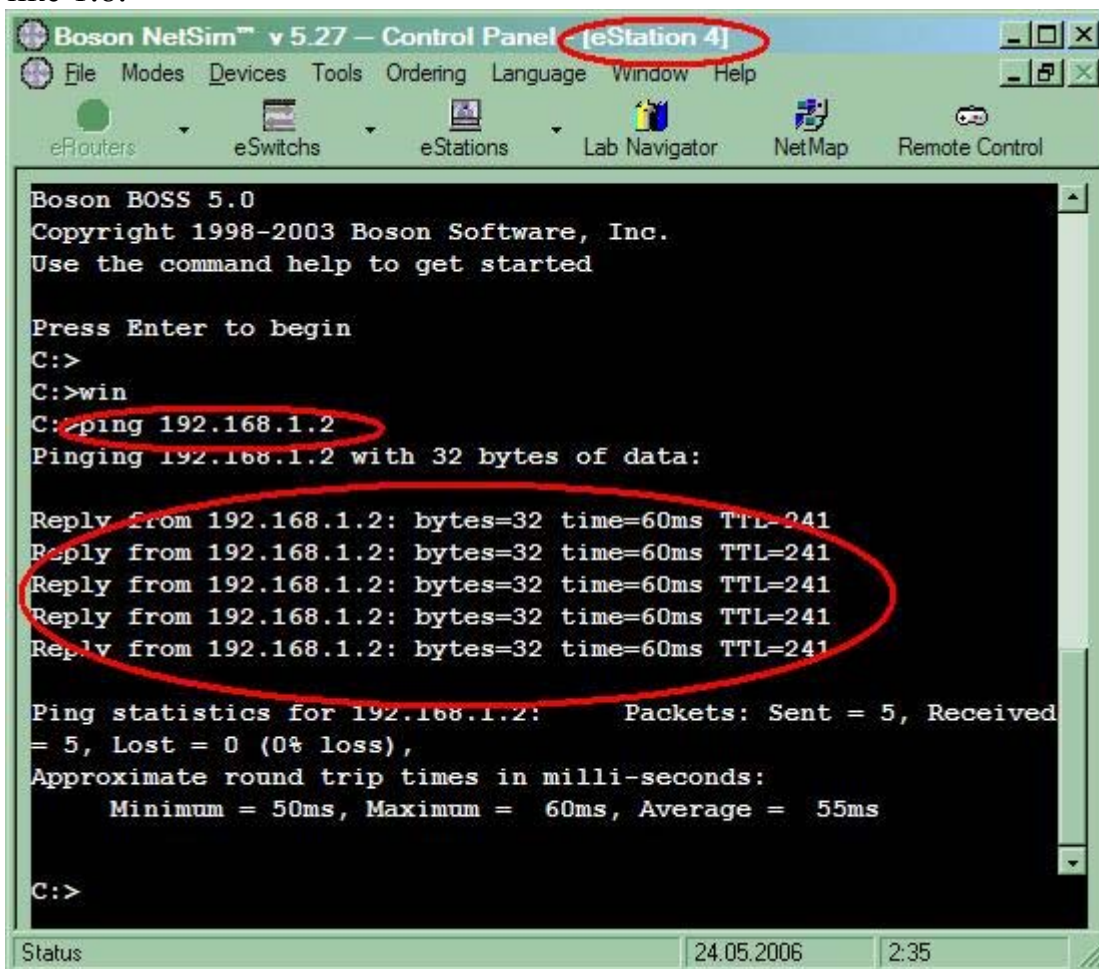


Рис. 1.8.

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. Все компьютеры при настройке имеют имена C:>. Это ненаглядно. Откройте в потерад поочерёдно rtr файлы компьютеров и задайте компьютерам информативные имена, отредактировав в rtr файле команду hostname. Старую команду закомментируйте символом ! в начале строки. Например, для PC1 возьмите hostname PC1>. Тогда компьютер PC1 при настройке будет иметь в командной строке имя PC1>, а не C:>.

14. 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.

15. 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.

16. 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 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 (1-12) и выполните в Boson **задание для самостоятельной работы**
5. Предъявите преподавателю результат выполнения задания для самостоятельной работы. Покажите ему свои rtr, nws и top файлы. Продемонстрируйте ему, что любой компьютер пингуется из любого компьютера.
6. Оформите отчёт.
7. Защитите отчёт.

Задание для самостоятельной работы

1. Создайте топологию на Рис 1.9
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

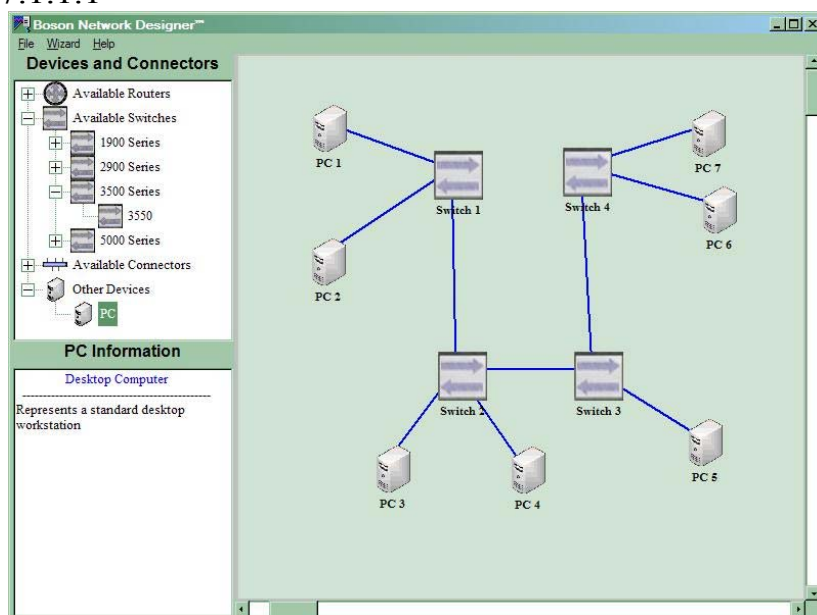


Рис 1.9

3. Назначьте компьютерам разные имена в командной строке.

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

Содержание отчёта.

Отчёт готовится в электронном виде и распечатывается. Отчёт содержит

1. Скриншот топологии из рисунка 1.6.

2. Конфигурации каждого из четырёх компьютеров из rtr файлов, созданных при выполнении практической части (уберите пустые строки и лишние комментарии).

3. Скриншот топологии из рисунка 1.9

2. Конфигурации каждого из семи компьютеров из rtr файлов, созданных при выполнении задания для самостоятельной работы (уберите пустые строки и лишние комментарии).

4. Скриншот выполнения команды пинг согласно варианту

Вариант v	Пинг из	Пинг в	Вариант v	Пинг из	Пинг в
1	PC1	PC6	7	PC7	PC5
2	PC2	PC7	8	PC1	PC6
3	PC3	PC1	9	PC2	PC7
4	PC4	PC2	10	PC3	PC1
5	PC5	PC3	11	PC4	PC2
6	PC6	PC4	12	PC5	PC3

Лабораторная работа №2. Введение в межсетевую операционную систему IOS компании Cisco.

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, подынтерфейса, 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 СОМ-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 переходу 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(config-if)#`, 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:

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-conf ig	Показывает содержание активной конфигурации
show interfaces	Показывает данные обо всех интерфейсах на устройстве
show protocols	Выводит данные о протоколах третьего сетевого уровня.

Таблица 1. Show романди

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

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. Поле 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
```

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

Таблица 2. Результаты команды ping

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

так, как вы видите на рисунке 1: для двух первых устройств, примите имена по умолчанию, а третье назовите Router 4.

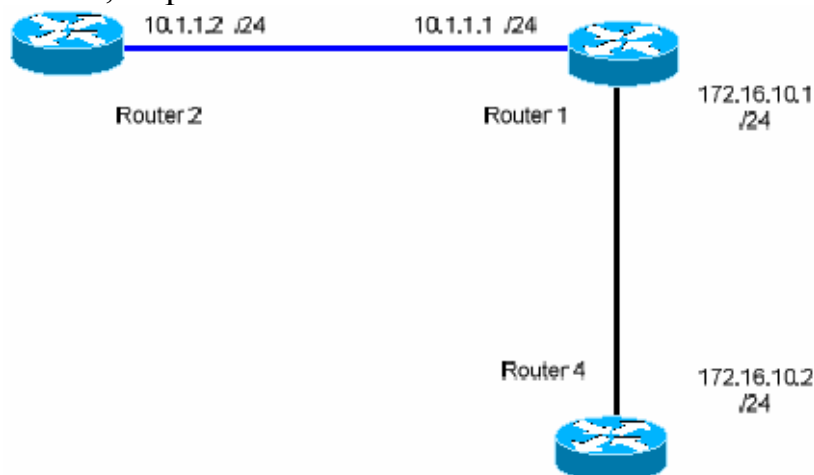


Рис.1

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.

Ознакомление 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 в 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 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 доступа "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

не 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: оконечным 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 подключенных 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 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 Ser0.

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 отконфигурированные 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 нас постигли 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 vty. 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> потом <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> потом <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, nws и top файлы. Продемонстрируйте ему, что компьютеры пингуются согласно таблице 4.
6. Продемонстрируйте работу telnet.
7. Оформите отчёт. Содержание отчёта смотри ниже.
8. Защитите отчёт.

Задание для самостоятельной работы

1. Поучите свой вариант

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 дизайнере подходящие устройства и создайте топологию, изображённую на Рис. 2. 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 3. Все маски равны 255.255.255.0. Например, для варианта 7 (v=7) имеем адреса из Таблицы 2.

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 2

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

Таблица 3

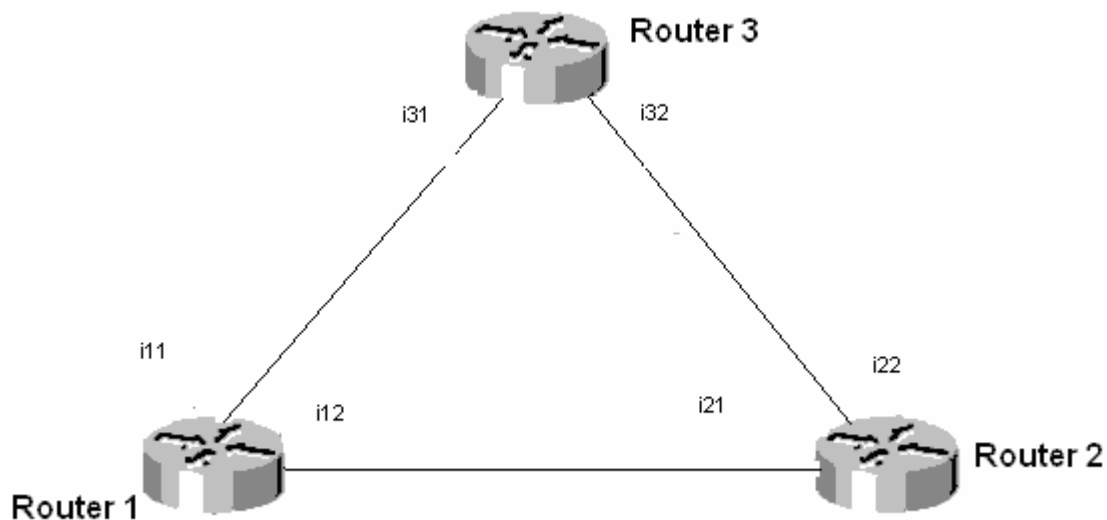


Рис 2.

3. Проверьте, что адреса назначены. На каждом устройстве выполните команду **show ip interface brief**. Сохраните скриншоты.

4. Если сделано всё правильно вы сможете пропинговать из любого компьютера определённые (но не все) адреса интерфейсов других компьютеров.

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

Таблица 4

Сделайте это. Сохраните скриншоты для пингов соединений, отмеченных в таблице 4 знаком *.

5. Выполните на Router1 расширенный пинг. Сохраните 3 скриншота для пингов: от i12 к i21, от i11 к i31 и от i22 к i32.

6. Настройте на Router1 Telnet. Задайте пароль.

7. Перейдите на Router2. Зайдите по Telnet на Router1. Выполните команду **show user**. Приостановите сессию. Возобновите сессию. Убейте сессию. Сохраните скриншот консоли Router2.

8. Сохраните топологию и конфигурацию.

Содержание отчёта.

Отчёт готовится в электронном виде и распечатывается. Отчёт содержит

1. Скриншот топологии из рисунка 1

2. Конфигурации трёх маршрутизаторов из rtr файлов, созданных при выполнении практической части (уберите пустые строки и лишние комментарии).

3 Скриншот топологии из рисунка 2

4. Все скриншоты, указанные в **Задании для самостоятельной работы**

5. Конфигурации трёх маршрутизаторов из rtr файлов (уберите пустые строки и лишние комментарии).

Лабораторная работа №3. Статическая маршрутизация

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 широковещательными (почему?).

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 достигим 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):

S10.6.0.0 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 используются, 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 хопа | 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

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 lo 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`

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

Таблица 1.

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 хопа 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(config)#**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(config)#**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(conf)#**int loopback 0 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. Почему для сетей типа Ethernet рекомендуется всегда использовать форму (2) команды маршрутизации?
10. Объясните значения полей в командах маршрутизации.
11. Почему в качестве поля Адрес рекомендуют использовать адрес следующего хопа по пути к сети назначения.
12. Когда используется маршрутизация по умолчанию?
13. Когда используют интерфейс петля?
14. Как работает команда трасировки?

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 пунктов 8 и 9 задания для самостоятельной работы.
6. Оформите отчёт. Содержание отчёта смотри ниже.
7. Защитите отчёт .

Задание для самостоятельной работы

1. Построить в Boson Network Designer топологию, представленную на рисунке. Использовать маршрутизаторы модели 2501.

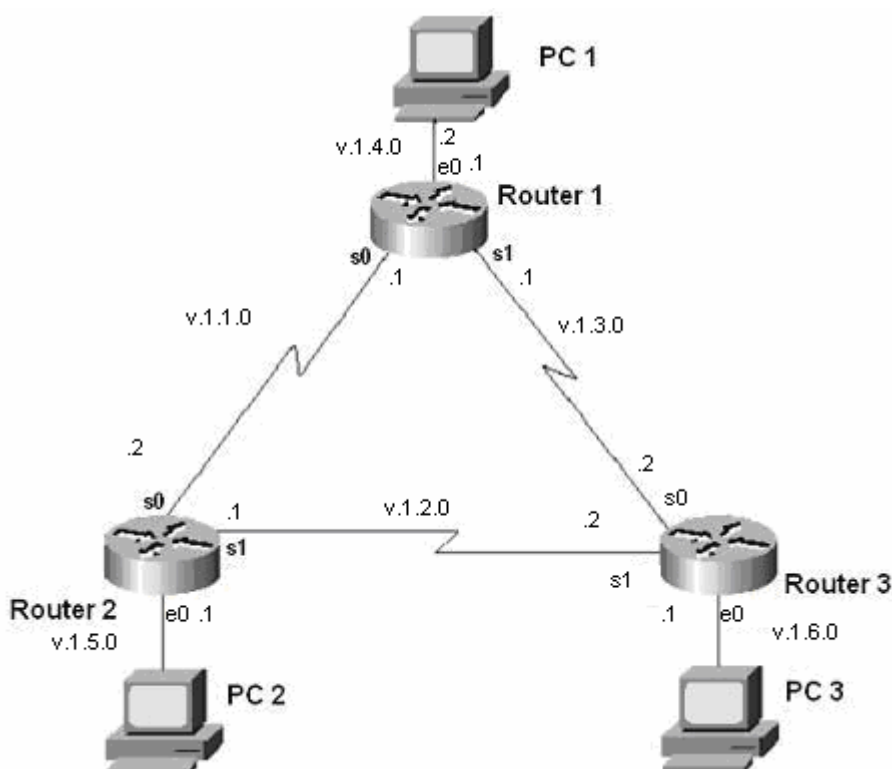


Рис. 1

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 1 и таблице 1, в которых *v* – это номер варианта. Все маски 255.255.255.0. Не забудьте назначить шлюзы по умолчанию для компьютеров согласно таблице 1.

	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

Таблица 1

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 1.1.1.2 ближайшего внешнего интерфейса на пути в эту сеть. Это сделает команда

```
router1(config)#ip route 1.1.5.0 255.255.255.0 1.1.1.2
```

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 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.

Содержание отчёта.

Отчёт готовится в электронном виде и распечатывается. Отчёт содержит

1. Скриншот топологии, созданной при выполнении практической части.
2. Конфигурации трёх маршрутизаторов из `rtg` файлов, созданных при выполнении практической части (уберите пустые строки и лишние комментарии).
3. Скриншот топологии из рисунка 1 с адресами своего варианта
4. Таблицу 2 с адресами своего варианта
5. Конфигурации трёх маршрутизаторов и трёх компьютеров из `rtg` файлов, созданных при выполнении задания для самостоятельной работы.
6. Все скриншоты, указанные в задании для самостоятельной работы

Лабораторная работа №4. Динамическая маршрутизация

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 используется 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.1).

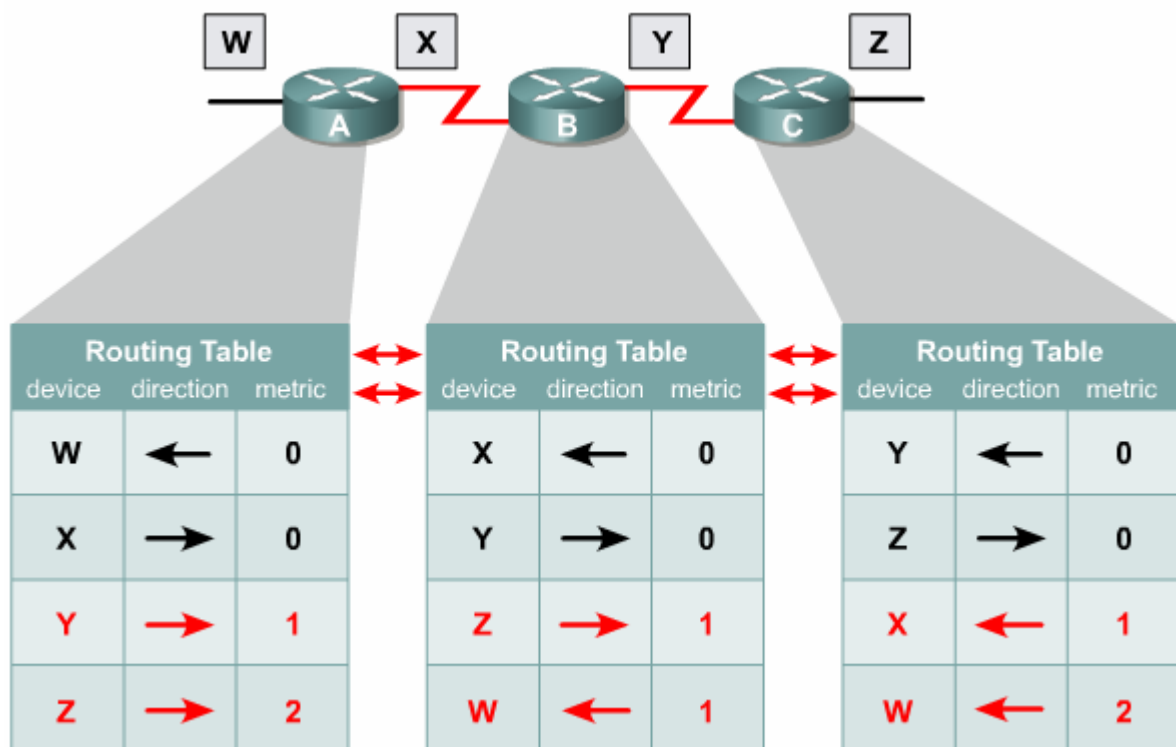


Рис.1. Дистанционно-векторная маршрутизация.

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 имеют 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 хопх 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 як 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.2 примере команды `network 1.0.0.0` и `network 2.0.0.0` задают непосредственно подключенные к маршрутизатора Cisco A сети.

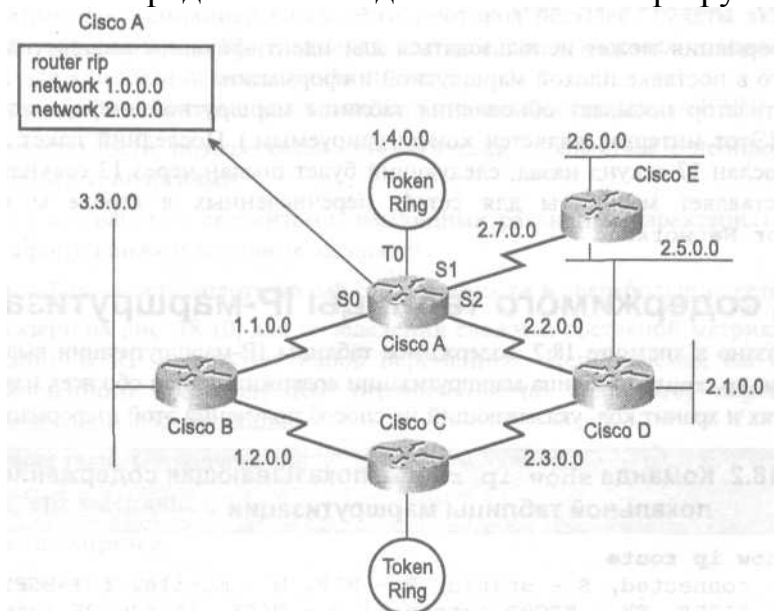


Рис.2.

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 собою 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 протоколу характеризует универсальность, позволяющая 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 протоколу 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.3 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.

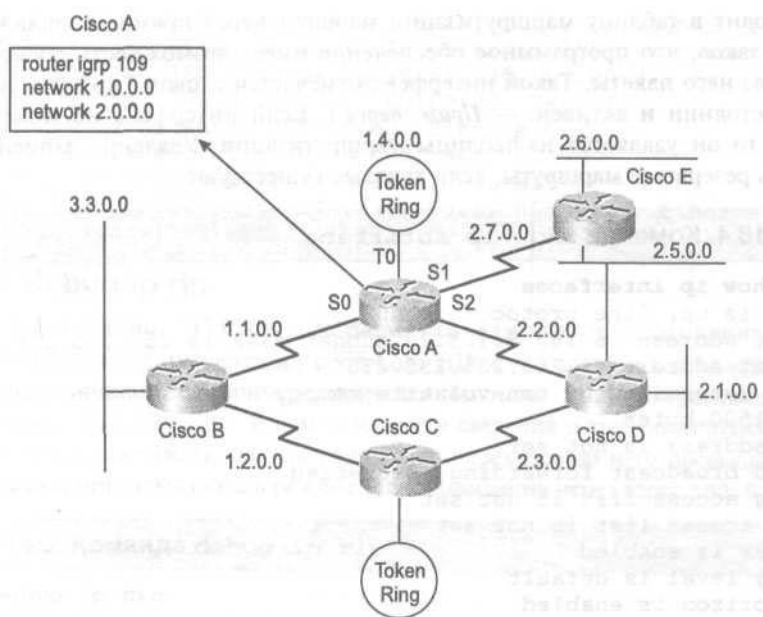


Рис.3.

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.

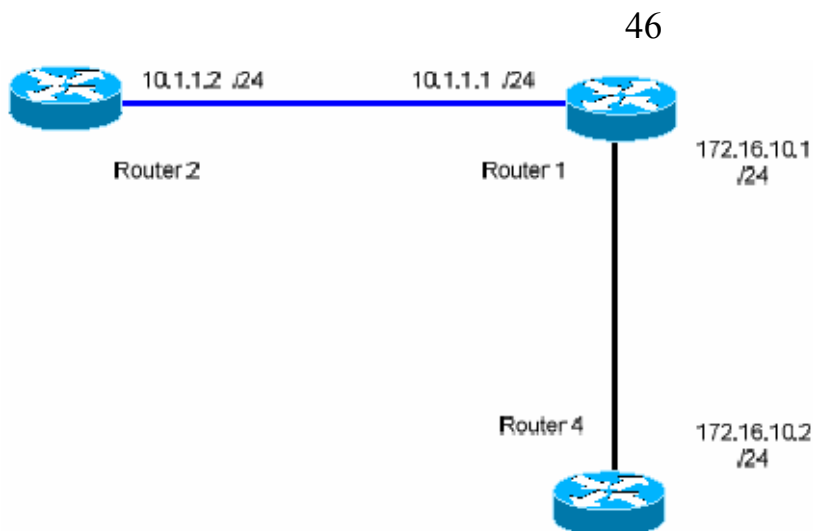
Команда **show ip ospf neighbor** показывает важную информацию, касающуюся состояния соседей.

Практическая часть

1. Загрузите в симулятор топологию и конфигурацию, использованную практической части лабораторной работы №2.

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 №2 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 №3 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)

```

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
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 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. 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. Образует OSPF область 0

Router1(config)#**router ospf 100**

Router1(config-router)# **network 172.16.10.0 0.0.255 area 0**

Router1(config-router)# **network 10.1.1.0 0.0.255 area 0**

Router2(config)#**router ospf 100**

Router2(config-router)# **network 10.1.1.0 0.0.255 area 0**

Router4(config)# **router ospf 100**

Router4(config-router)# **network 10.1.1.0 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?

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. Используя вариант из предыдущей лабораторной работы, выполните в Boson задание для самостоятельной работы.
5. Предъявите преподавателю результат выполнения пунктов 7, 10, 14 и 18 задания для самостоятельной работы.
6. Оформите отчёт. Содержание отчёта смотри ниже.
7. Защитите отчёт.

Задание для самостоятельной работы

1. Используйте топологию своего варианта из предыдущей лабораторной работы, представленную на рисунке 5.

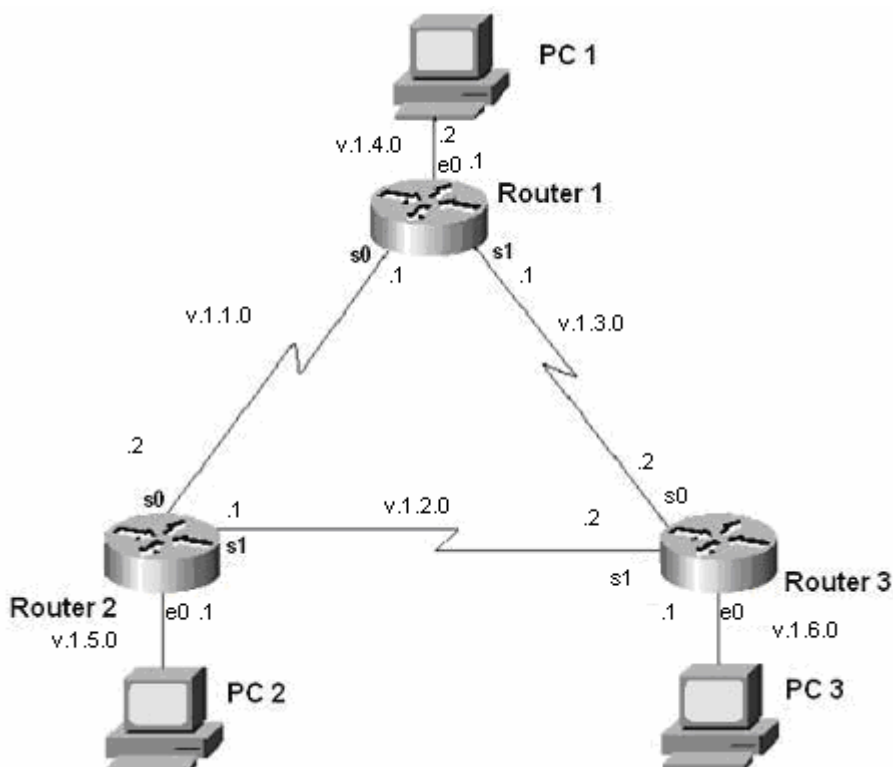


Рис. 5

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 `rtt` 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. Сделать 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 1.1.1.2 20 msec 16 msec 16 msec
 3 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`. Сделайте 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`. Сделайте 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.

Содержание отчёта.

Отчёт готовится в электронном виде и распечатывается. Отчёт содержит

1. Скриншот топологии, созданной при выполнении практической части.
2. Конфигурации трёх маршрутизаторов из `rtr` файлов, созданных при выполнении практической части для каждого протокола маршрутизации. Итого содержимое 9 (девяти) `rtr` файлов.
3. Скриншот топологии из рисунка 5 с адресами своего варианта
4. Таблицу 2 предыдущей лабораторной работы с адресами своего варианта
5. Конфигурации трёх маршрутизаторов из `rtr` файлов, созданных при выполнении задания для самостоятельной работы для каждого протокола маршрутизации. Итого содержимое 9 (девяти) `rtr` файлов.
6. Все скриншоты, указанные в задании для самостоятельной работы

Лабораторная работа №5. Бесклассовая адресация CIDR и маски переменной длины VLSM

Теоретическая часть

Масштабируемая сеть требует схемы адресации допускающей рост. Однако вследствие неконтролируемого роста сети могут возникнуть ряд непредвиденных последствий. По мере добавления узлов и подсетей в сеть предприятия может возникнуть нехватка свободных адресов и потребуются изменение схемы существующих адресов. Этого можно избежать путём тщательного планирования масштабируемой адресной системы сети предприятия.

К сожалению, архитекторы TCP/IP не могли предсказать экспоненциального роста Интернет, и в настоящее время остро стоит проблема распределения адресов.

Когда в 80-х годах внедрялся TCP/IP, он базировался на двухуровневой адресной схеме. Старшая часть 32-битового IP адреса определяла номер (адрес) сети, а младшая - номер хоста. Адрес сети необходим для взаимодействия сетей. Маршрутизаторы используют сетевую часть адреса для организации связи между хостами из различных сетей.

Для удобства человеческого восприятия IP адрес записывается в виде четырёх десятичных чисел, разделённых точками. 32-битовый адрес делится на четыре группы по восемь бит, называемых октетами. Каждый октет записывается в десятичном виде и разделяется точками. Например

101011000001111101000000000010001 <-> 10101100 00011110 10000000 00010001 <->

172 30 128 17 <-> 172.30.128.17

Возникает вопрос, как в любом IP адресе выделить адрес сети и адрес хоста? В начале использования TCP/IP для решения этого вопроса использовалась классовая система адресации. IP адреса были разбиты на пять непересекающихся классов. Разбивка осуществлена согласно значениям нескольких первых бит в первом октете.

Если первый бит в первом октете равен нулю, то это адрес класса А. Адреса класса В начинаются с бинарных 10. Адреса класса С начинаются с бинарных 110.

В адресах класса А адрес сети располагается в первом октете. В классе В для адресации сети используется первый и второй октеты. В классе С для адресации сети используется первый, второй и третий октеты. Использование классов D и E специфично и здесь не рассматривается.

В современных сетях классы часто игнорируются, а используется бесклассовая IP схема, основанная на масках подсетей.

Здесь и далее мы будем использовать маски в виде последовательности бинарных единиц, переходящей в последовательность бинарных нулей общей длиной в 32 бита. Маски принято записывать в десятичной форме подобно IP адресам

11111111111111111100000000000000 <-> 11111111 11111111 00000000 00000000
 <->
 255 255 0 0 <-> 255.255.0.0

Маска подсети является необходимым дополнением к IP адресу. Если бит в IP адресе соответствует единичному биту в маске, то этот бит в IP адресе представляет номер сети, а если бит в IP адресе соответствует нулевому биту в маске, то этот бит в IP адресе представляет номер хоста. Так для маски 255.255.0.0 и адреса 172.24.100.45 номер сети будет 172.24.0.0, а для маски 255.255.0.0 номер сети будет 172.24.100.0.

Другая форма записи маски - /N, где N – число единиц в маске. Эта форма используется только в сочетании с IP адресом. Например, для маски 255.255.0.0 и адреса 172.24.100.45 пишут 172.24.100.45/16.

Все адреса класса А имеют маску 255.0.0.0, адреса класса В имеют маску 255.255.0.0, а адреса класса С имеют маску 255. 255. 255.0. Обратное утверждение неправомерно, так как при определении класса используются первые биты в первом октете адреса.

Если организация располагает сетью класса В (маска 255.255.0.0), то она может разбить эту сеть на подсети, используя маску 255.255.255.0. Например, если адрес 172.24.100.45 принадлежит организации, то номером сети класса В будет 172.24.0.0, а номер внутрикорпоративной подсети будет равен 172.24.100.0. Заметим, что полученные подсети не будут являться сетями класса С.

Если число нулей в маске равно М, то число доступных адресов хостов в подсети равно 2^{M-2} . То есть два адреса в подсети использовать не рекомендуется. Один из этих адресов, у которого последние М бит равны нулю, называется адресом подсети, а второй из этих адресов у которого последние М бит равны единице называется широковещательным адресом. Так для адреса 172.24.100.45/24 адрес подсети равен 172.24.100.0, а широковещательным адрес равен 172.24.100.255. Число адресов в подсети равно $2^8 - 2 = 254$.

Адреса класса А и В составляют около 75 процентов адресного пространства. Количество сетей классов А и В приблизительно равно 17000. Приобретение сети класса В, а тем более класса А в настоящее время весьма проблематично. Адреса класса С составляют около 12.5 процентов адресного пространства. Количество сетей класса С приблизительно равно 2.1 миллиона. К сожалению сеть класса С ограничена 254 адресами, что не отвечает нуждам больших организаций, которые не могут приобрести адреса класса А или В.

Классовая IP адресация, даже с использованием подсетей, не может удовлетворить требования по масштабируемости для Интернет сообщества.

Уже в начале 90-х годов почти все сети класса В были распределены. Добавление в Интернет новых сетей класса С приводило к значительному росту таблиц маршрутов и перегрузке маршрутизаторов. Использование бесклассовой адресации позволило в значительной мере решить возникшие проблемы.

CIDR

Современные маршрутизаторы используют форму IP адресации называемую безклассовой междоменной маршрутизацией (Classless Interdomain Routing (CIDR)), которая игнорирует классы. В системах, использующих классы, маршрутизатор определяет класс адреса и затем разделяет адрес на октеты сети и октеты хоста, базируясь на этом классе. В CIDR маршрутизатор использует биты маски для определения в адресе сетевой части и номера хоста. Граница разделения адреса может проходить посреди октета.

CIDR значительно улучшает масштабируемость и эффективность IP по следующим пунктам:

- гибкость;
- экономичное использование адресов в выделенном диапазоне;
- улучшенная агрегация маршрутов;
- Supernetting - комбинация непрерывных сетевых адресов в адрес новый надсети, определяемый маской.

CIDR позволяет маршрутизаторам агрегировать или суммировать информацию о маршрутах. Они делают это путём использования маски вместо классов адресов для определения сетевой части IP адреса. Это сокращает размеры таблиц маршрутов, так как используется лишь один адрес и маска для представления маршрутов ко многим подсетям.

Без CIDR и агрегации маршрутов маршрутизатор должен содержать индивидуальную информацию для всех подсетей.

Рассмотрим сеть класса A 44.0.0.0/8, в которой рассматривается на 8 подсетей

Сетевой номер	Первый октет	Второй октет	Третий октет	Четвёртый октет
44.24.0.0/16	00101100	00011000	00000000	00000000
44.25.0.0/16	00101100	00011001	00000000	00000000
44.26.0.0/16	00101100	00011010	00000000	00000000
44.27.0.0/16	00101100	00011011	00000000	00000000
44.28.0.0/16	00101100	00011100	00000000	00000000
44.29.0.0/16	00101100	00011101	00000000	00000000
44.30.0.0/16	00101100	00011110	00000000	00000000
44.31.0.0/16	00101100	00011111	00000000	00000000

Первые два октета (16 бит) представляют адрес подсети. Так как первые 16 бит адреса каждой из этих восьми подсетей уникальны, то классовой маршрутизатор видит восемь уникальных сетей и должен создать строку в таблице маршрутов для каждой из этих подсетей.

Однако эти восемь адресов подсетей имеют общую часть: первые 13 бит одинаковы. CIDR-совместимый маршрутизатор может суммировать маршруты к этим восьми подсетям, используя общий 13-битовый префикс в адресах:

00101100 00011. Для представления этого префикса в десятичной форме дополним его справа нулями

10101100 00011000 00000000 00000000 = 172.24.0.0.

13-битовая маска подсети имеет вид

11111111 11111000 00000000 00000000 = 255.248.0.0.

Следовательно один адрес и одна маска определяет бесклассовый префикс, который суммирует маршруты к восьми подсетям: 172.24.0.0/13.

Supernetting

Supernetting это практика использования битовой маски для группировки нескольких классовых сетей в виде одного сетевого адреса. Supernetting и агрегирование маршрутов есть разные имена одного процесса. Однако термин supernetting чаще применяется, когда агрегируемые сети находятся под общим административным управлением. Supernetting берёт биты из сетевой порции маски, а subnetting берёт биты из порции маски, относящейся к хосту. Supernetting и агрегирование маршрутов является инверсным понятием по отношению к subnetting.

Так как сети классов А и В практически исчерпаны, то организации вынуждены запрашивать у провайдеров несколько сетей класса С. Если компания получает блок непрерывных адресов в сетях класса С, то можно использовать supernetting и все адреса в компании будут лежать в одной большей сети или надсети.

Рассмотрим компанию АБВ, которой требуется адреса для 400 хостов. При классовой адресации компания должна запросить у центральной интернет службы InterNIC сеть класса В. Если компания получит такую сеть, то десятки тысяч адресов в ней не будут использоваться. Альтернативой является получение двух сетей класса С, что даёт $254 \times 2 = 504$ адреса для хостов. Недостаток этого подхода состоит в необходимости поддержки маршрутизации для двух сетей.

При бесклассовой адресной системе supernetting позволяет компании АБВ получить необходимое адресное пространство с минимальным количеством неиспользуемых адресов и без увеличения размера таблиц маршрутизации. Используя CIDR, АБВ запрашивает блок адресов у своего Интернет провайдера, а не у центральной Интернет службы InterNIC. Провайдер определяет потребности АБВ и выделяет адресное пространство из своего адресного пространства. Провайдер берёт на себя управление адресным пространством в своей внутренней бесклассовой системе. Все внешние Интернет маршрутизаторы содержат только суммирующие маршруты к сети провайдера. Провайдер сам поддерживает маршруты, более специфичные для своих клиентов, включая АБВ. Этот подход существенно уменьшает размеры таблиц маршрутов для всех маршрутизаторов в Интернет.

Пусть АБВ получил у провайдера две сети класса С, адреса в которых непрерывны: 207.21.54.0 и 207.21.55.0.

207.21.54.0	110001111	00010101	00110110	00000000
207.21.55.0	110001111	00010101	00110111	00000000

Из таблицы видно, что адреса имеют общий 23-битовый префикс 11001111 00010101 0011011. Дополняя префикс справа нулями 11001111 00010101 00110110 00000000, получим сеть с 23- битовой маской , 207.21.54.0/23.

Провайдер предоставляет сеть компании АБВ внешнему миру как сеть 207.21.54.0/23.

CIDR позволяет провайдерам эффективно распределять и суммировать непрерывные пространства IP адресов.

VLSM

Маска переменной длины (Variable-Length Subnet Mask (VLSM)) позволяет организации использовать более одной маски подсети внутри одного и того же сетевого адресного пространства. Реализацию VLSM часто называют «подсети на подсети».

Рассмотрим подсети, созданные путём заимствования трёх первых бит в хостовой порции адреса класса C 207.21.24.0

Подсеть	Адрес подсети
0	207.21.24.0/27
1	207.21.24.32/27
2	207.21.24.64/27
3	207.21.24.96/27
4	207.21.24.128/27
5	207.21.24.160/27
6	207.21.24.192/27
7	207.21.24.224/27

Мы получили восемь подсетей, каждая из которых может содержать не более 30 хостов.

Каждое соединение через последовательный интерфейс требует для себя два адреса и отдельной подсети. Использование для этого любой из подсетей /27 приведёт к потере адресов. Для создания подсети из двух адресов лучше всего подходит 30-ти битовая маска. Это как раз то, что надо для последовательного соединения. Разобьём одну из подсетей 207.21.24.192/27 на восемь подсетей, используя 30-ти битовую маску.

0	207.21.24.192/30
1	207.21.24.196/30
2	207.21.24.200/30
3	207.21.24.204/30
4	207.21.24.208/30
5	207.21.24.212/30
6	207.21.24.220/30
7	207.21.24.224/30

То есть каждую из оставшихся семи подсетей /27 можно использовать для адресации хостов в семи локальных сетях. Эти локальные сети можно связать в глобальную сеть с помощью не более чем восьми последовательных соединений из наших восьми сетей.

Чтобы в сетях с VLSM правильно осуществлялась маршрутизация маршрутизаторы должны обмениваться информацией о масках в подсетях.

Использование CIDR и VLSM не только предотвращает пустую трату адресов, но и способствует агрегации маршрутов или суммированию. Без суммирования маршрутов Интернет перестал бы развиваться уже в конце 90-х годов. Рисунок иллюстрирует как суммирование сокращает нагрузку на маршрутизаторы.

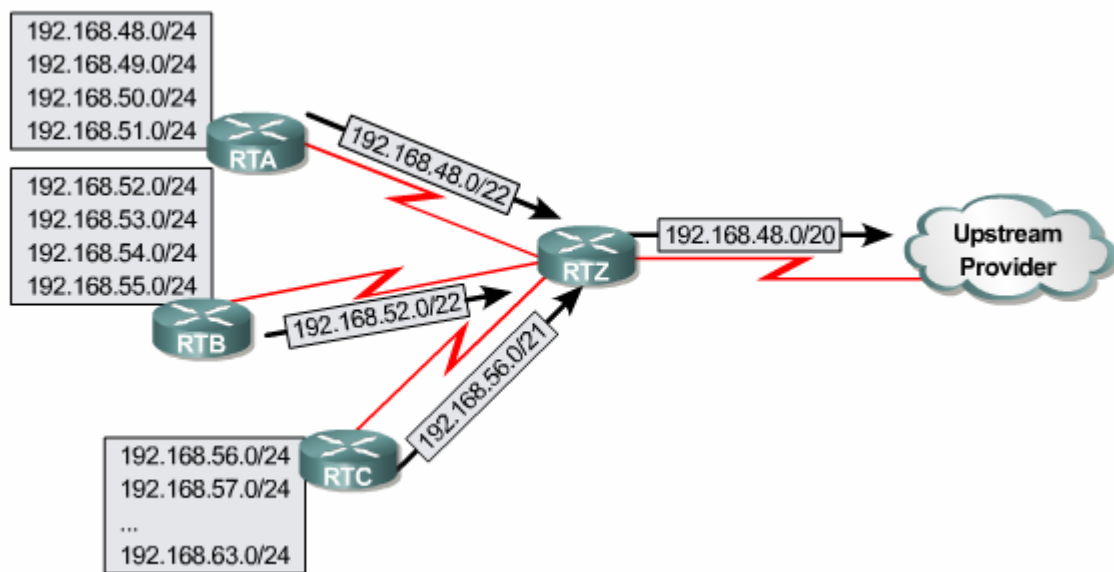


Рис. 1

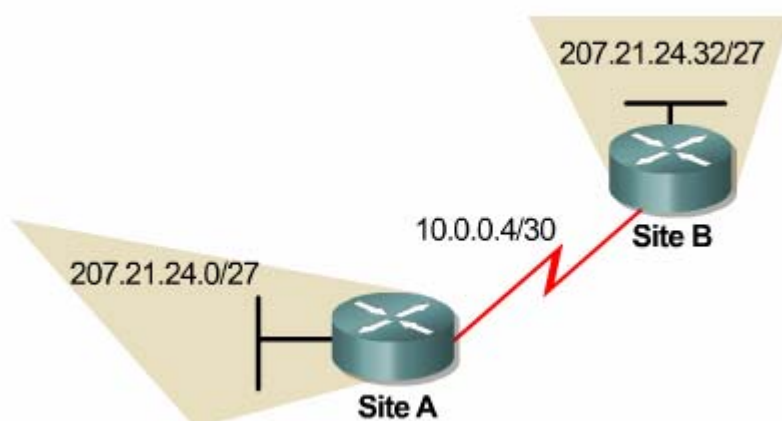


Рис. 2

Эта сложная иерархия сетей и подсетей суммируется в различных точках так, что вся сеть в целом выглядит извне как 192.168.48.0/20. Для правильной

работы суммирования маршрутов следует тщательно подходить к назначению адресов: суммируемые адреса должны иметь одинаковые префиксы.

Разорванные подсети

Разорванные подсети это сети из одной главной сети, разделённые сетью в совсем другом диапазоне адресов. Классовые протоколы маршрутизации RIP версии 1 и IGRP не поддерживают разрывные сети, так как маршрутизаторы не обмениваются масками подсетей. Если на рисунке 1 сайт А и сайт В работают на RIP версии 1, то сайт А будет получать от сайта В обновления маршрутной информации в сети 207.21.24.0/24, а не в сети 207.21.24.32/27.

Протоколы RIP v2 и EIGRP по умолчанию суммируют адреса на границах классов. Обычно такое суммирование желательно. Однако в случае разорванных подсетей не желательно. Отменить классическое автосуммирование можно командой no auto-summary.

Практическая часть

Конфигурируем VLSM и протестируем функциональность на двух протоколах маршрутизации RIP версии 1 и RIP версии 2.

Рассмотрим сеть класса С 192.168.1.0/24. Требуется выделить минимум по 25 адресов для двух локальных сетей и зарезервировать максимальное число адресов для дальнейшего развития.

Для поддержки 25 хостов в каждой подсети требуется минимум пять бит в восьмибитовой хостовой части адреса. Пять бит дадут максимум 30 возможных адресов хостов ($25 = 32 - 2$). Если пять бит должны быть использованы для хостов, то другие три бита в последнем октете адреса могут быть добавлены к 24-битовой маски нашей сети класса С. Следовательно, 27-битовая маска может быть использована для создания следующих 8 подсетей:

0	192.168.1.0/24	4	192.168.1.128/24
1	192.168.1.32/24	5	192.168.1.160/24
2	192.168.1.64/24	6	192.168.1.192/24
3	192.168.1.96/24	7	192.168.1.224/24

Для дальнейшей максимизации адресного пространства подсеть 192.168.1.0 /27 снова разделяется на 8 подсетей с использованием 30-битовой маски:

0	192.168.1.0/30	4	192.168.1.16/30
1	192.168.1.4/30	5	192.168.1.20/30
2	192.168.1.8/30	6	192.168.1.24/30
3	192.168.1.12/30	7	192.168.1.28/30

Эти подсети могут быть использованы для последовательных соединений точка-точка и минимизируют потери адресов, так как каждая подсеть содержит только два адреса.

Построим и настроим сеть, изображённую на рисунке 3. Для маршрутизатора Vista возьмём модель 2501, а для остальных двух модель 805. Используем коммутатор 2950.

1. Подымите интерфейсы и проверьте сеть командой `show cdp neighbors`.
2. Назначьте адреса согласно рисунку. Проверьте назначение командой `show ip interface brief`.
3. Для компьютер HostA имеем неоднозначность в назначении маршрута по умолчанию: либо на адрес 192.168.1.33 Ethernet интерфейса маршрутизатора SanJose1, либо на адрес 192.168.1.34 Ethernet интерфейса маршрутизатора SanJose2. Выберите произвольный, например
`hostA#ipconfig /dg 192.168.1.33`
4. На всех трёх маршрутизаторах настроим маршрутизацию по протоколу RIP с отключенным автосуммированием адресов. Это позволит прохождение информации о подсетях.

Router(config)# **router rip**

Router(config-router)#**version 2**

Router(config-router)# **no auto-summary**

Router(config-router)# **network 192.168.1.0**

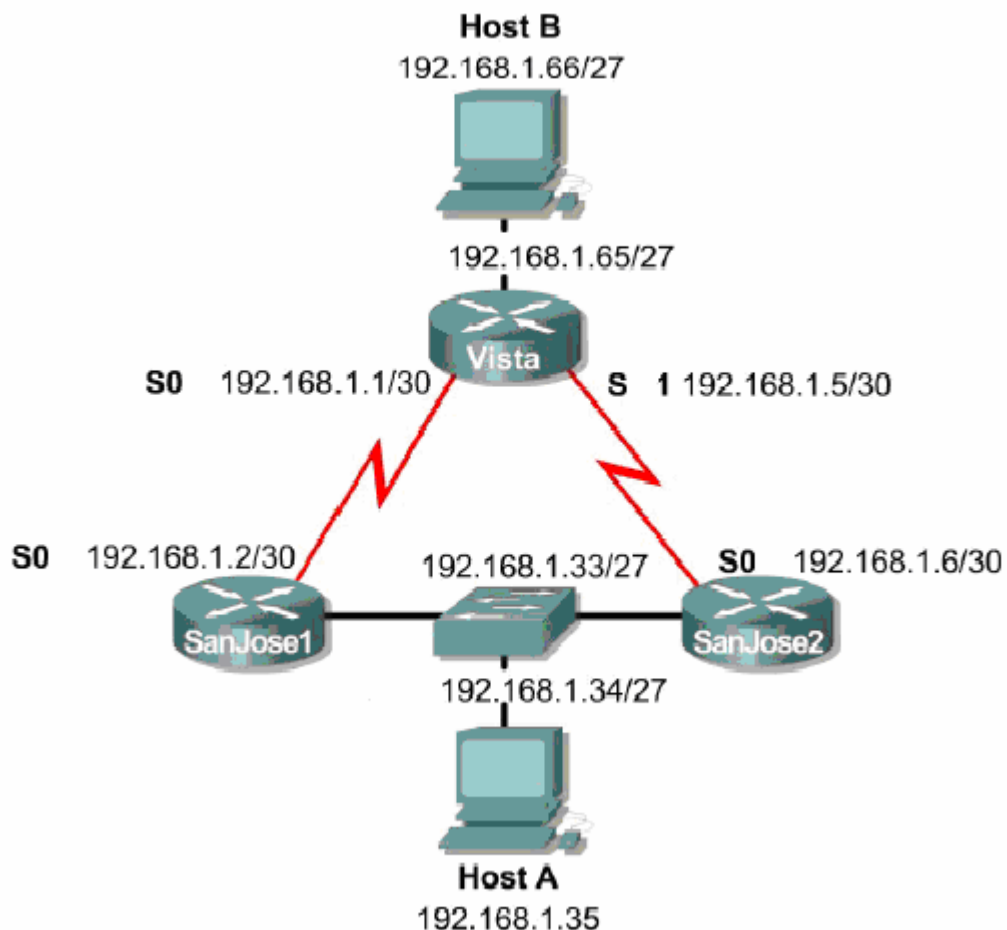


Рис.3

5. Должны увидеть, что на маршрутизаторе Vista есть маршрут на локальную сеть 192.168.1.32/27, где располагается компьютер hostA.

Vista#**show ip route**

```

C    192.168.1.64/27 is directly connected, Ethernet0
C    192.168.1.0/30 is directly connected, Serial0
C    192.168.1.4/30 is directly connected, Serial1
R    192.168.1.32/27 [120/1] via 192.168.1.2, 00:07:36, Serial0

```

Заметим, что в таблице выведен только один маршрут на сеть 192.168.1.32/27 – на адрес 192.168.1.2 через интерфейс Serial0, хотя есть ещё один маршрут на сеть 192.168.1.32/27 – через адрес 192.168.1.6 интерфейса Serial1. В реальном маршрутизаторе мы бы увидели оба эти маршрута. Симулятор выводит один маршрут, но обменивается маршрутной информацией по обоим интерфейсам Serial0 и Serial1:

Vista#debug ip rip

```

RIP: sending update to 255.255.255.255 via Serial0 (192.168.1.1)
      subnet 192.168.1.64, metric 1
      subnet 192.168.1.4, metric 1
      subnet 192.168.1.32, metric 2

RIP: sending update to 255.255.255.255 via Serial1 (192.168.1.5)
      subnet 192.168.1.64, metric 1
      subnet 192.168.1.0, metric 1

RIP: sending update to 255.255.255.255 via Ethernet0 (192.168.1.65)
      subnet 192.168.1.0, metric 1
      subnet 192.168.1.4, metric 1
      subnet 192.168.1.32, metric 2

RIP: received update from 192.168.1.2 on Serial0
      192.168.1.32 in 1 hops
      192.168.1.4 in 2 hops

RIP: received update from 192.168.1.6 on Serial1
      192.168.1.32 in 1 hops
      192.168.1.0 in 2 hops

```

Vista#no debug ip rip

Сделайте скриншоты.

- Должны увидеть, что на маршрутизаторе SanJose1 есть маршрут на локальную сеть 192.168.1.64/27, где располагается компьютер hostB.

```

C    192.168.1.0/30 is directly connected, Serial0
C    192.168.1.32/27 is directly connected, Ethernet0
R    192.168.1.64/27 [120/1] via 192.168.1.1, 00:07:40, Serial0
R    192.168.1.4/30 [120/1] via 192.168.1.34, 00:03:39, Ethernet0

```

Сделайте скриншот.

- Должны увидеть, что на маршрутизаторе SanJose2 также есть маршрут на локальную сеть 192.168.1.64/27, где располагается компьютер hostB.

```

C    192.168.1.4/30 is directly connected, Serial0
C    192.168.1.32/27 is directly connected, Ethernet0
R    192.168.1.64/27 [120/1] via 192.168.1.5, 00:08:16, Serial0
R    192.168.1.0/30 [120/1] via 192.168.1.5, 00:01:24, Serial0

```

Сделайте скриншот.

Пропингуйте компьютер hostA из компьютера hostB и наоборот. Сделайте 2 скриншота.

Заметим, что симулятор реализован так, что приведенный пример работает и без ввода команд `version 2` и `no auto-summary`.

Контрольные вопросы

1. Зачем нужна маска?
2. Что такое CIDR?
3. Что такое VLSM?
4. Как при использовании классов IP адресов в IP адресе выделяют адрес хоста и адрес подсети?
5. Как без использования классов IP адресов в IP адресе выделяют адрес хоста и адрес подсети?
6. Чему равно число доступных адресов в подсети?
7. По заданному преподавателем числу хостов в подсети определите минимальную маску.
8. Какие формы записи маски вы знаете?
9. Почему последовательное соединение выделяют в отдельную подсеть?
10. Какую маску рекомендуют использовать для сети последовательного соединения и почему?
11. Как CIDR и VLSM способствуют экономному использованию адресного пространства?
12. Что такое Supernetting?
13. Что такое агрегация маршрутов и как она способствует уменьшению таблиц маршрутов на маршрутизаторах?
14. Что такое разорванные подсети, и какие протоколы маршрутизации их не поддерживают?
15. Какие особенности работы симулятора при реализации протокола RIP?

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. Выполните в Boson задание для самостоятельной работы.
5. Предъявите преподавателю результат выполнения пункта 11 задания для самостоятельной работы.
6. Оформите отчёт. Содержание отчёта смотри ниже.
7. Защитите отчёт.

Задание для самостоятельной работы

Спроектируйте следующие четыре сети, согласно полученному варианту. См. рис. 4, 5, 6 и 7.

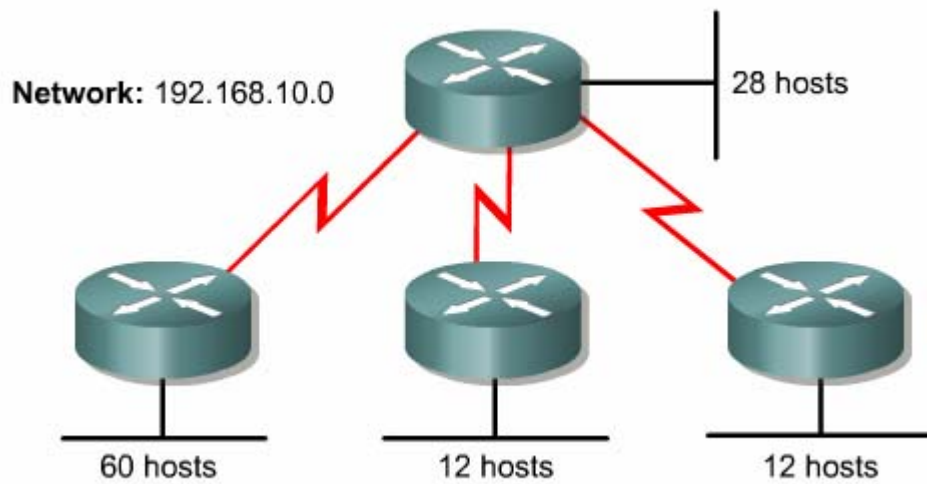


Рис 4. Вариант 1.

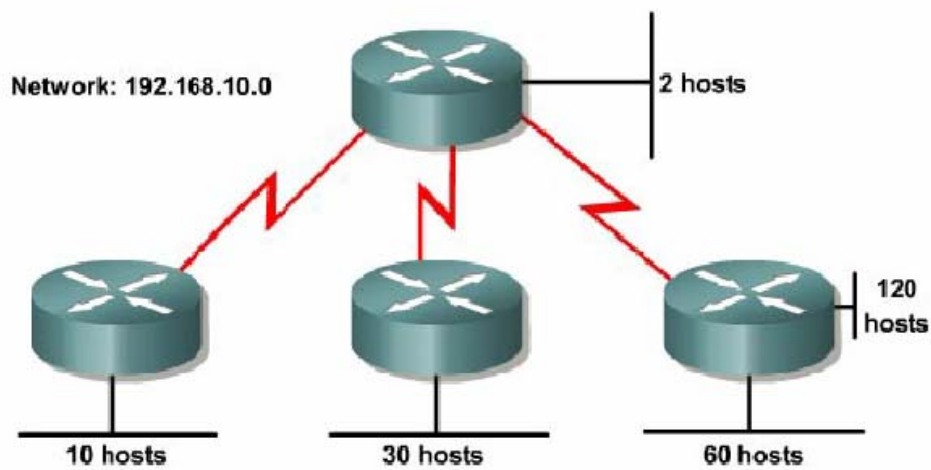


Рис 5. Вариант 2

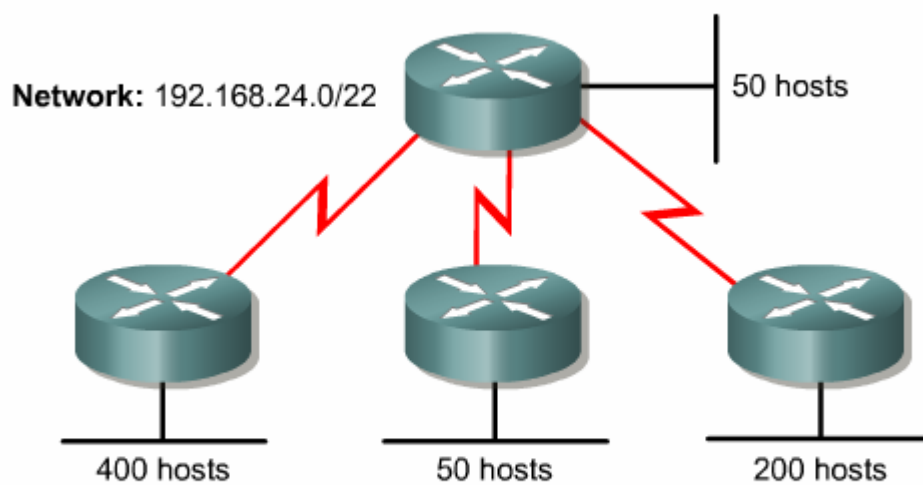


Рис 6. Вариант 3.

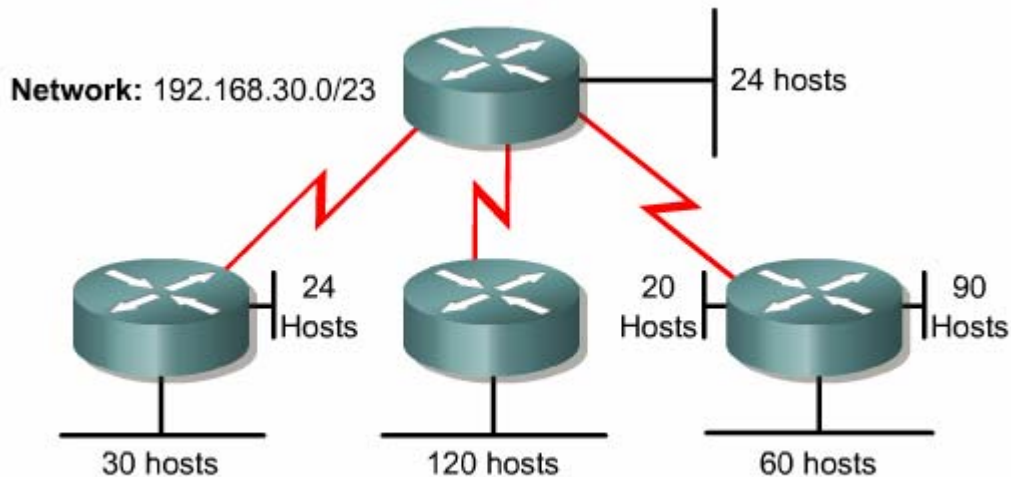
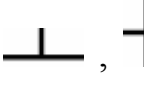



Рис 7. Вариант 4.

Порядок решения варианта

1. В дизайнере правильно подберите маршрутизаторы с нужным числом и типом интерфейсов. Используйте многослотовые устройства, имеющие наращиваемое число интерфейсов. В симуляторе они имеют адреса вида Ethernet 2/0, что означает интерфейс 0 в слоте 2.

2. Создайте топологию. Локальная сеть ,  представляется в виде коммутатора с одним компьютером



3. . Сделайте скриншот топологии.
4. Назначьте каждой подсети правильную маску с минимальным числом нулей. Помните, что маска для последовательного соединения равна /30.
5. Определитесь, какие адреса вы будете назначать на сетевые интерфейсы маршрутизаторов и компьютеров.
6. Нанесите в графическом редакторе на вашу топологию назначенные адреса и маски.
7. В симуляторе назначьте адреса на сетевые интерфейсы маршрутизаторов и компьютеров.
8. Проверьте правильность назначения адресов командами **show ip interface brief** (маршрутизатор) или **ipconfig** (компьютер).
9. Настройте на каждом маршрутизаторе динамическую маршрутизацию по протоколу RIP второй версии.
10. На каждом маршрутизаторе выведите таблицу маршрутизации и сделайте скриншот.
11. Вы должны пинговать из любого компьютера любой другой компьютер.

12. По желанию вы можете в маршрутизаторах использовать интерфейсы петля loopback для моделирования локальных сетей с последующей их заменой на Ethernet интерфейс, связывающий маршрутизатор с одним компьютером через коммутатор.

Содержание отчёта.

Отчёт готовится в электронном виде и распечатывается. Отчёт содержит

1. Скриншот топологии, созданной при выполнении практической части.
2. Конфигурации всех маршрутизаторов и компьютеров из rtr файлов, созданных при выполнении практической части.
2. Скриншот топологии вашего варианта с адресами и масками.
4. Конфигурации всех маршрутизаторов и компьютеров из rtr файлов, созданных при выполнении задания для самостоятельной работы.
5. Все скриншоты, указанные в задании для самостоятельной работы

Лабораторная работа №6. Списки управления доступом ACL

Теоретическая часть.

Список управления доступом (access control list ACL) это последовательный список правил, которые используются для разрешения или запрета потока пакетов внутри сети на основании информации, приведенной внутри списка. Без списка доступа все пакеты внутри сети разрешаются без ограничений для всех частей сети. Список доступа может быть использован для контроля распространения и получения информации об изменении таблиц маршрутов и, главное, для обеспечения **безопасности**. Политика безопасности в частности включает защиту от внешних атак, ограничения доступа между отделами организации и распределение загрузки сети.

Список доступа позволяет использовать маршрутизатор как межсетевой экран, брандмауэр, для запрета или ограничения доступа к внутренней сети из внешней сети, например, Интернет. Брандмауэр, как правило, помещаются в точках соединения между двумя сетями.

Стандартный ACL

При использовании стандартных ACL, единственным критерием для определения того, что пакет разрешен или запрещён, является IP адрес источника этого пакета. Формат элемента списка доступа следующий

Router(config)#access-list # permit | deny source-address source-mask,
где # – целое число – номер списка доступа, source-address обозначает адрес источника пакета, source-mask – маска в инверсной форме, накладываемая на адрес, permit – разрешить прохождение пакета, deny – запретить прохождение пакета. Число # определяет принадлежность элемента списка доступа к определённому списку доступа с номером #. Первая команда access-list определяет первый элемент списка доступа, вторая команда определяет второй элемент списка доступа и т.д. Маршрутизатор обрабатывает каждый определённый в нём список доступа по элементам сверху вниз. То есть, если адрес source-address пакета с учётом маски удовлетворяет условию элемента списка, то дальнейшие элементы списка маршрутизатор не обрабатывает. Следовательно, для избежания лишней обработки, элементы, определяющие более общие условия, следует помещать в начале списка. Внутри маршрутизатора может быть определено несколько списков доступа. Номер стандартного списка должен лежать в диапазоне 1 – 99. Маска в списке доступа задаётся в инверсной форме, например маска 255.255.0.0 выглядит как 0.0.255.255.

Маршрутизаторы Cisco предполагают, что все адреса, не упомянутые в списке доступа в явном виде, запрещены. То есть в конце списка доступа присутствует невидимый элемент

Router(config)#access-list # deny 0.0.0.0 255.255.255.255

Так, если мы хотим разрешить только трафик от адреса 1.1.1.1 и запретить весь остальной трафик достаточно в список доступа поместить один элемент

Router(config)#access-list 77 permit 1.1.1.1 0.0.0.0.

Здесь предполагается, что мы организовали список доступа с номером 77.

Рассмотрим возможность применения стандартных списков доступа для диапазона адресов. Возьмём к примеру диапазон 10.3.16.0 – 10.3.31.255. Для получения инверсной маски можно вычесть из старшего адреса младший и получить 0.0.15.255. Тогда пример элемента списка можно задать командой

```
Router(config)#access-list 100 permit 10.3.16.0 0.0.15.255
```

Для того, чтобы список доступа начал выполнять свою работу он должен быть применен к интерфейсу с помощью команды

```
Router(config-if)#ip access-group номер-списка-доступа in либо out
```

Список доступа может быть применён либо как входной (in) либо как выходной (out). Когда вы применяете список доступа как входной, то маршрутизатор получает входной пакет и сверяет его входной адрес с элементами списка. Маршрутизатор разрешает пакету маршрутизироваться на интерфейс назначения, если пакет удовлетворяет разрешающим элементам списка либо отбрасывает пакет, если он соответствует условиям запрещающих элементов списка. Если вы применяете список доступа как выходной, то роутер получает входной пакет, маршрутизирует его на интерфейс назначения и только тогда обрабатывает входной адрес пакета согласно элементам списка доступа этого интерфейса. Далее маршрутизатор либо разрешает пакету покинуть интерфейс либо отбрасывает его согласно разрешающим и запрещающим элементам списка соответственно. Так, созданный ранее список с номером 77 применяется к интерфейсу Ethernet 0 маршрутизатора как входной список командами

```
Router(config)#int Ethernet 0
```

```
Router(config-if)#ip access-group 77 in
```

Этот же список применяется к интерфейсу Ethernet 0 маршрутизатора как выходной список с помощью команд

```
Router(config-if)#ip access-group 77 out
```

Отменяется список на интерфейсе с помощью команды **no**

```
Router(config-if)# no ip access-group 77 out
```

Приступим к созданию более сложных списков доступа. Рассмотрим сеть на рисунке 1. Разрешим все пакеты, исходящие из сети 10.1.1.0 /25 (10.1.1.0 255.255.255.128) , но запретим все пакеты, исходящие из сети 10.1.1.128 /25 (10.1.1.128 255.255.255.128). Мы также хотим запретить все пакеты, исходящие из сети 15.1.1.0 /24 (15.1.1.0 255.255.255.0), за исключением пакетов от единственного хоста с адресом 15.1.1.5. Все остальные пакеты мы разрешаем. Списку дадим номер 2. Последовательность команд для выполнения поставленной задачи будет следующая

```
Router(config)#access-list 2 deny 10.1.1.128 0.0.0.127
```

```
Router(config)#access-list 2 permit 15.1.1.5 0.0.0.0
```

```
Router(config)#access-list 2 deny 15.1.1.0 0.0.0.255
```

```
Router(config)#access-list 2 permit 0.0.0.0 255.255.255.255
```

Отметим отсутствие разрешающего элемента для сети 10.1.1.0 255.255.255.128. Его роль выполняет последний элемент **access-list 2 permit 0.0.0.0 255.255.255.255**.

Удостоверимся, что мы выполнили поставленную задачу.

1. Разрешить все пакеты, исходящие из сети 10.1.1.0 255.255.255.128.

Последняя строка в списке доступа удовлетворяет этому критерию. Нет необходимости в явном виде разрешать эту сеть в нашем списке доступа так, как в списке нет строк, соответствующей этой сети за исключением последней разрешающей строки `permit 0.0.0.0 255.255.255.255`.

2. Запретить все пакеты, исходящие из сети 10.1.1.128 255.255.255.128.

Первая строка в списке выполняет этот критерий. Важно отметить вид инверсной маски 0.0.0.127 для этой сети. Эта маска указывает, что мы не должны брать в рассмотрение последние семь бит четвертого октета адреса, которые назначены для адресации в данной подсети. Маска для этой сети 255.255.255.128, которая говорит, что последние семь бит четвертого октета определяют адресацию в данной сети.

3. Запретить все пакеты, исходящие из сети 15.1.1.0 255.255.255.0, за исключением пакетов от единственного хоста с адресом 15.1.1.5

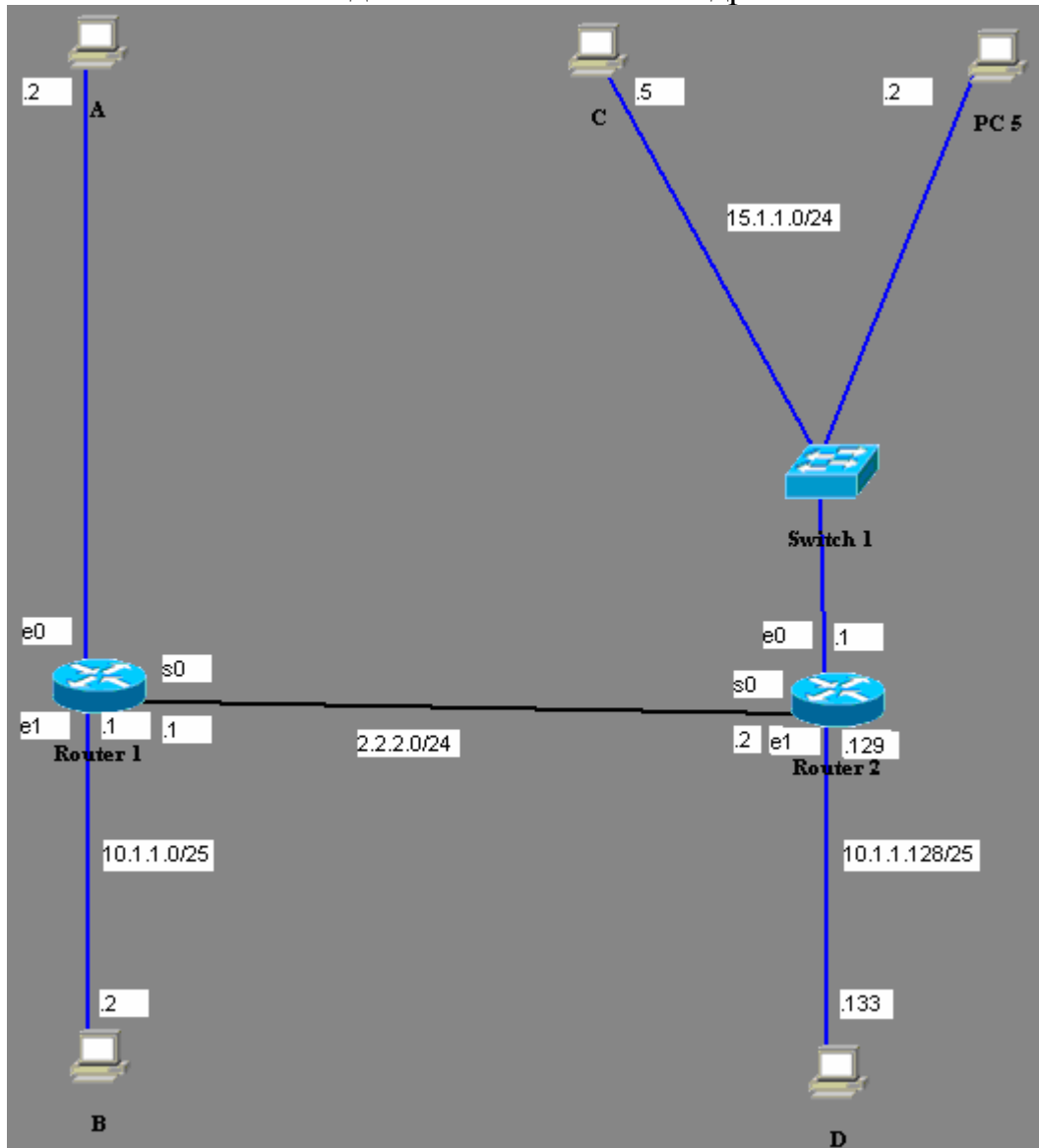


Рис. 1.

Это требование удовлетворяется второй и третьей строкой нашего списка доступа. Важно отметить, что список доступа осуществляет это требование не в том порядке как оно определено. Обязательно следует помнить, что список доступа обрабатывается сверху вниз и при первом совпадении обработка пакетов прекращается. Мы вначале требуем запретить все пакеты, исходящие из сети 15.1.1.0 255.255.255.0 и лишь затем разрешить пакеты с адресом 15.1.1.5. Если в командах, определяющих список доступа мы, переставим вторую и третью команды, то вся сеть 15.1.1.0 будет запрещена до разрешения хоста 15.1.1.5. То есть, адрес 15.1.1.5 сразу же в начале будет запрещён более общим критерием deny 15.1.1.0 0.0.0.255.

4. Разрешить все остальные пакеты

Последняя команда разрешает все адреса, которые не соответствуют первым трем командам.

Таким образом, имеем следующую последовательность действий для воплощения списка доступа.

1. Определить критерии и ограничения для доступа.
2. Воплотить их с помощью команд access-list, создав список доступа с определённым номером.
3. Применить список к определённому интерфейсу либо как входящий, либо как исходящий.

Остановимся на последнем пункте. В общем случае стандартный список доступа следует помещать как можно ближе к точке назначения, а не к источнику пакетов. Хотя могут быть исключения. Так как стандартный список доступа работает только с исходными адресами, то не всегда возможна детальная конфигурация. Требуется приложить усилия, чтобы избежать возникновения не желаемых конфигураций доступа. Если список помещён вблизи источника пакетов, то очень вероятно, что доступ к устройствам, на которых не осуществляется никакая конфигурация доступа, будет затруднён.

Конкретизируем политику безопасности для сети на рисунке 1. Наша цель создать политику для компьютера А (адрес 1.1.1.2 сеть 1.1.1.0/24), которая из всех устройств локальной сети 15.1.1.0 /24 в которую входит компьютер С (15.1.1.5) разрешит доступ к компьютеру А лишь самого компьютера С. Мы также хотим создать политику, запрещающую удалённый доступ к компьютеру А из любого устройства локальной сети 10.1.1.128 / 25 компьютера D (10.1.1.133). Весь остальной трафик мы разрешаем. На рисунке 1 компьютер PC5 (15.1.1.5) играет роль произвольного отличного от компьютера С представителя локальной сети 15.1.1.0/24.

Размещение списка критично для воплощения такой политики. Возьмём созданный ранее список с номером 2. Если список сделать выходным на последовательном интерфейсе маршрутизатора 2, то задача для компьютера А будет выполнена, однако возникнут ограничения на трафик между другими локальными сетями. Аналогичную ситуацию получим, если сделаем этот список входным на последовательном интерфейсе маршрутизатора 1. Если мы поместим этот список как выходной на Ethernet А интерфейс маршрутизатора 1, то задача будет выполнена безо всяких побочных эффектов.

Расширенные ACL

Со стандартным ACL вы можете указывать только адрес источника, а маска не обязательна. В расширенных ACL вы должны указать и адрес приёмника и адрес источника с масками. Можете добавить дополнительную протокольную информацию для источника и назначения. Например, для TCP и UDP разрешено указывать номер порта, а для ICMP разрешено указывать тип сообщения. Как и для стандартных ACL, можно с помощью опции log осуществлять лог.

Общая форма команды для формирования строки списка расширенного доступа

access-list access-list-number {permit | deny} protocol source source-wildcard [operator source-port] destination destination-wildcard [operator destination-port] [precedence precedence-number] [tos tos] [established] [log | log-input],

где access-list-number -100-199|2000-2699, protocol - ip, icmp, tcp, gre, udp, igmp, eigrp, igmp, ipinip, nos и ospf. Для порта source-port или destination-port можно использовать номер порта или его обозначение bgp, chargen, daytime, discard, domain, echo, finger, ftp, ftp-data, gopher, hostname, irc, klogin, kshell, lpd, nntp, pop2, pop3, smtp, sunrpc, syslog, tacacs-ds, talk, telnet, time, uucp, whois и www. Operator это eq (равно), neq (не равно), gt (больше чем), lt (меньше чем), range (указывается два порта для определения диапазона).

Как и для стандартных ACL расширенный ACL следует привязать к интерфейсу либо для входящего на интерфейс трафика

Router(config-if)# ip access-group #ACL in

либо для выходящего из интерфейса трафика

Router(config-if)# ip access-group #ACL out

здесь #ACL - номер списка.

Примеры элементов расширенного ACL

Разрешить SMTP отовсюду на хост

Router(config)# access-list 111 permit tcp any host 172.17.11.19 eq 25

Разрешить телнет отовсюду на хост

Router(config)# access-list 111 permit tcp any host 172.17.11.19 eq 23

Расширенный ACL позволяет очень тонко настроить права доступа.

Именованные ACL

К именованным ACL обращаются по имени, а не по номеру, что даёт наглядность и удобство для обращения. Для создания именованного ACL имеется команда

Router(config)# ip access-list extended ACL_name

и далее команды для создания элементов списка именно этого именованного списка

Router(config-ext-nacl)# permit|deny IP_protocol source_IP_address wildcard_mask [protocol_information] destination_IP_address wildcard_mask [protocol_information] [log]

Для завершения создания списка следует дать команду exit.

Имя именованного списка чувствительно к регистру. Команды для создания неименованного списка аналогичные командам для создания элементов нумерованного списка, но сам процесс создания отличен. Вы должны использовать ключевое слово `ip` перед главным ACL оператором и тем самым войти в режим конфигурации именно для этого именованного списка. В этом режиме вы начинаете с ключевых слов `permit` или `deny` и не должны вводить `access-list` в начале каждой строки.

Привязка именованных ACL к интерфейсу осуществляется командой

Router(config)# interface type [slot_#]port_#

Router(config-if)# ip access-group ACL_name in|out

ACL обрабатываются сверху вниз. Наиболее часто повторяющийся трафик должен быть обработан в начале списка. Как только обрабатываемый списком пакет удовлетворяет элементу списка, обработка этого пакета прекращается. Стандартные ACLs следует помещать ближе к точке назначения, где трафик должен фильтроваться. Выходные (out) расширенные ACLs следует помещать как можно ближе к источнику фильтруемых пакетов, а входные следует помещать ближе к точке назначения, где трафик должен фильтроваться.

Именованные ACLs разрешает вам себя редактировать. Для этого надо набрать команду, которая была использована для его создания

Router(config)# ip access-list extended ACL_name

С помощью клавиш с вертикальными стрелками найти строку списка, которую вы хотите изменить. Изменить её, используя горизонтальные стрелки. Нажать ввод. Новая строка добавится в конец списка. Старая не уничтожится. Для её уничтожения следует ввести `no` в начале строки.

Для редактирования числовых ACLs следует его уничтожить и создать заново или изменить список офлайн и загрузить в устройство с помощью TFTP (в симуляторе из `rtg` файла).

Практическая часть.

1. Загрузим в симулятор топологию, изображённую на рисунке 2.



Рис. 2.

Назначим адреса интерфейсам (маска 255.255.255.240) согласно таблице. Не забудьте на DCE устройстве последовательного соединения задать синхронизацию.

	Router2	Router1	Router4
Ethernet	24.17.2.2	24.17.2.1	
Serial		24.17.2.17	24.17.2.18

Осуществим конфигурацию RIP маршрутизации

Для Router1

Router1(config)#router rip

Router1(config-router)#network 24.0.0.0

На Router2

```
Router2(config)#router rip
```

```
Router2(config-router)#network 24.0.0.0
```

и на Router4

```
Router3(config)#router rip
```

```
Router3(config-router)#network 24.0.0.0
```

Проверьте свою сеть с помощью команды ping и, в частности, что вы можете пинговать интерфейс Ethernet0 (24.17.2.2) маршрутизатора 2 из маршрутизатора 4

```
Router4#ping 24.17.2.2
```

Создадим стандартный список доступа, который не позволит пинговать маршрутизатор 2 из маршрутизатора 4. Для этого блокируем единственный адрес 24.17.2.18 маршрутизатора 4 и разрешим остальной трафик. Список создадим на маршрутизаторе 2 командами

```
Router2(config)#access-list 1 deny 24.17.2.18 0.0.0.0
```

```
Router2(config)#access-list 1 permit 0.0.0.0 255.255.255.255
```

Применим список к интерфейсу Ethernet маршрутизаторе 2

```
Router2(config)#interface Ethernet0
```

```
Router2(config-if)#ip access-group 1 in
```

Проверим, что список доступа запущен. Для этого посмотрим работающую конфигурацию

```
Router2#show running-config
```

Мы также можем видеть, что список применён к интерфейсу, используя команду “show ip interface”. Найдите в выводимой информации с строку “Innbound access list is 1”.

```
Router2#show ip interface
```

Команда “show access-lists” покажет нам содержимое созданного списка доступа.

```
Router2#show access-lists
```

```
Standard IP access list 1
```

```
1 deny host 24.17.2.18 (5 matches)
```

```
1 permit 0.0.0.0 255.255.255.255 (35 matches)
```

Отметим, что host 24.17.2.18 равносильно 24.17.2.18 0.0.0.0. Теперь при попытке пинговать интерфейс Ethernet0 (24.17.2.2) роутера 2 из роутера 4

```
Router4#ping 24.17.2.2
```

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

```
Sending 5, 100-byte ICMP Echos to 24.17.2.2, timeout is 2 seconds:
```

```
UUUUU
```

```
Success rate is 0 percent (0/5), round-trip min/avg/max = 1/2/4 ms
```

Получим строку “UUUUU”, которая означает, что список доступа работает корректно.

2. Создадим и загрузим в симулятор топологию на рисунке 2,

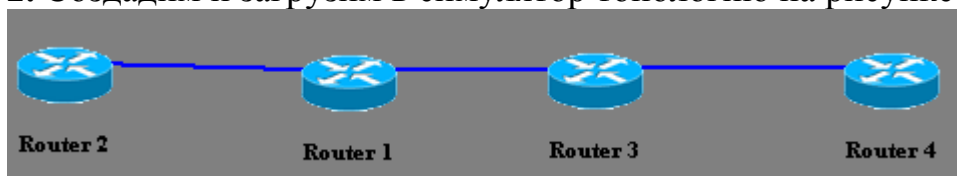


Рис. 3.

Назначим адреса интерфейсам (маска 255.255.255.0) согласно таблице

	Router2	Router1	Router3	Router4
Ethernet 0	160.10.1.2	160.10.1.1	175.10.1.2	180.10.1.2
Ethernet 1		175.10.1.1	180.10.1.1	

Осуществим конфигурацию OSPF маршрутизации

Для Router1

```
Router1(config)#router ospf 1
```

```
Router1(config-router)#network 160.10.1.0 0.0.0.255 area 0
```

```
Router1(config-router)#network 175.10.1.0 0.0.0.255 area 0
```

Для Router2

```
hostname router2
```

```
interface Ethernet0
```

```
ip address 160.10.1.2 255.255.255.0
```

```
Router2(config)#router ospf 1
```

```
Router2(config-router)#network 160.10.1.0 0.0.0.255 area 0
```

```
end
```

Для Router3

```
Router3(config) #router ospf 1
```

```
Router3(config-router)#network 175.10.1.0 0.0.0.255 area 0
```

```
Router3(config-router)#network 180.10.1.0 0.0.0.255 area 0
```

Для Router4

```
Router4(config) #router ospf 1
```

```
Router4(config-router)#network 180.10.1.0 0.0.0.255 area 0
```

Для проверки пропингуйте крайние точки

```
router2#ping 180.10.1.2
```

```
router4#ping 160.10.1.2
```

Создадим стандартный список доступа для фильтрации трафика, приходящего на интерфейс ethernet0 1-го маршрутизатора router1 и разрешает трафик от подсети 175.10.1.0 (router3) и блокирует трафик от других устройств.

```
router1(config)#access-list 1 permit 175.10.1.0 0.0.0.255
```

Проверьте, что он создан

```
router1#show access-list
```

```
Standard IP access list 1
```

```
1 permit 175.10.1.0 0.0.0.255 (15 matches)
```

Присоедините список как входной к интерфейсу Ethernet 1

```
router1(config)#interface Ethernet1
```

```
router1(config-if)#ip access-group 1 in
```

Проверьте присоединение командой

```
router1# show runnig-config
```

Проверьте связь между 3 и 2 маршрутизаторами и между 4 и 2 .

```
router3# ping 160.10.1.2
```

```
router4# ping 160.10.1.2
```

Связь между 3 и 2-м роутерами должна быть, а между 4 и 2 - нет.

Изменим список доступа и разрешим трафик от подсети 180.10.1.0 (router3) и блокирует трафик от других устройств.

```
router1(config)# no access-list 1
router1(config)# access-list 1 permit 180.10.1.0 0.0.0.255
```

Проверьте, что он изменился

```
router1#show access-list
Standard IP access list 1
  1 permit 180.10.1.0 0.0.0.255 (0 matches)
```

Проверьте связь между 3 и 2 маршрутизаторами и между 4 и 2 .

```
router3# ping 160.10.1.2
```

```
router4# ping 160.10.1.2
```

Связь между 4 и 2-м роутерами должна быть, а между 3 и 2 - нет.

3. Осуществите и проверьте конфигурацию IP для сети на рисунке 1 и примените OSPF для организации динамической маршрутизации.

Для маршрутизатора router 1

```
router1(config)#router ospf 1
router1(config-router)#network 2.2.2.0 0.0.0.255 area 0
router1(config-router)#network 1.1.1.0 0.0.0.255 area 0
router1(config-router)#network 10.1.1.0 0.0.0.127 area 0
```

Для маршрутизатора router 2

```
Router2(config)#router ospf 1
Router2(config-router)#network 10.1.1.128 0.0.0.127 area 0
Router2(config-router)#network 15.1.1.0 0.0.0.255 area 0
Router2(config-router)#network 2.2.2.0 0.0.0.255 area 0
```

Проверте работоспособность сети: вы должны из любого устройства пинговать любой интерфейс. Или проще: все компьютеры А, В, С, D, PC5 должны взаимно попарно пинговаться.

Создадим список доступа из теоретической части

3.1 На маршрутизатора router 1 создадим список доступа

```
router1(config)#access-list 2 deny 10.1.1.128 0.0.0.127
router1(config)#access-list 2 permit host 15.1.1.5
router1(config)#access-list 2 deny 15.1.1.0 0.0.0.255
router1(config)#access-list 2 permit 0.0.0.0 255.255.255.255
```

и применим его к интерфейсу Ethernet0 как выходной

```
router1(config)#interface Ethernet0
router1(config-if)#ip access-group 2 out
```

Создать скриншот результата выполнения команды

```
router1#show access-list
```

Попарно пропингуем А, В, С, PC5, D. В результате должна получиться следующая матрица доступа

Из\В	А	В	С	PC5	D
А	+	+	+	-	-
В	+	+	+	+	+
С	+	+	+	+	+
PC5	-	+	+	+	+
D	-	+	+	+	+

Таблица 1

Видим, что политика безопасности из теоретической части полностью реализована.

3.2 Удалим ACL с интерфейса e0 и применим как входной к интерфейсу s0

```
router1(config)#interface Ethernet0
```

```
router1(config-if)#no ip access-group 2 out
```

```
router1(config-if)#int s0
```

```
router1(config-if)#ip access-group 2 in
```

Попарно пропингуем А, В, С, PC5, D. В результате должна получиться следующая матрица доступа

Из\В	А	В	С	PC5	D
А	+	+	+	-	-
В	+	+	+	-	-
С	+	+	+	+	+
PC5	-	-	+	+	+
D	-	-	+	+	+

Таблица 2

Видим, что теперь трафик между сетями 10.1.1.0/25 и 10.1.1.128/25 запрещен. Невозможен также трафик между сетью 10.1.1.0/25 и сетью 15.1.1.0/24 за исключением компьютера С с адресом 15.1.1.5.

4. Используем топологию и конфигурацию пункта 3 этой лабораторной работы

Отменим конфигурацию доступа, сделанную в пункте 1

```
Router2(config)#no access-list 1 deny 24.17.2.18 0.0.0.0
```

```
Router2(config)#no access-list 1 permit 0.0.0.0 255.255.255.255
```

Применим список к интерфейсу Ethernet маршрутизаторе 2

```
Router2(config)#interface Ethernet0
```

```
Router2(config-if)# no ip access-group 1 in
```

Разрешим заходить на router1 телнетом на его два интерфейса с паролем router1

```
Router1(config)#line vty 0 4
```

```
Router1(config-line)#login
```

```
Router1(config-line)password router1
```

Наши EACL будут делать пару различных вещей. Первое мы разрешим только телнет из подсети последовательного соединения 24.17.2.16/240 для входа на router1

```
router1(conf)#access-list 101 permit tcp 24.17.2.16 0.0.0.15 any eq telnet log
```

Опция log заставит маршрутизатор показывать выход при срабатывании списка доступа.

Разрешим на маршрутизаторе router1 весь трафик из Ethernet 0 подсети 24.17.2.0/240

```
router1(conf)#access-list 102 permit ip 24.17.2.0 0.0.0.15 any
```

Проверим установку списков

```
router1#show access-list
```

```
Extended IP access list 101
  permit tcp 24.17.2.16 0.0.0.15 any eq telnet log (0 matches)
Extended IP access list 102
  permit ip 24.17.2.0 0.0.0.15 any log (1 matches)
```

Теперь применим списки к интерфейсам для входящих пакетов

```
router1(conf)# interface Serial0
router1(conf-if)# ip access-group 101 in
router1(conf-if)# interface Ethernet0
router1(conf-if)# ip access-group 102 in
```

Для проверки, что EACL присутствуют на интерфейсах, используйте команду

```
router1#show running-config
```

или

```
router1#show ip interface
```

Проверим функционирования EACL. Присоединимся к router4 и попытаемся безуспешно пропинговать интерфейс Serial0 на router1

```
router4#ping 24.17.2.17
```

EACL номер 101 блокировал ping. Но должен разрешить telnet

```
router4#telnet 24.17.2.17
```

Успешно. Введём пароль router1. Промпт router4# изменился на router1>. Нажав одновременно ctrl-shift-6 и затем 6, вернёмся на router4. О срабатывании EACL 101 на router1 нам укажет лог

```
00:06:50: %SEC-6-IPACCESSLOGDP: list 101 permitted TCP 24.17.2.18 -> 24.17.2.17 (8/0), 5 packets
```

Посмотрим номер сессии и убьём телнет соединение

```
router4#show sess
```

```
router4#disconnect 1
```

Присоединимся к router2 и посмотрим, можем ли мы пропинговать интерфейс Serial0 на router4.

```
Router2# ping 24.17.2.18
```

Почему неудачно? Пакет стартует в Router2, идёт через Router1 (о срабатывании EACL 102 на router1 нам укажет лог

```
00:03:29: %SEC-6-IPACCESSLOGDP: list 102 permitted IP 24.17.2.2 -> 24.17.2.18 (8/0), 5 packets
```

) и приходит на Router4. На Router4 он переформируется и отсылается обратно к Router1. Когда Router4 переформирует пакет, адрес источника становится адресом приёмника и адрес приёмника становится адресом источника. Когда пакет приходит на интерфейс Serial0 на router1 он отвергается, так как его адрес источника равен IP адресу интерфейса Serial0 на router4 24.17.2.17, а здесь разрешён лишь tcp.

Присоединимся к router2 и посмотрим, можем ли мы пропинговать интерфейс Ethernet0 на router1.

```
router2#ping 24.17.2.1
```

Успешно. Аналогично и для телнета

```
router2#telnet 24.17.2.1
```

EACL работают успешно. О срабатывании EACL 102 на router1 нам укажет лог.

```
00:05:22: %SEC-6-IPACCESSLOGDP: list 102 permitted IP 24.17.2.2 -> 24.17.2.1 (8/0), 5 packets
```

Заметим, что лог так же постоянно показывает RIP обновления

```
00:06:42: %SEC-6-IPACCESSLOGDP: list 102 permitted IP 24.17.2.2 -> 255.255.255.255 (8/0), 5 packets
00:06:12: %SEC-6-IPACCESSLOGDP: list 102 permitted IP 24.17.2.2 -> 255.255.255.255 (8/0), 5 packets
00:07:42: %SEC-6-IPACCESSLOGDP: list 102 permitted IP 24.17.2.2 -> 255.255.255.255 (8/0), 5 packets
00:07:12: %SEC-6-IPACCESSLOGDP: list 102 permitted IP 24.17.2.2 -> 255.255.255.255 (8/0), 5 packets
```

5. Именованные ACL

Отменим на router1 привязку EACL к интерфейсам

```
router1(conf)# interface Serial0
router1(conf-if)# no ip access-group 101 in
router1(conf-if)# interface Ethernet0
router1(conf-if)#no ip access-group 102 in
и отменим на router1 EACL
router1(conf)#no access-list 101
router1(conf)#no access-list 102
```

Поставим задачу запретить по всей сети лишь пинги от router4 на router2. Список доступа можно расположить и на router1 и на router2. Хотя рекомендуют располагать ACL ближе к источнику (для сокращения трафика), в этом примере расположим именованный список с именем deny_ping на router2.

```
router2(config)#ip access-list extended deny_ping
router2(config-ext-nacl)#deny icmp 24.17.2.18 0.0.0.0 24.17.2.2 0.0.0.0 log
router2(config-ext-nacl)# permit ip any any log
```

Первая команда указывает, что мы создаём именованный расширенный список доступа с именем deny_ping. Вторая команда указывает на запрещение ICMP трафика с адресом источника строго 24.17.2.18 и адресом приёмника строго 24.17.2.2. Третья команда разрешает остальной IP трафик.

Проверим создание списка

```
router2#show access-list
Extended IP access list deny_ping
deny icmp host 24.17.2.18 host 24.17.2.2 log (0 matches)
permit ip any any log (0 matches)
```

Всё правильно, мы видим в первой строке просто другую форму представления команды deny icmp 24.17.2.18 0.0.0.0 24.17.2.2 0.0.0.0 log.

Применим список для входного трафика интерфейса Ethernet0 на router2

```
Router2(conf)#interface Ethernet0
Router2(conf-if)#ip access-group deny_ping in
```

Присоединимся к router4 и пропингуем роутер2

```
router4#ping 24.17.2.2
```

Неудача. Присоединимся к router1 и пропингуем роутер2

```
Router1#ping 24.17.2.2
```

Успех. Присоединимся к router2 и посмотрим на два отдельных лог-сообщения: первое о запрещении пинга от router4 и второе о разрешении пинга от router1

```
00:11:18: %SEC-6-IPACCESSLOGDP: list 0 permitted IP 24.17.2.1 -> 24.17.2.2 (8/0), 5 packets
00:12:30: %SEC-6-IPACCESSLOGDP: list 0 permitted IP 24.17.2.1 -> 255.255.255.255 (8/0), 5 packets
```

6. Рассмотрим более сложные вопросы расширенных списков доступа. Создадим топологию

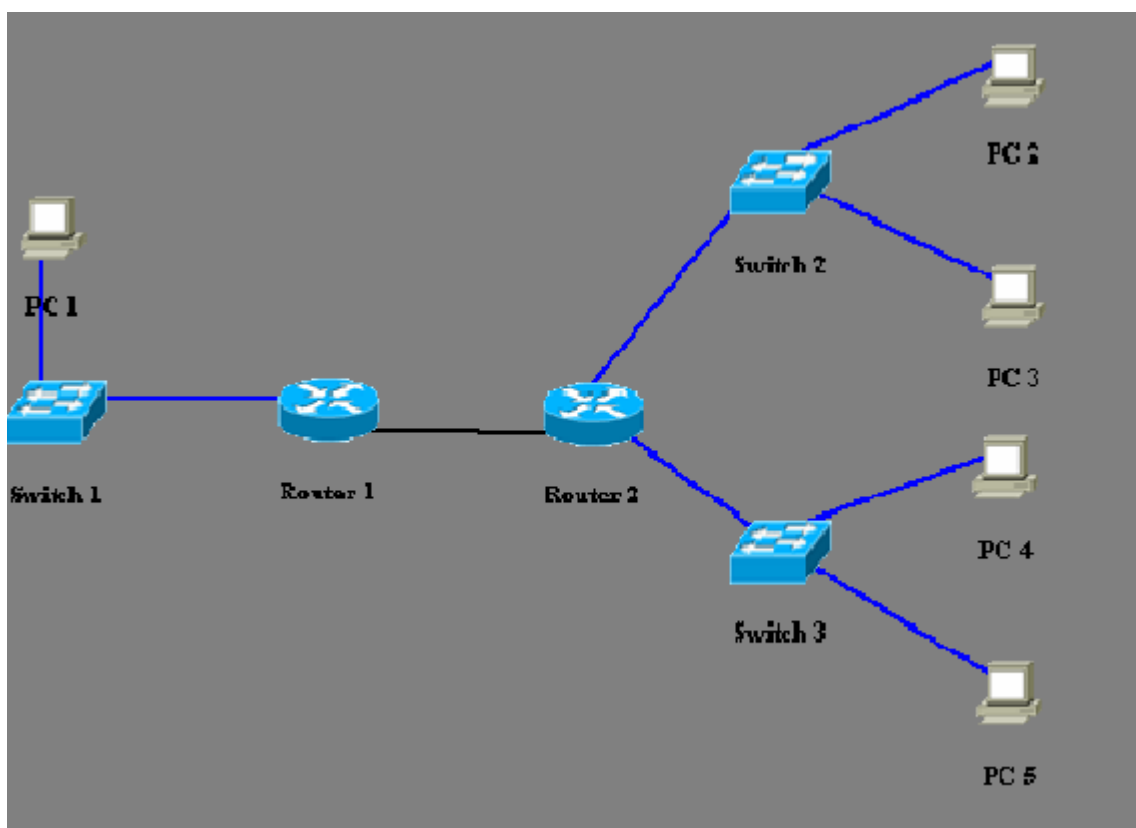


Рис. 4.

Используйте коммутаторы модели 1912. Маршрутизатор Router1 - модели 805. Маршрутизатор Router2 - модели 1605.

Назначим IP адреса маршрутизаторам

	Router1	Router2
Fa0/0	1.1.3.1/24	1.1.1.129/25
Fa0/1		1.1.1.1/25
Serial0	1.1.2.1/24	1.1.2.2/24

и компьютерам

Hostname	IP на ethernet0	Шлюз
PC1	1.1.3.2 255.255.255.0	1.1.3.1
PC2	1.1.1.130 255.255.255.128	1.1.1.129
PC3	1.1.1.131 255.255.255.128	1.1.1.129
PC4	1.1.1.2 255.255.255.128	1.1.1.1
PC5	1.1.1.3 255.255.255.128	1.1.1.1

На Router1 и Router2 конфигурируем RIP

Router(config)#**router rip**

Router(config-router)#**network 1.0.0.0**

Интерфейсы всех устройств должны пинговаться со всех устройств.

6.1. Список доступа сеть-сеть.

Создадим список, который разрешает трафик от локальной сети компьютеров PC4 и PC5 в локальную сеть компьютера PC1 и запрещает трафик от локальной сети компьютеров PC2 и PC3 в локальную сеть компьютера PC1.

Так как трафик приходит от router2 к router1, то следует поместить список доступа на интерфейс serial0 router1 для входного трафика

```
Router1(conf)#access-list 100 permit ip 1.1.1.0 0.0.0.127 1.1.3.0 0.0.0.255 log
```

```
Router1(conf)#access-list 100 permit ip 1.1.2.0 0.0.0.255 any log
```

Первая команда непосредственно решает поставленную задачу, а вторая разрешает широковещание RIP протоколов. Проверим создание

```
Router1#show access-list
```

```
Extended IP access list 100
```

```
permit ip 1.1.1.0 0.0.0.127 1.1.3.0 0.0.0.255 log (0 matches)
```

```
permit ip 1.1.2.0 0.0.0.255 any log (11 matches)
```

Применим список доступа к интерфейсу.

```
Router1(conf)#interface Serial0
```

```
Router1(conf-if)#ip access-group 100 in
```

Для тестирования списка доступа, попытайтесь пропинговать PC1 от PC2, PC3, PC4 и PC5.

```
PC#Ping 1.1.3.2
```

Для PC2 и PC3 пинги не пойдут. Для PC4 и PC5 пинги пойдут. Список доступа работает. Посмотрите логи на router1

```
01:31:39: %SEC-6-IPACCESSLOGDP: list 100 permitted IP 1.1.1.2 -> 1.1.3.2 (8/0), 5 packets
```

6.2. Список доступа хост-хост.

Создадим на router2 список доступа, который блокирует доступ к PC5 только с PC2. Контролировать попытки доступа можно по логам на router2.

```
Router2(conf)# access-list 101 deny ip 1.1.1.130 0.0.0.0 1.1.1.3 0.0.0.0 log
```

```
Router2(conf)# access-list 101 permit ip any any
```

Проверим создание

```
Router2#show access-list
```

```
Extended IP access list 101
```

```
deny ip host 1.1.1.130 host 1.1.1.3 log (0 matches)
```

```
permit ip any any log (0 matches)
```

Применим список доступа к fast Ethernet интерфейсу router2

```
Router2(conf)#interface FastEthernet0/0
```

```
Router2(conf-if)#ip access-group 101 in
```

Присоединитесь к PC2 и проверьте, что вы не можете пиновать PC5

```
PC2# Ping 1.1.1.3
```

На router2 появится лог

```
01:51:44: %SEC-6-IPACCESSLOGDP: list 101 denied IP 1.1.1.130 -> 1.1.1.3 (8/0), 5 packets
```

Присоединитесь к PC3 и проверьте, что вы можете пиновать PC5.

```
PC3# Ping 1.1.1.3
```

На router2 появится лог

```
01:54:41: %SEC-6-IPACCESSLOGDP: list 101 permitted IP 1.1.1.131 -> 1.1.1.3 (8/0), 5 packets
```

6.3. Список доступа сеть-хост.

Вначале удалим предыдущие списки доступа с интерфейсов Router1 и Router2.

```
Router1(conf)#interface Serial0
```

```
Router1(conf-if)#no ip access-group 100 in
```

и

```
Router2(conf)#interface FastEthernet0/0
```

```
Router2(conf-if)#no ip access-group 102 in
```

Создадим расширенный список доступа, который блокирует весь трафик к PC1 из локальной сети компьютеров PC2 и PC3. Так как мы блокируем весь трафик, то будем использовать IP протокол.

```
Router2(conf)#access-list 102 deny ip 1.1.1.128 0.0.0.127 1.1.3.2 0.0.0.0 log
```

```
Router2(conf)#access-list 102 permit ip any any
```

Проверим создание

```
Router2#show access-list
```

```
Extended IP access list 102
```

```
deny ip 1.1.1.128 host 1.1.3.2 log (0 matches)
```

```
permit ip any any (0 matches)
```

Применим список к исходящему трафику на интерфейсе Serial0 Router2

```
Router2(conf)#interface Serial0
```

```
Router2(conf-if)#ip access-group 102 out
```

Для проверки списка попытайтесь пропинговать PC1 (1.1.3.2) из PC2 и PC3. Пинги не пройдут. Симулятор почему-то не даёт лог на консоли Router2. Но эффект выможете увидеть так

```
router2#sh ac
```

```
Extended IP access list 102
```

```
deny ip 1.1.1.128 host 1.1.3.2 log (70 matches)
```

```
permit ip any any (0 matches)
```

```
router2#sh ac
```

```
Extended IP access list 102
```

```
deny ip 1.1.1.128 host 1.1.3.2 log (140 matches)
```

```
permit ip any any (0 matches)
```

Вы видите после каждого неудачного пинга количество отслеженных (matches) пакетов возрастает.

Контрольные вопросы

1. Что такое ACL?
2. Какой адрес является критерием для разрешения/запрещения пакета?
3. Где применяются ACL?
4. Как задать элемент ACL и что такое инверсная маска?
5. Как роутер обрабатывает элементы ACL?
6. Какой элемент всегда неявно присутствует в ACL?
7. Как ACL применить к интерфейсу и затем его отменить?
8. Чем отличается входной ACL от выходного?
9. Где в сети рекомендуется размещать ACL?
10. Какими тремя командами можно проверить содержимое ACL и привязку к интерфейсу.
11. Что фильтруют расширенные ACL?
12. Какую дополнительную функциональность имеют расширенные ACL по сравнению со стандартными?

13. Можно ли, используя расширенные ACL, наложить ограничения на трафик к определённой TCP/IP службе?
14. Опишите процедуру создания именованного ACL.
15. Как отредактировать конкретную строку в числовом ACL?
16. Как отредактировать конкретную строку в именованном ACL?
17. Чем отличаются форматы команд для ввода элементов числового и именованного ACL?

Порядок выполнения и сдачи работы

1. Изучить теоретическую и практическую часть.
2. Сдать преподавателю теорию работы путём ответа на контрольные вопросы.
3. Выполнить подряд три пункта практической части. Проверить доступ согласно таблице 1, а затем таблице 2.
4. Показать преподавателю, что сеть удовлетворяет матрице доступа из таблицы 1.
5. Показать преподавателю, что сеть удовлетворяет матрице доступа из таблицы 2.
6. Выполните в Boson задание для самостоятельной работы.
7. Построить свою матрицу доступа согласно варианту и показать преподавателю, что сеть удовлетворяет этой матрице.
8. Оформите отчёт. Содержание отчёта смотри ниже.
9. Защитите отчёт.

Задание для самостоятельной работы

1. Применить список доступа 2, созданный в пункте 3 практической части, согласно варианту

Вариант	Маршрутизатор	Интерфейс	Вх/вых
1	router 1	E0	ВХ
2	router 1	S0	ВЫХ
3	router 1	E1	ВХ
4	router 1	E1	ВЫХ
5	router 2	E0	ВХ
6	router 2	E0	ВЫХ
7	router 2	S0	ВХ
8	router 2	S0	ВЫХ
9	router 2	E1	ВХ
10	router 2	E1	ВЫХ

2. Создать скриншот результата выполнения команд
`router1#show access-list`
и
`router1#show running-config`
3. Построить матрицу доступа.

4. Выполните три подпункта пункта 6 практической части для топологии на рисунке 4 и своего варианта **v** конфигурации, приведенной в таблице

	Router1	Router2
Fa0/0	1.v.3.1/24	1.v.1.129/25
Fa0/1		1.v.1.1/25
Serial0	1.v.2.1/24	1.v.2.2/24

Hostname	IP на ethernet	Шлюз
PC1	1.v.3.2 255.255.255.0	1.v.3.1
PC2	1.v.1.130 255.255.255.128	1.v.1.129
PC3	1.v.1.131 255.255.255.128	1.v.1.129
PC4	1.v.1.2 255.255.255.128	1.v.1.1
PC5	1.v.1.3 255.255.255.128	1.v.1.1

Сделайте те же скриншоты, которые сделаны в пункте 6 практической части.

Содержание отчёта.

Отчёт готовится в электронном виде и распечатывается. Отчёт содержит

1. Скриншоты топологий, созданных при выполнении практической части.
2. Все скриншоты, созданные при выполнении практической части.
3. Конфигурации всех маршрутизаторов и компьютеров из rtr файлов, созданных при выполнении практической части.
4. Конфигурацию маршрутизаторов из rtr файлов, созданную при выполнении пункта 1 задания для самостоятельной работы.
5. Матрицу доступа, указанные в пункте 3 задания для самостоятельной работы.
6. Конфигурацию маршрутизаторов из rtr файлов, созданную при выполнении пункта 4 задания для самостоятельной работы.
7. Все скриншоты, указанные в задании для самостоятельной работы.

Лабораторная работа №7. Преобразование сетевых адресов NAT

Теоретическая часть

Network address translation (NAT - перенос сетевых адресов) создан для упрощения и сокрытия IP адресации. NAT позволяет представить внешнему миру внутреннюю структуру IP адресации предприятия иначе, чем она на самом деле выглядит. Это разрешает организации соединяться с Интернетом, не имея внутри себя глобальной уникальной IP адресации. Это даёт возможность выхода в Интернет для корпоративных внутренних IP сетей с внутренними IP адресами (intranet), которые глобально неуникальны и поэтому не могут маршрутизироваться в Интернете. NAT применяется также для связи территориально распределённых подразделений организации через Интернет.

Мировым сообществом для Интранет адресации выделены следующие диапазоны адресов

Class A: 10.0.0.0-10.255.255.255

Class B: 172.16.0.0-172.16.255.255

Class C: 192.168.1-192.168.255.255

NAT переводит внутренний IP адрес из внутреннего адресного пространства в IP адрес во внешнем адресном пространстве. Когда NAT получает пакет из intranet, он изменяет в нём адрес источника, пересчитывает контрольную сумму и отправляет его в Интернет.

NAT преобразует и отображает адреса из одной области в другую. Это обеспечивает прозрачную маршрутизацию от узла к узлу. В NAT существует несколько способов трансляции адресов, используемых в различных частных случаях.

Cisco использует для NAT специфическую терминологию для узлов в intranet и интернет как до, так и после преобразования адресов:

Внутренний (inside) адрес. Адрес, используемый в организации. Разные организации могут иметь одинаковые внутренние адреса.

Внешний адрес (outside). Адрес, определённый где-либо вне данной организации. Внешний адрес иной организации может совпадать с внутренним адресом данной организации.

Глобальный адрес. Это зарегистрированный и законный адрес IP, который может проходить через Интернет.

Локальный адрес. Адрес IP, используемый внутренне в Intranet. Эти адреса не пересекают Интернет адреса и поэтому рассматриваются как локальные.

Внутренний локальный адрес (inside local). Адрес, используемый в организации, не пересекающие Интернет адреса.

Внутренний глобальный адрес (inside global). Адрес, используемый в организации, являющийся Интернет адресом.

Внешний локальный адрес (outside local). Адрес, определённый где-либо вне данной организации, не являющийся Интернет адресом.

Внешний глобальный адрес (outside global). Адрес, определённый где-либо вне данной организации, являющийся Интернет адресом.

Симулятор всегда показывает, что Внешний локальный адрес (Outside Local) всегда равен внешнему глобальному адресу (outside global).

При отправке пакетов от интерфейса внутреннего хоста NAT заменяет в нём адрес источника на некоторый глобальный адрес. При приёме ответного пакета NAT заменяет в нём глобальный адрес приёмника (адрес внешнего интерфейса локального маршрутизатора) на адрес интерфейса внутреннего хоста. Для такой замены маршрутизатор поддерживает специальные таблицы преобразований адресов, которые постоянно обновляются. Различают три способа преобразования адресов: статический, динамический и перегрузка (overload). При статическом NAT в явном виде с помощью команд IOS задаются пары внутренний_адрес - глобальный _адрес. При динамическом преобразовании глобальные адреса берутся из определённого пула внешних адресов. При перегрузке все внутренние адреса, подлежащие преобразованию, заменяются на единственный глобальный адрес внешнего интерфейса маршрутизатора.

Для конфигурирования NAT следует определить на маршрутизаторе внутренние и внешние сети с помощью команд **ip nat inside | outside**. Эти команды определяются на уровне интерфейсов, то есть в контексте команды **interface**. Дополнительные команды зависят от используемого типа NAT. Это либо задание статического NAT, либо определение пула внешних адресов либо задание команды для перегрузки. Как правило, следует также задать список управления доступом ACL для определения внутреннего трафика, который будет преобразовываться. Сам по себе ACL не осуществляет никакого NAT преобразования.

Остановимся дополнительно на перегрузке, которую, как правило называют PAT (Port Address Translation - преобразование адресов с помощью портов). PAT разрешает нескольким внутренним хостам использовать один глобальный адрес. Один из вариантов реализации PAT базируется на использовании последовательности свободных TCP и UDP портов и состоит в следующем.

Служба NAT находится на маршрутизаторе M с внутренним глобальным адресом X. Пусть имеется пакет П от внутреннего локального адреса Y_i и порта P_i к внешнему адресу G и порту p. Пакет проходит через маршрутизатор M. NAT заменяет в нём адрес Y_i источника и порт источника P_i внутреннего хоста на уникальную пару X:P_{inew}, где P_{inew} – очередной свободный номер порта. Строка (Y_i:P_i , X: P_{inew}, G:p) заносится в таблицу NAT маршрутизатора M. Ответный пакет в поле адрес приёмника будет содержать X, а в поле порт приёмника – P_{inew}. В поле адрес:порт источника будет G:p. Маршрутизатор M по значениям X, P_{inew}, G, p находит в своей NAT таблице строку (Y_i:P_i , X: P_{inew}, G:p) и заменяет в пакете адрес приёмника X на Y_i, а порт назначения P_{inew} на P_i. Ответный пакет придёт по нужному адресу Y_i с нужным номером порта назначения P_i. Порт P_{inew} возвращается в список свободных портов.

Процесс NAT прозрачен для внутренних адресов. Так хост с внутренним адресом Y_i, отправивший пакет во внешний мир и получивший ответ «не догадывается», что пакет прошел NAT преобразование на маршрутизаторе как

при отправке так и при приёме. Внутреннему хосту представляется, что он имеет непосредственный выход во внешний мир.

Практическая часть.

1. Загрузите в симулятор топологию изображённую на рисунке.

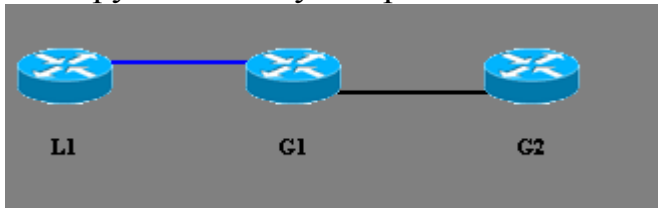


Рис. 1

Создайте конфигурацию

	L1	G1	G2
Ethern	10.10.1.2 255.255.255.0	10.10.1.1 255.255.255.0	
Serial 0		175.10.1.1 255.255.255.0	175.10.1.2 255.255.255.0

Задайте OSPF маршрутизацию на G1

G1(config)#**router ospf 1**

G1(config- router)#**network 10.10.1.0 0.0.0.255 area 0**

G1(config- router)#**network 175.10.1.0 0.0.0.255 area 0**

На L1

L1 (config)#**router ospf 1**

L1 (config- router)#**network 10.10.1.0 0.0.0.255 area 0**

Для G2

G2 (config)#**router ospf 1**

G2(config- router)#**network 175.10.1.0 0.0.0.255 area 0**

На G2 разрешим вход по телней с паролем G2

G2 (config)#**line vty 0 4**

G2 (config-line)#**login**

G2 (config-line)#**password G2**

Проверьте работоспособность сети с помощью команды ping. Проверьте Telnet-соединение из L1 (10.10.1.2)к G2

L1# **telnet 175.10.1.2**

Password:**G2**

G2>**show users**

	Line	User	Host(s)	Idle	Location
*	0 con 0		idle	00:00:00	
*	1 vty 1		idle	10.10.1.2	

Выйдите из телнета:ctrl+shift+6 затем x. Закройте сессии телнет с помощью команды disconnect на L1 и G2.

Необходимо настроить NAT/PAT на маршрутизаторе G1. Нужно настроить три формы перевода адресов: статистический перевод сетевых адресов, динамический перевод и перегрузка (перевод адресов портов). Предполагаем, что сеть Ethernet 10.10.1.0/24 это внутренняя сеть intranet,

которая соединяется с внешним миром через G1, а именно через сеть 175.10.1.0/24 последовательного интерфейса.

1.1 На маршрутизаторе G1 настроим NAT на статистический перевод внутреннего локального Ethernet-адреса маршрутизатора L1 10.10.1.2 в новый внутренний глобальный адрес 169.10.1.2, имеющийся у организации:

```
G1(config)#ip nat inside source static 10.10.1.2 169.10.1.2
```

Определим сеть ethernet0 интерфейса как внутреннюю

```
G1(config)#interface ethernet0
```

```
G1(config-if)#ip nat inside
```

Определим сеть serial0 интерфейса как внешнюю

```
G1(config-if)#interface serial0
```

```
G1(config-if)#ip nat outside
```

Для вывода таблицы переводов адресов введите команду на маршрутизаторе

```
G1#show ip nat translations
```

Pro	Inside global	Inside local	Outside local	Outside global
---	169.10.1.2	10.10.1.2	---	---

Проверьте перевод, создав Telnet-соединение из L1 к G2. Войдя по Telnet на G2 введите команду show users. Вы должны увидеть переведенный IP-адрес 169.10.1.2.

```
L1# telnet 175.10.1.2
```

```
Password:G2
```

```
G2> show users
```

Line	User	Host(s)	Idle Location
* 0 con 0		idle	00:00:00
* 1 vty 1		idle	169.10.1.2

Вместо 169.10.1.2 можно использовать и задействованный адрес интерфейса Serial0 G1

```
175.10.1.1
```

```
G1(config)#no ip nat inside source static 10.10.1.2 169.10.1.2
```

```
G1(config)#ip nat inside source static 10.10.1.2 1 175.10.1.1
```

```
L1# telnet 175.10.1.2
```

```
Password:G2
```

```
G2> show users
```

Line	User	Host(s)	Idle Location
* 0 con 0		idle	00:00:00
* 1 vty 1		idle	175.10.1.1

1.2. На маршрутизаторе G1 удалите предыдущую NAT-команду статистического перевода

```
G1(config)#no ip nat inside source static 10.10.1.2 175.10.1.1
```

и настройте NAT для динамического перевода Ethernet-адреса маршрутизатора L1. Используем область адресов в диапазоне: 169.10.1.50 - 169.10.1.100.

Определим пул адресов **pool1**

```
G1(config)#ip nat pool pool1 169.10.1.50 169.10.1.100 netmask 255.255.255.0
```

Зададим NAT преобразование адресов из списка 1 в пул адресов **pool1**

```
G1(config)#ip nat inside source list 1 pool pool1
```

Настроим и проверим список доступа 1 для внутренних адресов, которые будут преобразовываться

```
G1(config)#access-list 1 permit 10.10.1.0 0.0.0.255
```

```
^Z
```

```
G1#Show access-list
```

Обязательно с помощью команды show running-config проверьте, что маршрутизатор G1 «подхватил» введенные команды. Среди вывода должно быть

```
!
ip nat pool pool1 169.10.1.50 169.10.1.100 netmask 255.255.255.0
ip nat inside source list 1 pool pool1
ip classless
no ip http server
access-list 1 permit 10.1.1.0 0.0.0.255
!
!
```

Проверьте перевод, создав Telnet-соединение из L1 к маршрутизатору G2. Войдя по Telnet на G2 введите команду show users. Вы должны увидеть переведенный IP-адрес 169.10.1.50.

```
L1# telnet 175.10.1.2
```

```
Password:G2
```

```
G2> show users
```

Line	User	Host(s)	Idle Location
* 0 con 0		idle	00:00:00
* 1 vty 1		idle	169.10.1.50

То есть NAT взял из пула адресов первый свободный адрес 169.10.1.50.

Для вывода таблицы переводов адресов введите команду show ip nat translations на маршрутизаторе G1.

Pro	Inside global	Inside local	Outside local	Outside global
icmp	169.10.1.50:9392	10.10.1.2:9392	175.10.1.2:9392	175.10.1.2:9392
icmp	169.10.1.50:9393	10.10.1.2:9393	175.10.1.2:9393	175.10.1.2:9393
icmp	169.10.1.50:9394	10.10.1.2:9394	175.10.1.2:9394	175.10.1.2:9394
icmp	169.10.1.50:9395	10.10.1.2:9395	175.10.1.2:9395	175.10.1.2:9395
icmp	169.10.1.50:9396	10.10.1.2:9396	175.10.1.2:9396	175.10.1.2:9396

1.3 На маршрутизаторе G1 удалите предыдущие NAT-команды динамического перевода и настройте перегрузку (перевод адресов портов) Ethernet-адреса маршрутизатора L1 (10.10.1.2) на адрес интерфейса serial0 (175.10.1.1) на маршрутизаторе G1.

```
G1(config)#no ip nat pool pool1 169.10.1.50 169.10.1.100 netmask 255.255.255.0
```

```
G1(config)#no nat inside source list 1 pool pool1
```

```
G1(config)#ip nat inside source list 1 interface serial0 overload
```

Обязательно с помощью команды show running-config проверьте, что маршрутизатор G1 «подхватил» введенные команды. Среди вывода должно быть

```
ip nat inside source list 1 interface Serial0 overload
ip classless
no ip http server
access-list 1 permit 10.10.1.0 0.0.0.255
```

Проверьте перевод, создав Telnet-соединение из L1 к маршрутизатору G2. Находясь на маршрутизаторе G2, введите команду `show users`. Вы должны увидеть переведенный IP-адрес 175.10.1.1.

L1# **telnet 175.10.1.2**

Password:G2

G2> **show users**

```

Line      User      Host(s)      Idle Location
*  0 con 0          idle          00:00:00
*  1 vty 1          idle          175.10.1.1

```

Выведите список NAT-переводов на маршрутизаторе G1.

G1#`show ip nat translations`

```

Pro  Inside global      Inside local      Outside local      Outside global
icmp175.10.1.1:9392    10.10.1.2:9392    175.10.1.2:9392    175.10.1.2:9392
icmp175.10.1.1:9393    10.10.1.2:9393    175.10.1.2:9393    175.10.1.2:9393
icmp175.10.1.1:9394    10.10.1.2:9394    175.10.1.2:9394    175.10.1.2:9394
icmp175.10.1.1:9395    10.10.1.2:9395    175.10.1.2:9395    175.10.1.2:9395
icmp175.10.1.1:9396    10.10.1.2:9396    175.10.1.2:9396    175.10.1.2:9396

```

2. Создайте топологию. Модели маршрутизаторов L1, L2, G1 и G2 -805, а LG и G – 1605.

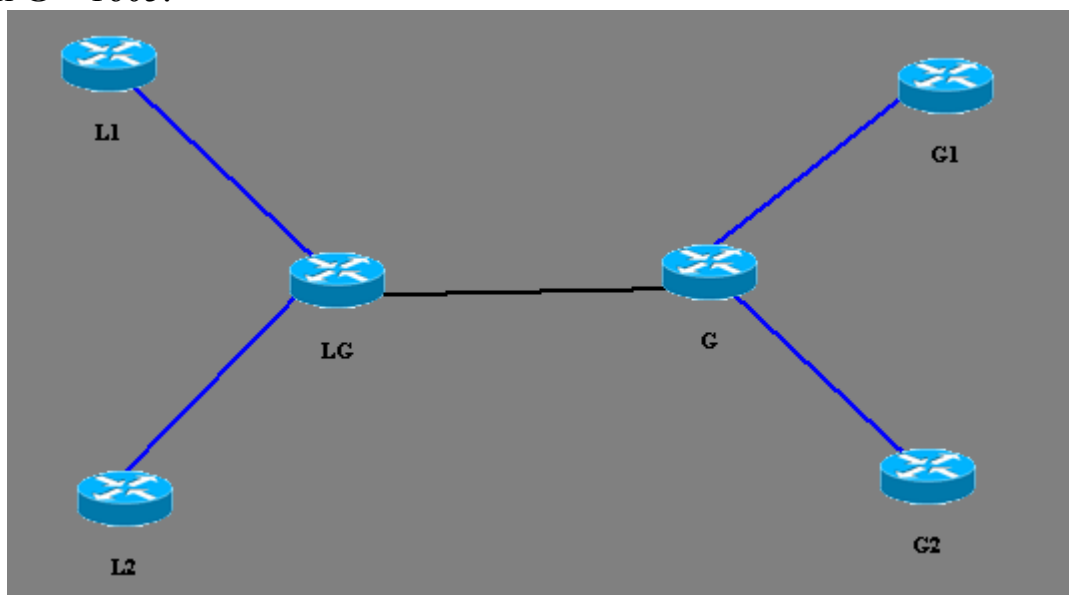


Рис. 2

Сконфигурируем адреса согласно таблицы. Все маски 255.255.255.0

	L1	L2	LG	G	G1	G2
Eth0	10.1.1.2		10.1.1.1	1.1.2.1	1.1.2.2	
Eth1		10.1.2.2	10.1.2.1	1.1.3.1		1.1.3.2
serial			1.1.1.1	1.1.1.2		

Настроим маршрутизацию OSPF

hostname LG

```
LG(config)#router ospf 1
LG(config-router)#network 1.1.1.0 0.0.0.255 area 10
LG(config-router)#network 10.1.1.0 0.0.0.255 area 10
LG(config-router)#network 10.1.2.0 0.0.0.255 area 10
```

```
G(config)# router ospf 1
LG(config-router)#network 1.1.1.0 0.0.0.255 area 10
LG(config-router)#network 1.1.2.0 0.0.0.255 area 10
LG(config-router)#network 1.1.3.0 0.0.0.255 area 10
```

```
L1(config)# router ospf 1
L1(config-router)#network 10.1.1.0 0.0.0.255 area 10
```

```
L2(config)# router ospf 1
L2(config-router)#network 10.1.2.0 0.0.0.255 area 10
```

```
G1(config)#router ospf 1
G1(config-router)#network 1.1.2.0 0.0.0.255 area 10
```

```
G2(config)#router ospf 1
G2(config-router)#network 1.1.3.0 0.0.0.255 area 10
```

и разрешим телнет на G1 и G2:

```
G1(config)#line vty 0 4
G1(config-line)#login
G1(config-line)#password G1
```

```
G2(config)# line vty 0 4
G2(config-line)#login
G2(config-line)#password G2
```

Проверьте функционирование: зайдите телнетом из L1 на G1. из L1 на G2, из L2 на G1, из L2 на G2.

Обязательно убейте все телнет сессии командой **disconnect 1**. Всего вам понадобится 8 команд. По две на каждое устройство: L1, L2, G1 и G2.

Проверьте отсутствие телнет соединений командой **show session**.

Объявите на роутере LG сети 10.1.1.0/24 и 10.1.2.0/24 как внутренние, а сеть 1.1.1.0/24 как внешнюю

```
LG(config)#interface Serial0
LG(config-if)#ip nat outside
LG(config-if)#interface Ethernet0
LG(config-if)#ip nat inside
LG(config-if)#interface Ethernet1
LG(config-if)#ip nat inside
```

Создадим список доступа для адресов двух внутренних Ethernet сетей

LG(config)#**access-list 2 permit 10.1.0.0 0.0.255.255**

И запустим PAT преобразование этих адресов на внутренний глобальный адрес 1.1.1.1 интерфейса Serial0

LG(config)#**ip nat inside source list 2 interface Serial0 overload**

Проверьте функционирование: зайдите телнетом из L1 на G1

L1# **telnet 1.1.2.2**

Password:G1

G1> **show users**

	Line	User	Host(s)	Idle Location
*	0 con 0		idle	00:00:00
*	1 vty 1		idle	1.1.1.1

Выйдите из телнет сессии, нажав Ctrl-shift-6 затем x.

Зайдите телнетом из L1 на G2

L1# **telnet 1.1.3.2**

Password:G2

G2> **show users**

	Line	User	Host(s)	Idle Location
*	0 con 0		idle	00:00:00
*	1 vty 1		idle	1.1.1.1

Выйдите из телнет сессии, нажав Ctrl-shift-6 затем x.

Перейдём на устройство L2. Зайдите телнетом из L2 на G1

L2# **telnet 1.1.2.2**

Password:G1

G1>**sh us**

	Line	User	Host(s)	Idle Location
*	0 con 0		idle	00:00:00
*	1 vty 1		idle	1.1.1.1
*	2 vty 2		idle	1.1.1.1

Выйдите из телнет сессии, нажав Ctrl-shift-6 затем x.

Зайдите телнетом из L2 на G2

L2# **telnet 1.1.2.2**

Password:G2

G2>**sh us**

	Line	User	Host(s)	Idle Location
*	0 con 0		idle	00:00:00
*	1 vty 1		idle	1.1.1.1
*	2 vty 2		idle	1.1.1.1

Мы видим и на G1 и на G2 по две телнет сессии. Все четыре сессии показаны как сессии от одного адреса - внутреннего глобального адреса 1.1.1.1 интерфейса Serial0.

Перейдём на LG. Посмотрим четыре трансляции адресов

LG#**show ip nat tr**

Pro	Inside global	Inside local	Outside local	Outside global
icmp	1.1.1.1:9392	10.1.1.2:9392	1.1.2.2:9392	1.1.2.2:9392
icmp	1.1.1.1:9393	10.1.1.2:9393	1.1.2.2:9393	1.1.2.2:9393
icmp	1.1.1.1:9394	10.1.1.2:9394	1.1.2.2:9394	1.1.2.2:9394
icmp	1.1.1.1:9395	10.1.1.2:9395	1.1.2.2:9395	1.1.2.2:9395
icmp	1.1.1.1:9396	10.1.1.2:9396	1.1.2.2:9396	1.1.2.2:9396
icmp	1.1.1.1:9392	10.1.1.2:9392	1.1.3.2:9392	1.1.3.2:9392
icmp	1.1.1.1:9393	10.1.1.2:9393	1.1.3.2:9393	1.1.3.2:9393
icmp	1.1.1.1:9394	10.1.1.2:9394	1.1.3.2:9394	1.1.3.2:9394
icmp	1.1.1.1:9395	10.1.1.2:9395	1.1.3.2:9395	1.1.3.2:9395
icmp	1.1.1.1:9396	10.1.1.2:9396	1.1.3.2:9396	1.1.3.2:9396
icmp	1.1.1.1:9392	10.1.2.2:9392	1.1.2.2:9392	1.1.2.2:9392
icmp	1.1.1.1:9393	10.1.2.2:9393	1.1.2.2:9393	1.1.2.2:9393
icmp	1.1.1.1:9394	10.1.2.2:9394	1.1.2.2:9394	1.1.2.2:9394
icmp	1.1.1.1:9395	10.1.2.2:9395	1.1.2.2:9395	1.1.2.2:9395
icmp	1.1.1.1:9396	10.1.2.2:9396	1.1.2.2:9396	1.1.2.2:9396
icmp	1.1.1.1:9392	10.1.2.2:9392	1.1.3.2:9392	1.1.3.2:9392
icmp	1.1.1.1:9393	10.1.2.2:9393	1.1.3.2:9393	1.1.3.2:9393
icmp	1.1.1.1:9394	10.1.2.2:9394	1.1.3.2:9394	1.1.3.2:9394
icmp	1.1.1.1:9395	10.1.2.2:9395	1.1.3.2:9395	1.1.3.2:9395
icmp	1.1.1.1:9396	10.1.2.2:9396	1.1.3.2:9396	1.1.3.2:9396

Порядок трансляций соответствует порядку ввода команд **telnet**

Первая порция трансляций соответствует команде

L1# **telnet 1.1.2.2**

Вторая порция трансляций соответствует команде

L1# **telnet 1.1.3.2**

Третья порция трансляций соответствует команде

L2# **telnet 1.1.2.2**

Четвёртая порция трансляций соответствует команде

L2# **telnet 1.1.3.2**

Во всех случаях используется внутренний глобальный адрес 1.1.1.1 интерфейса Serial0.

Контрольные вопросы

1. Какие задачи решает NAT.
2. Как связана проблема нехватки IP адресов и NAT.
3. Какие вы знаете диапазоны для Интранет адресации?
4. Что такое внутренний адрес?
5. Что такое внешний адрес?
6. Что такое глобальный адрес?
7. Что такое локальный адрес?
8. Что такое внутренний локальный адрес?
9. Что такое внутренний глобальный адрес?
10. Что такое внешний глобальный адрес?
11. Что такое внешний локальный адрес?
12. Назовите три способа преобразования адресов.

13. Объясните как работает PAT.

14. Какая функциональность теряется у хоста с внутренним локальным адресом в сети, имеющей связь с внешним миром через NAT, по сравнению с хостом с глобальным адресом?

Порядок выполнения и сдачи работы

1. Изучить теоретическую и практическую часть.
2. Сдать преподавателю теорию работы путём ответа на контрольные вопросы.
3. Выполнить подряд все пункты практической части.
4. Создать все скриншоты, приведенные в практической части.
5. Показать преподавателю результат выполнения последней команды **show ip nat tr** из пунктов 1 и 2 практической части.
6. Выполните в Boson задание для самостоятельной работы.
7. Показать преподавателю результат выполнения последней команды **show ip nat tr** из пунктов 1 и 2 задания для самостоятельной работы.
8. Оформите отчёт. Содержание отчёта смотри ниже.
9. Защитите отчёт.

Задание для самостоятельной работы

1. Выполните пункт 1 практической части для топологии на рисунке 1 и своего варианта **v** конфигурации, приведенной в таблице

	L1	G1	G2
Ethern	10.10.v.2 255.255.255.0	10.10.v.1 255.255.255.0	
Serial 0		175.v.1.1 255.255.255.0	175.v.1.2 255.255.255.0

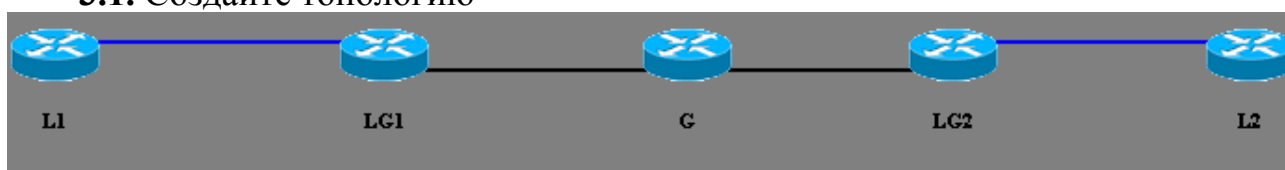
Сделайте те же скриншоты, которые сделаны в пункте 1 практической части.

2. Выполните пункта 2 практической части для топологии на рисунке 2 и своего варианта **v** конфигурации, приведенной в таблице

	L1	L2	LG	G	G1	G2
Eth0	10.v.1.2		10.v.1.1	1.v.2.1	1.v.2.2	
Eth1		10.v.2.2	10.1.2.1	1.v.3.1		1.v.3.2
serial			1.v.1.1	1.v.1.2		

Сделайте те же скриншоты, которые сделаны в пункте 2 практической части.

3.1. Создайте топологию



3.2. Сконфигурируем адреса согласно таблицы. Все маски 255.255.255.0. **v** – номер варианта.

	L1	LG1	G	LG2	L2
Serial0		1.v.1.2	1.v.1.1		
Serial1			1.v.2.1	1.v.2.2	
Ethernet	10.v.1.2	10.v.1.1		10.v.1.1	10.v.1.2

Мы видим совершенно одинаковые адреса для двух Ethernet сетей. Тем самым мы моделируем реальную ситуацию, когда сетевые устройства L1 и L2 двух различных фирм имеют одинаковые локальные внутренние адреса 10.v.1.2 и связаны с внешним миром (G) через маршрутизаторы LG1 и LG2, имеющими NAT.

3.3. Разрешите доступ по телнет к маршрутизатору G.

3.4. Дайте доказательный ответ на вопрос: можно ли в симуляторе так настроить маршрутизацию и NAT, чтобы маршрутизатор L1 (L2) вошел по телнет на маршрутизатор G под глобальным внутренним адресом 1.v.1.2 (1.v.2.2). Ответ (положительный или отрицательный) снабдите содержимым `rtg` файлов и скриншотами.

Содержание отчёта.

Отчёт готовится в электронном виде и распечатывается. Отчёт содержит

1. Скриншоты топологий, созданных при выполнении практической части.
2. Все скриншоты, созданные при выполнении практической части.
3. Конфигурации всех из `rtg` файлов, созданных при выполнении практической части.
6. Конфигурацию маршрутизаторов из `rtg` файлов, созданные при выполнении для самостоятельной работы.
7. Все скриншоты, указанные в задании для самостоятельной работы.

Лабораторная работа №8. Удалённый доступ. Frame Relay

Теоретическая часть

Удалённый доступ

Существуют три типа последовательных каналов связи:

1. Сети с коммутацией каналов: коммутируемая телефонная сеть, ISDN.
2. Выделенные или арендованные каналы: DSL, кабельные модемы и каналы связи типа 64к, T1, T3, OC-12,
3. Сети с коммутацией пакетов: Frame Relay, X.25 и ATM.

Передача данных может осуществляться как синхронно, так и асинхронно. Арендованные выделенные каналы обычно требуют синхронного последовательного соединения.

Каждая линия связи соединена с последовательный порт маршрутизатора устройством CSU/DSU (channel service unit/data service unit – устройство обслуживания канала/ устройство обслуживания данных).

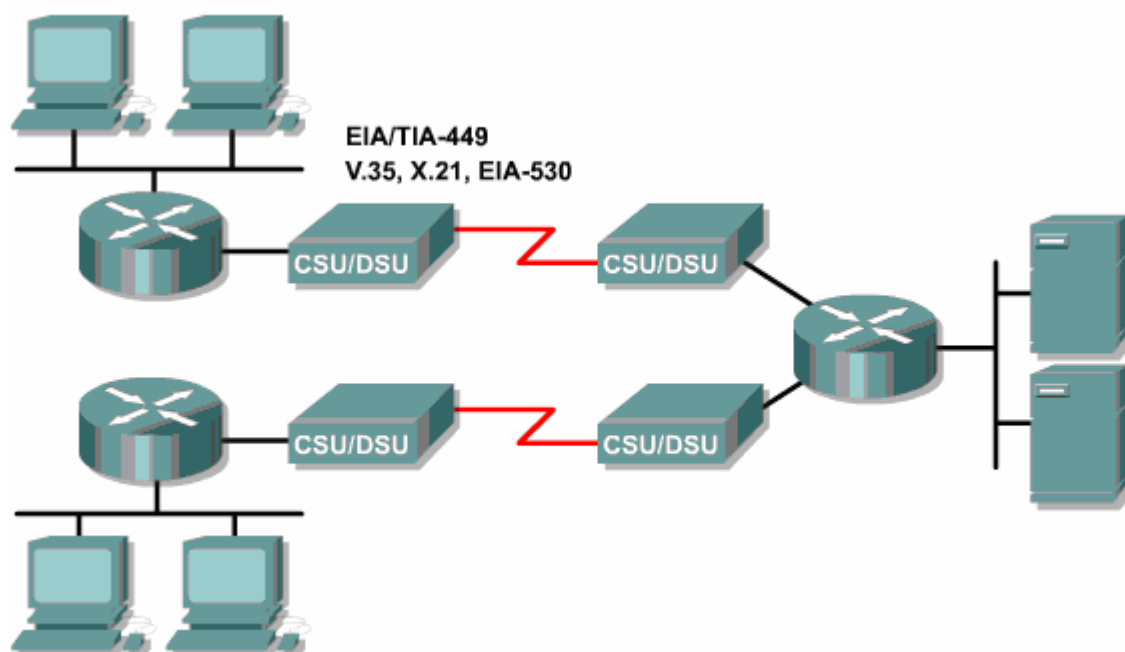


Рис. 1.

Следовательно, помимо собственно проводов от коммуникационного провайдера, каждое соединение требует отдельный синхронный последовательный порт на маршрутизаторе и устройство CSU/DSU. Как результат, стоимость содержания нескольких выделенных линий резко возрастает. Поэтому большинство компаний находят построение полносвязных WAN на выделенных линиях слишком дорогим решением.

CSU/DSU является DCE (data communications equipment – оборудование передачи данных) устройством. DCE адаптирует физический интерфейс на DTE (data terminal equipment – терминальное оборудование данных) устройстве к

правилам передачи сигналов в несущей сети. Примером DTE устройства является маршрутизатор.

CSU/DSU обеспечивает модуляцию, временную синхронизацию и используется для взаимодействия с цифровой передающей аппаратурой. Существенно, что CSU/DSU используется маршрутизатором для связи с цифровым каналом во многом таким же образом, как ПК использует модем для связи с аналоговым телефонным каналом. То есть в связке ПК-модем ПК является DTE устройством, а модем – DCE устройством.

Выделенные или арендованные каналы

Обычно последовательные соединения в сетях оперируют со следующими скоростями

56 kbps

64 kbps

T1 (1.544 Mbps) стандарт США

E1 (2.048 Mbps) европейский стандарт

E3 (34.064 Mbps) европейский стандарт

T3 (44.736 Mbps) стандарт США

DTE (маршрутизатор) связан с DCE (CSU/DSU) через синхронный последовательный порт с использованием одного из следующих стандартов: EIA/TIA-232 (RS-232), EIA/TIA-449, V.35, X.21, EIA-530.

Для соединения DTE (маршрутизатор, ПК) к аналоговому модему (DCE (CSU/DSU)) используется EIA/TIA-232. Выпущенный более 30 лет назад, он стал классикой. Однако он не приемлем на выделенных линиях для скоростей более чем 120 kbps.

Стандарт V.35 пригоден для более высоких скоростей (более 2 Mbps). Он используется для подсоединения порта маршрутизатора к T1/E1 через отдельное CSU/DSU.

Сегодня для многих последовательных каналов связи, таких как T1, многие маршрутизаторы имеют внутри себя CSU/DSU, интегрированную в интерфейсную карту. То есть, нет нужды в отдельном DCE (CSU/DSU).

Сети с коммутацией пакетов СКП.

В отличие от арендованных линий и соединений с переключением каналов, переключение пакетов не полагается на выделенное, точка-точка соединение через несущую сеть. Вместо этого пакеты данных маршрутизируются в несущей сети на основании адресации, содержащейся в заголовках пакетов или фреймов, что позволяет провайдеру поддерживать много потребителей на одних физических линиях и коммутаторах.

В сетях с коммутацией пакетов провайдер конфигурирует свою аппаратуру коммутации для создания виртуальных цепей (virtual circuits (VCs)) для обеспечения сквозной связи. Виртуальные цепи могут быть постоянными или могут быть построены по требованию. Frame Relay является типовой службой по коммутации пакетов в WAN. Часто используются также технологии ATM и X.25.

Первые СКП строились по протоколу X.25. Так как X.25 был спроектирован для работы на ненадёжных телефонных медных цепях, то он обеспечивает обнаружение ошибок

СКП это сети с множественным доступом, использующие специальное оборудование для доставки пользовательского трафика. Физическим носителем сигналов внутри СКП являются, как правило, высокоскоростные выделенные каналы связи, чаще всего оптические. Для обеспечения требуемого сквозного соединения внутри СКП осуществляется настройка коммутационного оборудования.

СКП предлагают администратору сети меньший контроль, чем соединение точка-точка. Однако, стоимость виртуальной цепи в СКП меньше, чем выделенной линии. Доступ к самой СКП, как правило, осуществляется через синхронный последовательный порт маршрутизатора по единственной выделенной линии T1 или T3 и позволяет соединиться со многими удалёнными сайтами. При отсутствии СКП потребуется отдельная выделенная линия к каждому удалённому сайту.

Виртуальные цепи Frame Relay предоставляют скорость вплоть до скорости T3, делая технологию переключения пакетов высокоскоростной эффективной по стоимости альтернативой выделенным линиям. Единственное синхронное последовательное соединение к СКП может поддерживать несколько виртуальных цепей в конфигурациях точка-многоточка (один-ко-многим) и точка-точка (Рис. 2). Процесс комбинирования нескольких передач данных в одной физической линии называется мультиплексированием.

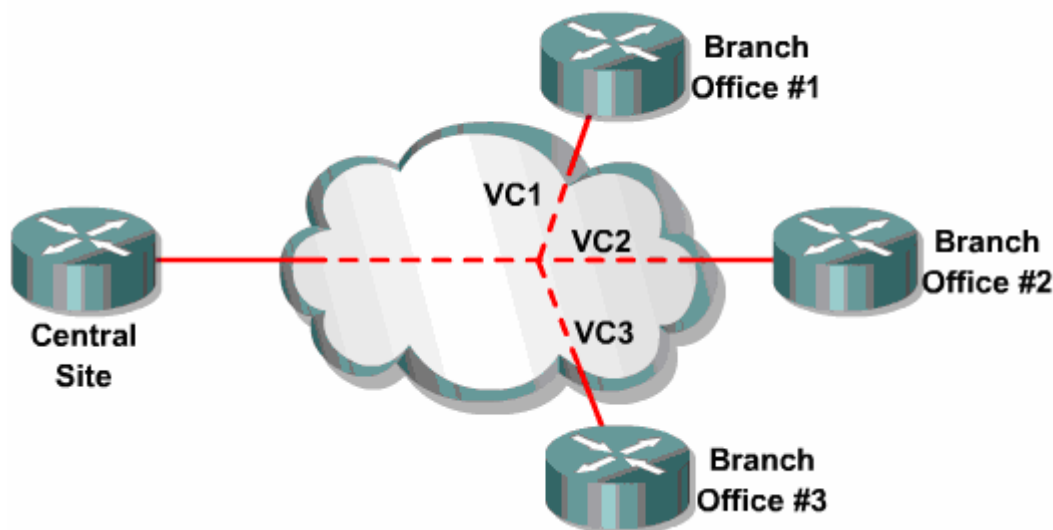


Рис. 2.

Мультиплексирование становится возможным благодаря тому, что DTE (как правило маршрутизатор) инкапсулирует данные в пакеты СКП, содержащие адреса СКП. Коммутаторы провайдера используют эти адреса для определения, как и куда доставить отдельный пакет. В случае Frame Relay эти адреса называются DLCI (Data Link Connection Identifiers). Способность к мультиплексированию означает, что один порт маршрутизатора вместе с одним устройством CSU/DSU может поддерживать десятки виртуальных цепей,

каждая из которых ведёт к отдельному сайту. Следовательно, переключение пакетов делает доступной по цене даже создание полносвязной топологии.

Frame Relay не исправляет ошибок и хорошо приспособлен для работы на высоконадёжных цифровых каналах передачи.

Frame Relay не обеспечивает ту степень надёжности, гибкости и безопасности, которую предоставляют выделенные линии, которые более предпочтительны для критически важного трафика и обмена большим объемом информации.

Другой популярной технологией коммутации пакетов является АТМ (Asynchronous Transfer Mode). АТМ это международный стандарт для поэлементной передачи, в котором информация различных типов таких, как данные, голос и видео помещаются в ячейки фиксированной длины в 53 байта. Фиксированная длина ячеек позволяет вести их обработку аппаратным способом, что сокращает временные задержки. АТМ спроектирована так чтобы воспользоваться преимуществами высокоскоростных сред передачи Е3, Т3и SONET.

Протоколы глобальных сетей

Маршрутизаторы прежде чем передать IP пакеты по каналу связи, инкапсулируют их во фреймы второго сетевого уровня.

Типовые протоколы второго уровня глобальных сетей:

Point-to-Point Protocol (PPP) – протокол связи маршрутизатор - маршрутизатор и маршрутизатор -хост для синхронных и асинхронных каналов.

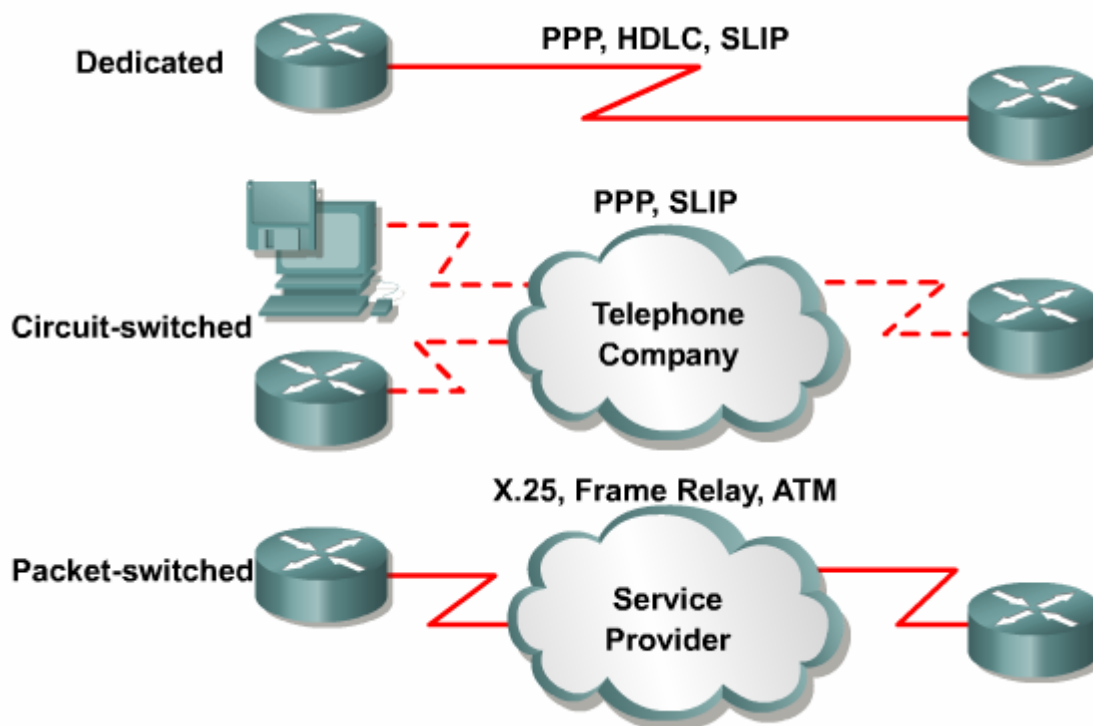


Рис. 3.

Serial Line Internet Protocol (SLIP) – предшественник PPP, используемый для последовательного соединения точка-точка по протоколу TCP/IP.

High-Level Data Link Control (HDLC) – является собственностью Cisco и используется для соединения двух устройств Cisco.

X.25/LAPB – это стандарт, который определяет способ соединения между DTE и DCE устройствами при удалённом терминальном доступе и компьютерных коммуникациях в публичных сетях данных.

Frame Relay – высокопроизводительный протокол, используемый на многих сетевых интерфейсах

Рисунок показывает, какие протоколы связи данных используются в каждой из трёх типов соединений в WAN

Frame Relay

В настоящее время Frame Relay заменяет X.25 как технология переключения пакетов. Стандартизированный в 1990, Frame Relay упрощает функции канального уровня и обеспечивают только проверку на ошибки, но не их исправление. Такой малоизбыточный подход к коммутации пакетов увеличивает производительность и эффективность. Современные волоконнооптические каналы связи и цифровые средства обслуживания передачи обеспечивают намного меньше ошибок, чем их медные предшественники. По этой причине, использование как в X.25 механизмов надёжности на канальном уровне теперь вообще расценивается как ненужное. Исправление ошибок осуществляется в протоколах более высокого уровня, например TCP.

Frame Relay - ITU и ANSI стандарт. ITU-T - International Telecommunications Union (Международный Союз Телесвязи) и ANSI - American National Standards Institute (Американский Национальный Институт Стандартов). Стандарты определяют процесс для передачи данных по сети с коммутацией пакетов. Frame Relay это ориентированная на соединение технология связи данных, которая оптимизирована, чтобы обеспечить высокое быстродействие и эффективность.

Современные сети телекоммуникаций характеризуются относительно свободной от ошибок цифровой передачей и высоконадёжной инфраструктурой. Frame Relay полагается почти полностью на протоколы верхнего уровня, чтобы обнаружить и исправить ошибки. Frame Relay не имеет механизмов повторной передачи, которые используются в X.25. Без механизмов исправления ошибки Frame Relay выигрывает у X.25 по скорости. В результате Frame Relay приемлем для использования там, где требуется высокая производительность, как в локальных сетях. Физическая сеть, на которой развёрнут Frame Relay, может быть или общественной или частной сетью и иметь различную физическую природу: оптика, спутниковые каналы связи, выделенные линии.

Frame Relay определяет процесс взаимосвязи между DTE клиента, типа маршрутизатор, и DCE провайдера. Как только трафик достигает коммутатора провайдера, Frame Relay не определяет способ, по которому данные передаются

в пределах сети Frame Relay. Поэтому, провайдер Frame Relay может использовать разнообразные технологий, типа ATM или PPP, чтобы перемещать данные с одного конца его сети к другому.

Устройства Frame Relay

Устройства, присоединённые к глобальной сети Frame Relay могут быть либо DTE либо DCE устройствами. Устройства DTE рассматриваются как окончательное оборудование и обычно располагаются на территории клиентов - потребителей услуг Frame Relay. Примерами DTE устройств являются маршрутизаторы и устройства доступа FRAD (Frame Relay Access Devices). FRAD это специальное устройство для связи между LAN и Frame Relay.

Внутрисетевые устройства DCE принадлежат провайдеру. Они обеспечивают синхронизацию и/или услуги коммутации пакетов в сети и обеспечивают действительную передачу данных в WAN.

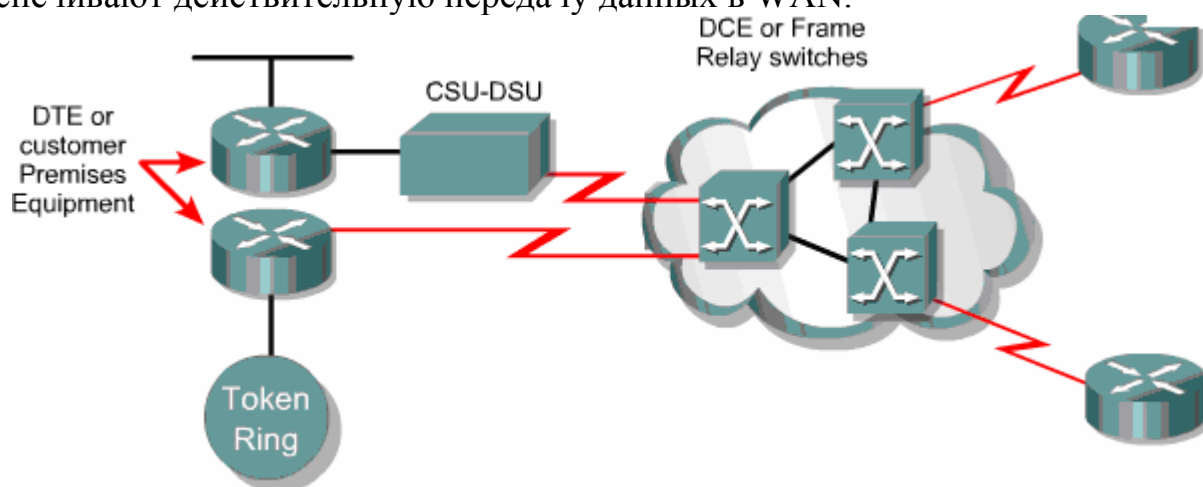


Рис. 4.

Сеть Frame Relay строится с помощью коммутаторов (switch) Frame Relay, выступающих в роли DCE. Внутри сети Frame Relay могут использоваться различные технологии передачи данных, например ATM. Физические каналы связи также не регламентируются - это может быть оптика, спутниковые каналы связи, выделенные линии.

Не взирая на технологию внутри сети Frame Relay, связь между потребителем и провайдером Frame Relay осуществляется по протоколу Frame Relay.

Функционирование Frame Relay

Обычно, чем больше расстояние покрывает выделенная линия, тем более дорога услуга. Поддержка полносвязного соединения удалённых сайтов с помощью выделенных линий слишком накладна для многих организаций. С другой стороны сети с коммутацией пакетов предоставляют способ мультиплексирования нескольких логических передач данных по единственной физической связи. Единственное соединение к сети с коммутацией пакетов провайдера будет менее дорогим, чем отдельные выделенные линии между потребителем и каждым удалённым сайтом. Сети с коммутацией пакетов

используют виртуальные цепи для доставки пакетов из конца в конец через разделяемую инфраструктуру.

Служба по коммутации пакетов, такая как Frame Relay требует, чтобы потребитель поддерживал только одну цепь, обычно T1, к центральному офису (ЦО) провайдера. Frame Relay обеспечивает огромную эффективность по стоимости, так как один сайт может соединиться со многими географически удалёнными сайтами, используя единственную линию T1 и одно DCE (CSU/DSU) устройство для подключения к локальному ЦО.

Для коммуникации между любыми двумя сайтами провайдер услуг должен установить виртуальную цепь между этими сайтами внутри сети Frame Relay. Хотя оплата идёт за каждую виртуальную цепь, эта плата невелика. Это делает Frame Relay идеальной технологией для создания полносвязной топологии.

Сети Frame Relay поддерживают как постоянные виртуальные цепи PVC (permanent virtual circuits) так и коммутируемые виртуальные цепи SVC (switched virtual circuits). PVC – наиболее типичны для Frame Relay. PVC являются постоянно установленными соединениями, которые используются, когда сети Frame Relay имеется устойчивый трафик между определёнными DTE устройствами.

SVC являются временными соединениями, используемыми при наличии единичного трафика между DTE устройствами. Так как они временны, соединение SVC требует установки и завершения для каждого соединения. Большинство провайдеров поддерживает только PVC.

В Frame Relay каждому концу виртуальной цепи назначается идентификатор соединения. Коммутационное оборудование провайдера поддерживает таблицу, отображающую эти идентификаторы на выходные порты. При получении фрейма коммутатор анализирует идентификатор и доставляет фрейм на соответствующий выходной порт.

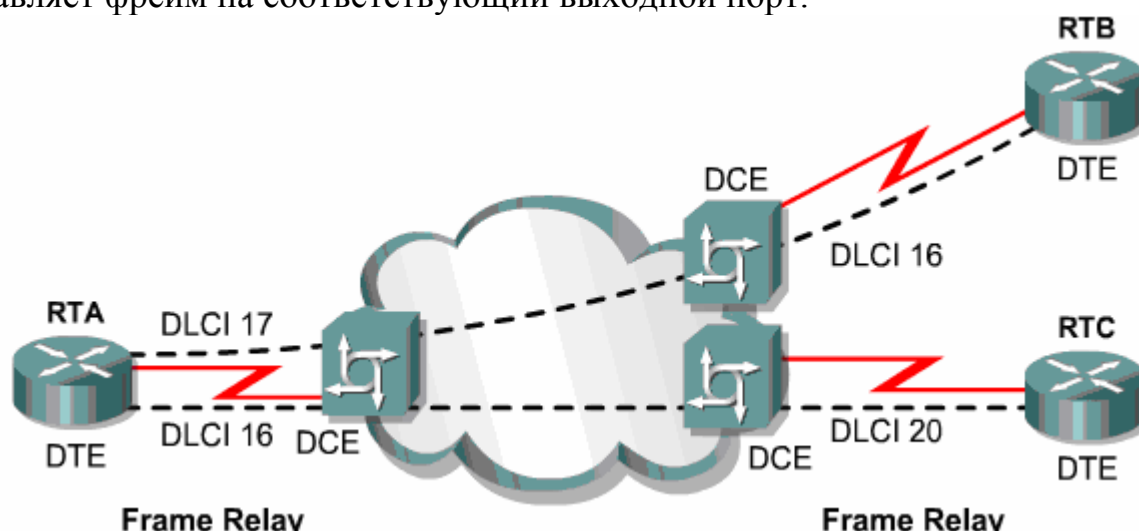


Рис. 5. Frame Relay DLCI

В сетях Frame Relay такой идентификатор называется DLCI (data-link connection identifier). Они идентифицирует виртуальную цепь. Для создания PVC коммутатор использует два DLCI для каждой пары DTE устройств (маршрутизаторы).

Два соединённые виртуальной цепью устройства DTE могут использовать различные значения DLCI для ссылки на одно и то же соединение. На рисунке 5 PVC, связывающее маршрутизаторы RTA и RTB, имеет DLCI равное 17, назначенное между RTA и непосредственно соединённым коммутатором. DLCI с номером 16 на RTB определяет то же PVC соединение между RTB и непосредственно соединённым коммутатором. Между тем, RTA использует DLCI 16 для ссылки на PVC, которое соединяется с RTC.

Для того, чтобы маршрутизатор RTA знал, какой PVC использовать на третьем сетевом уровне, IP адреса должны быть отображены в номера DLCI. Так на рисунке маршрутизатор RTA должен отобразить адреса третьего уровня в доступные DLCI.

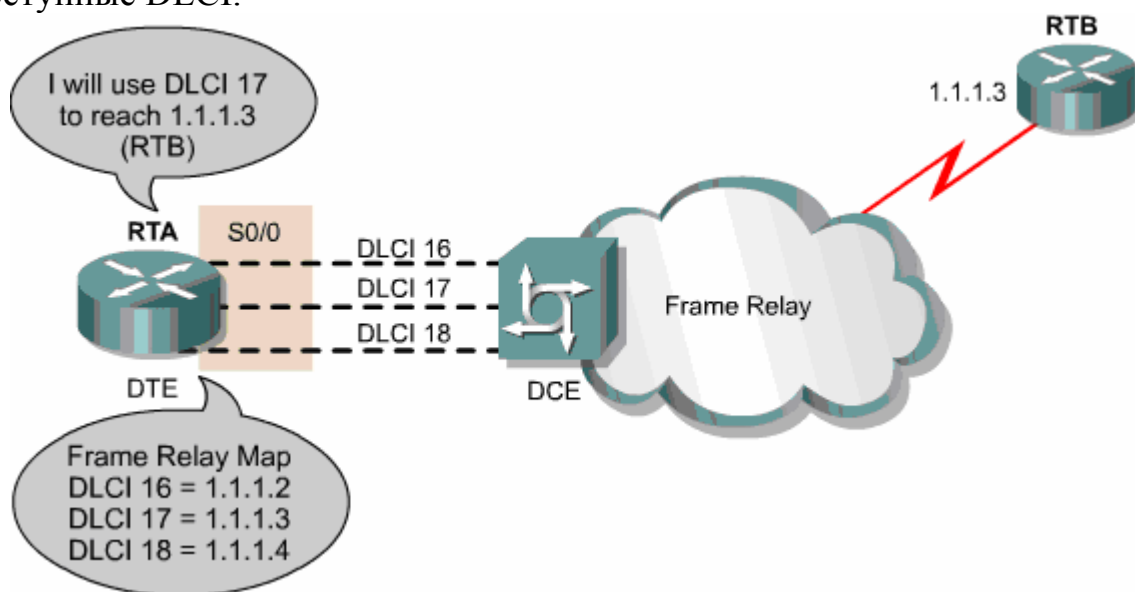


Рис. 6.

Например, RTA отображает IP адрес 1.1.1.3 маршрутизатора RTB на DLCI 17. Поскольку RTA знает, какой DLCI использовать, то для достижения получателя следует инкапсулировать IP пакет во фрейм Frame Relay, содержащий соответствующий номер DLCI.

Маршрутизаторы Cisco поддерживают два типа заголовков Frame Relay: cisco и ietf. Первый тип – для оборудования Cisco, второй – для устройств разных производителей.

Включив номер DLCI в заголовок Frame Relay, RTA может связываться как с RTB, так и с RTC по одной физической цепи. При таком статистическом мультиплексировании полоса пропускания автоматически выделяется для активных каналов. Если RTA не имеет пакетов для отправки на RTB, то RTA может использовать всю доступную полосу пропускания для связи с RTC. При TDM (Time-division multiplexing) мультиплексировании для каждого канала, вне зависимости от наличия в канале данных на передачу, выделяется определённая полоса.

Для организации WAN провайдер услуг Frame Relay назначает своим клиентам номера DLCI. Обычно, DLCI от 0 до 15 и от 1008 до 1023 резервируются для специальных целей. Клиентам провайдер услуг назначает

номера DLCI в диапазоне от 16 до 1007. Для широковещания можно использовать DLCI 1019 и 1020. Локальный интерфейс управления Local Management Interface (LMI) использует DLCI 1023 или 0. Некоторые провайдеры услуг Frame Relay могут разрешить своим клиентам выбрать собственные номера DLCI.

Для построения отображения номеров DLCI в адреса третьего уровня, маршрутизатор должен вначале знать, какие VC доступны. Обычно процесс определения доступных VC и их номеров DLCI осуществляется по стандарту LMI.

Как только маршрутизатору стали известны номера DLCI для доступных VC, он должен определить какие адреса третьего уровня отображать на какие номера DLCI. Отображение адресов может быть конфигурировано либо в ручную либо динамически. Вне зависимости от того, осуществляется ли отображение DLCI на удалённый IP адрес вручную или динамически, используемое DLCI не должно иметь одинаковых значений на обоих концах PVC.

Frame Relay LMI

LMI это сигнальный стандарт между DTE устройством (маршрутизатором) и DCE устройством (коммутатором Frame Relay). LMI отвечает за управление соединением между устройствами, проверяет, что данные передаются, периодически сообщает о появлении новых PVC и об уничтожении уже существующих PVC.

Сейчас существует три несовместимых реализации LMI cisco, ansi и q933a.

При использовании Cisco IOS выпуска 11.2 и позже, маршрутизатор пытается автоматически определить тип LMI, используемый коммутатором провайдера

Инверсный ARP

С помощью приемлемых конфигурационных команд номер DLCI может быть вручную отображён в адрес третьего уровня. В сложных сетях построение статического отображения может потребовать больших усилий, и такое отображение не приспособлено к изменению топологии Frame Relay. С помощью LMI коммутатор Frame Relay может уведомить маршрутизаторы о DLCI новой виртуальной цепи. Это уведомление не содержит адрес 3 уровня. Станция, получившая уведомление будет знать о новом соединении, но не будет иметь возможность адресовать другую сторону. Без новой конфигурации или механизма определения адреса другой стороны, новая виртуальная цепь не может быть использована.

Инверсный протокол разрешения адреса (Inverse Address Resolution Protocol (Inverse ARP)) был развит для обеспечения механизма динамического отображения DLCI на адрес третьего уровня. Инверсное ARP работает во многом также как и ARP в LAN. Однако в ARP устройство знает удалённый IP адрес и нуждается в определении MAC адреса удалённого устройства. В Inverse

ARP, маршрутизатор знает адрес 2 уровня, которым является DLCI, но нуждается в определении удалённого IP адреса.

Как только маршрутизатор получил от коммутатора информацию о доступных PVC и их DLCI, он может послать запрос инверсного ARP на другой конец каждого PVC об его адресе третьего уровня. В то же время этот запрос снабжает удалённую систему адресом третьего уровня локальной системы. Информация, принятая от инверсного ARP, используется для построения отображения Frame Relay на IP.

Если в маршрутизаторе Cisco интерфейс конфигурируется на использование инкапсуляции Frame Relay, то инверсный ARP включается автоматически. Если для определённого интерфейса конфигурируется статическое отображение, то Inverse ARP автоматически отключается для данного протокола и данного DLCI. Если маршрутизатор на другом конце не поддерживает инверсный ARP, то следует использовать статическое отображение.

Конфигурация инкапсуляции в Frame Relay

При конфигурировании последовательного интерфейса маршрутизатора для подключения к Frame Relay должна быть определена инкапсуляция Frame Relay. Имеется две возможные инкапсуляции - ietf и cisco. На устройствах cisco по умолчанию используется инкапсуляция cisco. Метод cisco является собственностью компании и не может быть использован, если маршрутизатор cisco соединён по сети Frame Relay с оборудованием другого производителя.

Для базовой конфигурации Frame Relay в Cisco IOS версии старше 11.1, использующей инверсное ARP и автоопределение типа LMI, следует лишь задать адрес третьего уровня и установить инкапсуляцию в Frame Relay

```
Router(config-if)#encapsulation frame-relay {cisco | ietf}
```

Если используется Cisco IOS версии 11.1 или раньше, то необходимо задать тип LMI

```
Router(config-if)#frame-relay lmi-type {ansi | cisco | q933a }
```

Важно помнить, что провайдер Frame Relay услуг создаёт виртуальную цепь внутри сети Frame Relay, соединяющую два удалённых сетевых устройства пользователя, как правило, маршрутизатора. Как только маршрутизатор и коммутатор Frame Relay, к которому он подсоединён, завершают обмен информацией LMI, сеть Frame Relay имеет всё необходимое для создания виртуальной цепи к другому удалённому маршрутизатору. Сеть Frame Relay не похожа на Интернет, где любые два устройства, подсоединённые к нему, могут связаться между собой. В сети Frame Relay до того как два маршрутизатора смогут обмениваться информацией, провайдер услуг Frame Relay должен заранее установить между ними виртуальную цепь.

Конфигурация отображений Frame Relay

Если используется динамическое отображение адресов, то для каждого активного PVC служба инверсного ARP маршрутизатора запрашивает IP адрес у маршрутизатора следующего хопа. Как только запрашивающий

маршрутизатор получает отклик инверсного ARP, он обновляет таблицу отображения адресов третьего уровня в номера DLCI. Для устройств cisco динамическое отображение адресов включено по умолчанию. Если оборудование Frame Relay поддерживает инверсное ARP и автоопределение типа LMI, то динамическое отображение адресов имеет место автоматически. Следовательно, не требуется статического отображения адресов.

Если оборудование не поддерживает инверсное ARP и автоопределение типа LMI, то статическое отображение должно быть настроено вручную с помощью команды `frame-relay map`. Как только для данной DLCI задаётся статическое отображение, на этом служба инверсного ARP отключается.

Для конфигурации статического отображения используется следующий синтаксис

```
Router(config-if)#frame-relay map protocol protocol-address dlci  
[broadcast] [ietf | cisco],
```

где `protocol` – `appletalk`, `clns`, `decnet`, `ip`, `xns`, `vines`, `dlci` – номер DLCI, `ietf | cisco` – определяет тип инкапсуляции, по умолчанию - `cisco`.

Опция `broadcast` - означает широковещательную передачу. Используется при настройке протоколов маршрутизации и позволяет рассматривать сети с множественным доступом и без широковещания (какой и является Frame Relay) во многом также как и широковещательные сети с множественным доступом (LAN). Например

```
Router(config)#interface Serial0
```

```
Router(config-if)#ip address 1.1.1.1 255.255.255.0
```

```
Router(config-if)#encapsulation frame-relay
```

```
Router(config-if)# frame-relay map ip 1.1.1.2 110 broadcast cisco
```

Здесь 1.1.1.1 локальный, а 1.1.1.2 удалённый IP адрес.

Если на последовательном интерфейсе конфигурируется инкапсуляция Cisco, то эта инкапсуляция применяется ко всем VC на этом интерфейсе. Если на интерфейсе взаимодействует оборудование Cisco и неCisco, то следует выборочно задать инкапсуляцию IETF.

Проверка конфигурации интерфейса Frame Relay

После конфигурации Frame Relay следует проверить, что соединения активны. Это осуществляется с помощью нескольких команд `show`:

Команда **`show interfaces serial`** показывает информацию об инкапсуляции и статусе протоколов первого и второго уровня, а также о широковещательном DLCI, о всех используемых в последовательном интерфейсе номерах DLCI и о DLCI, используемом для LMI.

Команда **`show frame-relay pvc`** показывает статус каждого конфигурированного соединения и статистику трафика. Команда показывает статус всех PVC, конфигурированных на маршрутизаторе.

Команда **`show frame-relay map`** показывает элементы текущего отображения адресов и информацию о соединениях.

Команда **show frame-relay lmi** показывает статистику трафика LMI: число сообщений о статусе, которыми обменялись локальный маршрутизатор и коммутатор Frame Relay.

Топологии Frame Relay

Frame Relay разрешает взаимодействие удалённых сайтов несколькими способами

Звезда, иногда называют hub and spoke

Полносвязная (Full mesh)

Частично связанная (Partial mesh)

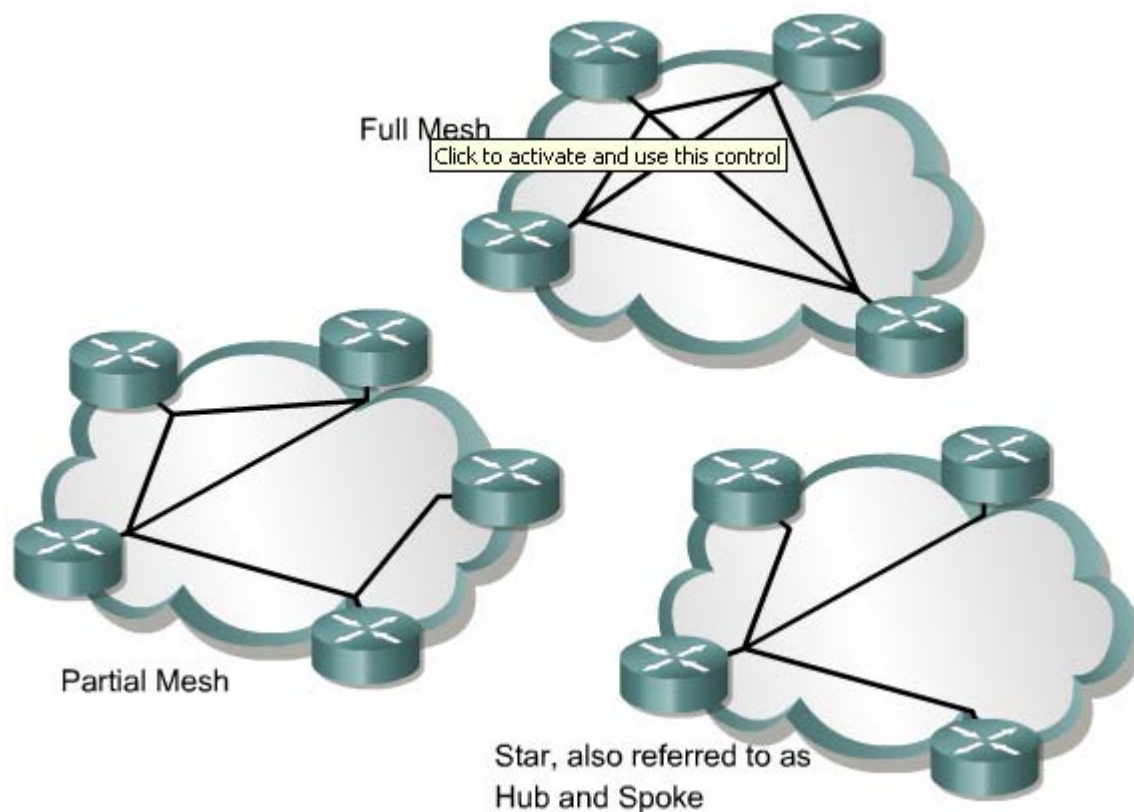


Рис. 7.

Топология звезда наиболее популярна из-за своей стоимостной эффективности. В этой топологии удаленные сайты подсоединяются к центральному сайту, который предоставляет услуги и службы. Это наименее дорогая топология требующая минимального числа PVC. Обычно центральный маршрутизатор использует один интерфейс для связи со многими PVC.

В полносвязной топологии все маршрутизаторы имеют PVC ко всем маршрутизаторам. Топология избыточна, но надёжна: если какое-то соединение отключится, маршрутизаторы могут изменить маршруты. С ростом числа узлов стоимость резко возрастает.

В частично связанной топологии не все сайты связаны напрямую. Наличие PVC между маршрутизаторами обуславливается объёмом трафика, архитектурой сети или другими соображениями.

Конфигурирование подынтерфейсов для Frame Relay.

Подынтерфейсы есть логическое подразделение физического интерфейса. С помощью конфигурации подынтерфейсов каждый PVC может быть настроен как соединение точка-точка. Это позволяет каждому подынтерфейсу действовать как арендованная линия. Это потому, что каждый подынтерфейс точка-точка рассматривается как отдельный физический интерфейс.

В случае точка-точка используется единственный подынтерфейс для установления PVC соединения с другим физическим интерфейсом или подынтерфейсом на удалённом компьютере. В этом случае каждая пара подынтерфейсов может находиться в своей собственной подсети и каждый подынтерфейс имеет единственный DLCI.

Для конфигурирования подынтерфейсов на физическом интерфейсе назначается инкапсуляция Frame Relay (cisco либо ietf). Если физическом интерфейс уже имеет IP адрес, его надо удалить, так как каждый подынтерфейс имеет собственный IP адрес. Если физический интерфейс имеет адрес, то фреймы не будут получены локальными подынтерфейсами:

```
RTA(config)#interfaces0/0
```

```
RTA(config-if)#encapsulation frame-relay ietf
```

Далее, определите подынтерфейсы, используя следующие команды.

```
Router(config-if)#interface serial number subinterface-number {multipoint | point-to-point}
```

Следующая команда создаёт подынтерфейс 2 типа точка-точка на Serial 0/0:

```
RTA(config)#interface serial s0/0.2 point-to-point
```

Следующая команда создаёт подынтерфейс 5 типа multipoint на Serial 2/0:

```
RTA(config)#interface serial s2/0.5 multipoint
```

Заметим, что после введения этих команд операционная система IOS изменяет строку приглашения на **config-subif**, означающую режим конфигурации подынтерфейса.

Номера подынтерфейсов могут быть назначены в режиме конфигурации подынтерфейса или в глобальном конфигурационном режиме в диапазоне от 1 до 4294967295. При конфигурации подынтерфейса точка-точка, обычной практикой является номеровать подынтерфейс согласно значению DLCI данного PVC.

После создания подынтерфейса следует задать IP адрес

```
RTA(config-subif)#ip address 2.1.1.1 255.255.255.0
```

Далее либо конфигурируется статическое отображение Frame Relay либо используется команда **frame-relay interface-dlci** для асоциации подынтерфейса с DLCI. Эта команда требуется для всех подынтерфейсов точка-точка. Она также требуется для подынтерфейсов multipoint с разрешенным режимом инверсного ARP. Она не требуется для подынтерфейсов multipoint, которые конфигурируются с помощью статических отображений маршрутов.

Некий коммутатор Frame Relay на рисунке 8 использует LMI для информирования RTA, что доступны три активных PVC с DLCI номерами 18, 19, and 20.

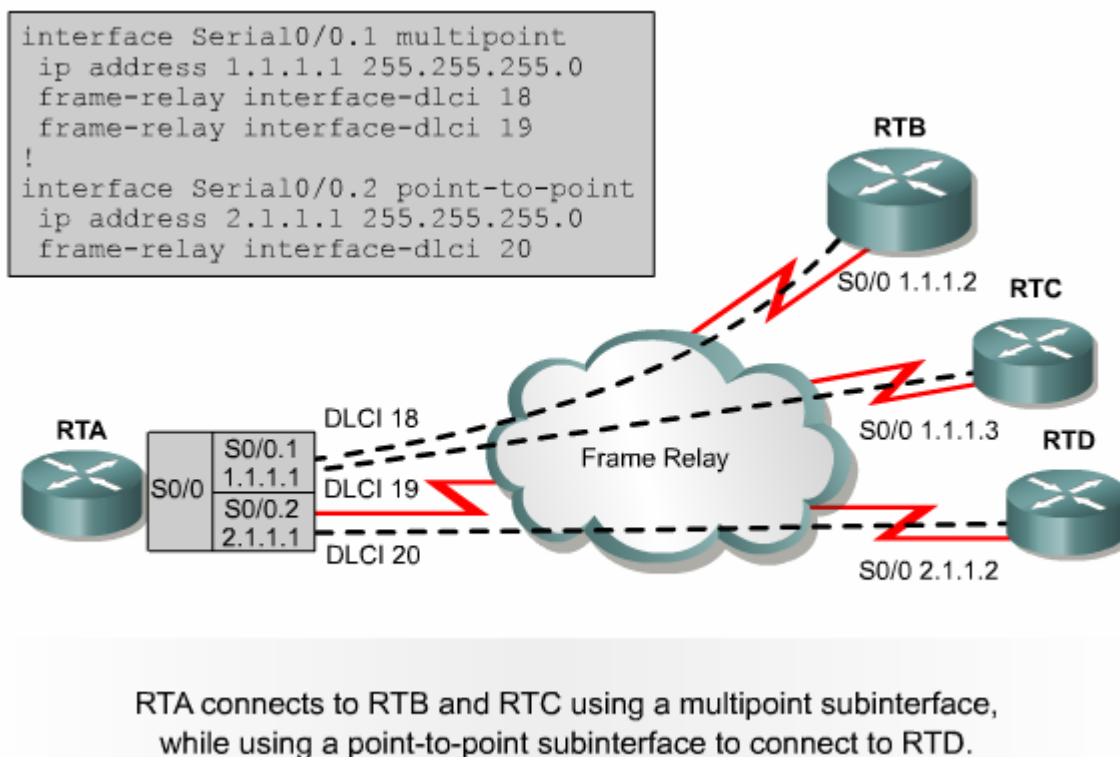


Рис. 8.

Когда RTA узнает о DLCI 18, 19 и 20 на интерфейсе S0/0, RTA не знает какой DLCI использовать с каким подинтерфейсом. Это потому, что LMI не предоставляет способа извещения RTA о том, что DLCI 20 должен быть использован интерфейсом S0/0.2, а не S0/0.1. Следовательно каждый подинтерфейс должен быть вручную ассоциирован с приемлемым номером DLCI.

Режим **multipoint** поддерживается симулятором не полностью и рассматриваться не будет.

Практическая часть

Босон дизайнер всегда создаёт полносвязную топологию Frame Relay. Для создания сетевой топологии с использованием Frame Relay, вначале надо определиться, какие маршрутизаторы и через какие свои последовательные интерфейсы будут подсоединены к Frame Relay. Далее нажмите правой кнопкой мыши на одном из таких маршрутизаторов, который будет присоединён к Frame Relay, и выберите пункты контекстного меню Add connection затем нужный интерфейс serial. В появившемся модальном окне выделите радиокнопку Point-to-Multy-Point Serial Connection (Frame Relay). Нажмите кнопку Next. Появится новое модальное окно. Выберите в первом списке первый маршрутизатор, который вы хотите подсоединить к Frame Relay, во втором списке выберите его интерфейс, которым он будет соединён к Frame Relay и нажмите кнопку Add. Выберите в первом списке второй маршрутизатор который вы хотите подсоединить к Frame Relay, во втором списке выберите его интерфейс, которым он будет соединён к Frame Relay и нажмите кнопку Add.

Так повторите для всех маршрутизатор. Нажмите кнопку Connect. Дизайнер создаст полносвязную топологию, определив PVC и DLCI. Появится окно с назначенными DLCI. Можно их отредактировать. Рекомендуется с ними можно согласиться, нажав ОК. В последствии можно изменить значения DLCI, нажав правой кнопкой мыши на облаке Frame Relay.

Симулятор не в полной мере поддерживает Frame Relay.

1. Создаём топологию на 805 маршрутизаторах

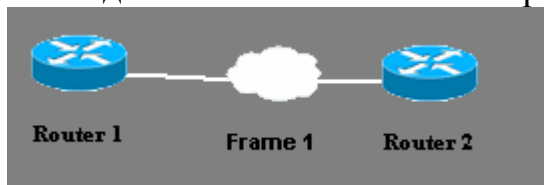


Рис. 9.

Конфигурируем Frame Relay

TO ROUTER					
F R O M R O U T E R		Router 1	Router 2		
	Router 1		102		
	Router 2	201			

Рис. 10.

1.1 Динамическое назначения адресов

```

router1(config)# interface serial0
router1(config-if)# encapsulation frame-relay
router1(config-if)# ip address 215.10.1.1 255.255.255.0
router1(config-if)# no frame-relay inverse-arp
router1(config-if)# no shut
router2(config)# interface serial0
router2(config-if)# encapsulation frame-relay
router2(config-if)# ip address 215.10.1.2 255.255.255.0
router2(config-if)# no shut

```

Введём команду **show interfaces serial0** на обоих маршрутизаторах. Она покажет подсоединены ли вы к Frame Relay. Вы должны увидеть фразы 'up and line protocol is up' и 'DTE LMI up'.

Введём команду **show frame-relay map** на обоих маршрутизаторах. На router1 она ничего не покажет, так как отключен **inverse-arp**. На router2 она покажет отображение локального DLCI на удалённый IP адрес 201-215.10.1.1.

Команда **show frame-relay pvc** должна показать статус PVC 'ACTIVE' на router2 и статус PVC 'INACTIVE' на router1.

Включим **inverse-arp**

```

router1(config-if)#frame-relay inverse-arp

```

Теперь на router1 она покажет отображение локального DLCI на удалённый IP адрес 102-215.10.1.2. Команда **show frame-relay pvc** должна показать статус PVC 'ACTIVE' на router1. Пропингуйте оба маршрутизатора.

1.2 Статическое назначения адресов симулятор версии 5.3 не поддерживает. Симулятор версии 6.0 Final beta поддерживает.

Загрузите прежнюю топологию из рисунков 9 и 10.

и создайте новую конфигурацию

```
router1(config)# interface serial0
router1(config-if)# encapsulation frame-relay
router1(config-if)# ip address 215.10.1.1 255.255.255.0
router1(config-if)# frame-relay map ip 215.10.1.2 102 broadcast
router1(config-if)# no shut
```

```
router2(config)# interface serial0
router2(config-if)# encapsulation frame-relay
router2(config-if)# ip address 215.10.1.2 255.255.255.0
router2(config-if)# frame-relay map ip 215.10.1.1 201 broadcast
router2(config-if)# no shut
```

Введите на обоих маршрутизаторах команды "show interfaces serial0", "show frame-relay map", "show frame-relay pvc" и "show frame-relay lmi". Пропингуйте оба маршрутизатора.

1.3 Использование подынтерфейсов

Загрузите прежнюю топологию из рисунков 9 и 10 и создайте новую конфигурацию

```
router1(config)# interface serial0
router1(config-if)# encapsulation frame-relay
router1(config-if)# no shut
router1(config-if)# interface serial0.102 point-to-point
router1(config-subif)# ip address 215.10.1.1 255.255.255.0
router1(config-subif)# frame-relay interface-dlci 102
router2(config)# interface serial0
router2(config-if)# no shut
router2(config-if)# encapsulation frame-relay
router2(config-if)# interface serial0.201 point-to-point
router2(config-subif)# ip address 215.10.1.2 255.255.255.0
router2(config-subif)# frame-relay interface-dlci 201
```

Введите на обоих маршрутизаторах команду **show ip interface brief**. Вы должны увидеть назначенные адреса и активные состояния интерфейсов (status = up и up). Заметим, что команда "show frame-relay map" не показывает адреса для подинтерфейсов, а выводит соотношение подинтерфейс – DLCI. Эту же информацию выводит и команда "show frame-relay pvc". Пропингуйте оба маршрутизатора.

2. Создадим топологию глобальной сети фирмы из трёх филиалов. Не теряя общности в качестве моделей локальных сетей филиалов используем по одному компьютеру.

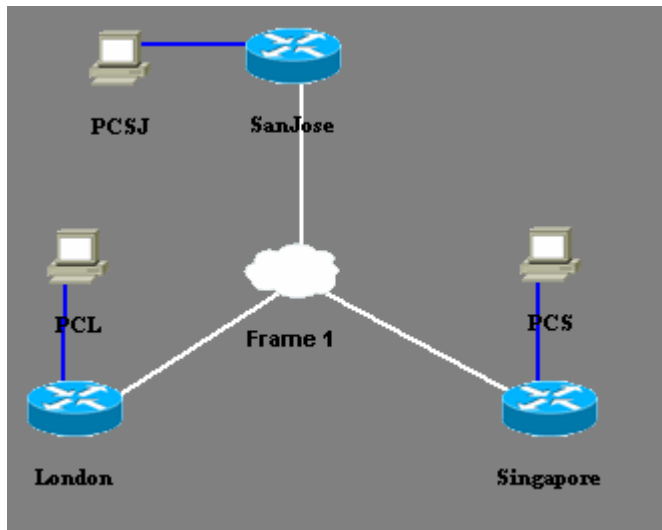


Рис. 11.

Назначим DLCI согласно рисунку

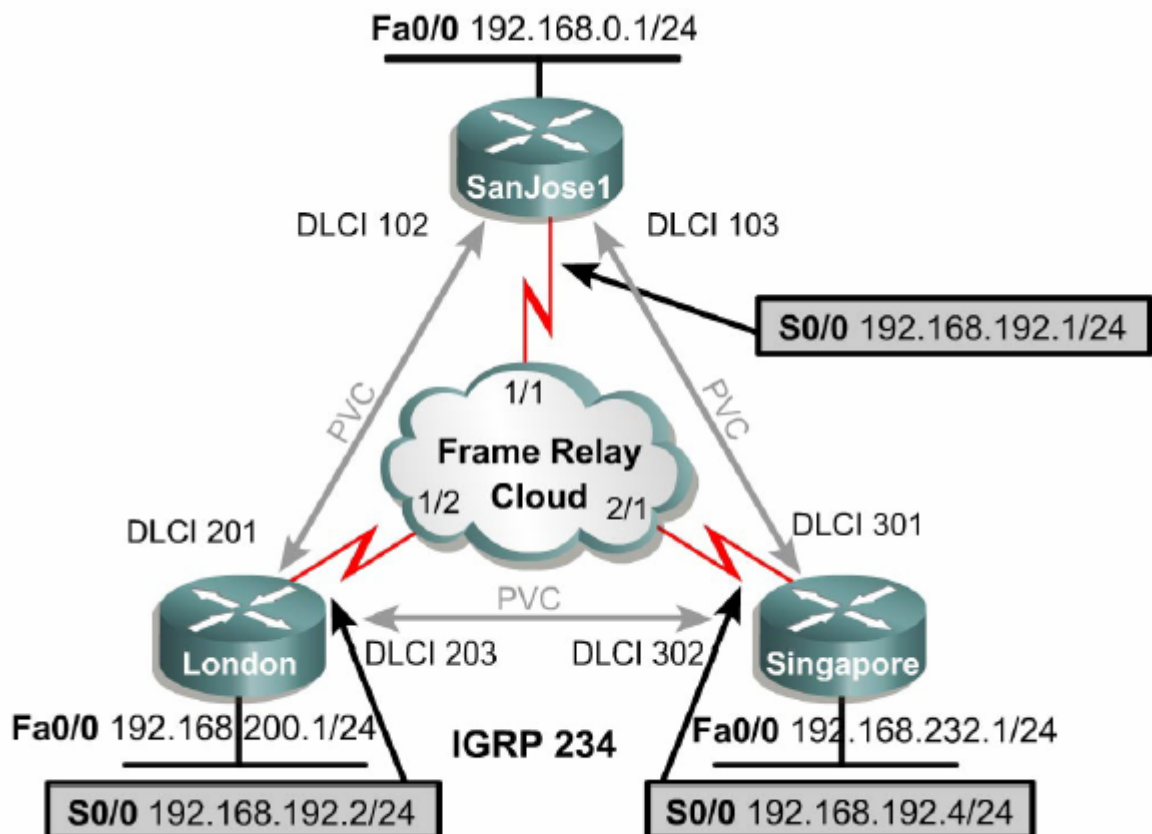


Рис 12.

Конфигурируем три маршрутизатора для Frame Relay для связи трёх филиалов фирмы по полносвязной топологии. В качестве протокола маршрутизации используем OSPF.

2.1. Динамическая настройка

На маршрутизаторе SanJose1

```
Router(config)#hostname SanJose
SanJose(config)#interface Serial0
SanJose(config-if)#ip address 192.168.192.1 255.255.255.0
SanJose(config-if)#encapsulation frame-relay
SanJose (config-if)# no shut
SanJose(config-if)#interface Ethernet0
SanJose(config-if)#ip address 192.168.0.1 255.255.255.0
SanJose (config-if)# no shut
SanJose(config-if)#exit
SanJose(config)#router ospf 100
SanJose(config-router)#network 192.168.0.0 0.0.255.255 area 1
```

На маршрутизаторе London

```
Router(config)#hostname London
London(config)#interface Serial0
London(config-if)#ip address 192.168.192.2 255.255.255.0
London(config-if)#encapsulation frame-relay
London (config-if)# no shut
London(config-if)#interface Ethernet0
London(config-if)#ip address 192.168.200.1 255.255.255.0
London (config-if)# no shut
London(config-if)# exit
London(config)#router ospf 100
London(config-router)#network 192.168.0.0 0.0.255.255 area 1
```

На маршрутизаторе Singapore

```
Router(config)#hostname Singapore
Singapore(config)#interface Serial0
Singapore(config-if)#ip address 192.168.192.4 255.255.255.0
Singapore(config-if)#encapsulation frame-relay
Singapore (config-if)# no shut
Singapore(config-if)#interface Ethernet0
Singapore(config-if)#ip address 192.168.232.1 255.255.255.0
Singapore (config-if)# no shut
Singapore (config-if)# exit
Singapore(config)#router ospf 100
Singapore(config-router)#network 192.168.0.0 0.0.255.255 area 1
```

Введите на всех маршрутизаторах команды “show ip interface brief” и “show interfaces serial0” . Вы можете увидеть состояния интерфейсов. Введите

на обоих маршрутизаторах команды “show frame-relay pvc” Вы можете увидеть состояния каналов frame relay (pvc status). Например

Singapore#show frame-relay pvc

PVC Statistics for interface Serial0 (Frame Relay DTE)

DLCI = 301, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0

input pkts 83	output pkts 102	in bytes 9360
out bytes 3551	dropped pkts 21	in FECN pkts 22
in BECN pkts 18	out FECN pkts 29	out BECN pkts 113
in DE pkts 98	out DE pkts 114	
out bcast pkts 65	out bcast bytes 537	
pvc create time 00:32:04, last time pvc status changed 00:32:05		

PVC Statistics for interface Serial0 (Frame Relay DTE)

DLCI = 302, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0

input pkts 93	output pkts 11	in bytes 2025
out bytes 2818	dropped pkts 7	in FECN pkts 64
in BECN pkts 11	out FECN pkts 81	out BECN pkts 45
in DE pkts 35	out DE pkts 9	
out bcast pkts 56	out bcast bytes 3868	
pvc create time 00:32:04, last time pvc status changed 00:32:05		

Мы видим активные каналы на SanJose (DLCI=301) и London (DLCI=302).

В реальных сетях Frame Relay состояния каналов могут быть ACTIVE, INACTIVE и DELETED. ACTIVE это работоспособная цепь от конца в конец (DTE к DTE). Состояние INACTIVE показывает успешное подключение к коммутатору Frame Relay (DTE к DCE) при отсутствии маршрутизатора (DTE) на другом конце PVC. Это состояние случается, когда маршрутизатор на другом конце PVC либо не участвует в Frame Relay для нашей сети либо не настроен. Состояние DELETED случается когда маршрутизатор (DTE) и коммутатор Frame Relay не договорились о параметрах Frame Relay на этом PVC.

Симулятор версии 5.3 не в полной мере поддерживает состояния.

Команда show frame-relay map покажет динамически назначенное отображение «внешний IP адрес»-«локальный dlcі для канала pvc на удалённый маршрутизатор с этим внешним IP адресом». Например

Singapore#show frame-relay map

```
Serial0 (up): ip 192.168.192.1 dlcі 301(0x66,0x1860), dynamic,
              broadcast,CISCO, status defined, active
Serial0 (up): ip 192.168.192.2 dlcі 302(0x66,0x1860), dynamic,
              broadcast,CISCO, status defined, active
```

Мы видим адреса других маршрутизаторов. Как маршрутизатор Singapore их узнал? Когда маршрутизатор Singapore получает по протоколу LMI список DLCI, он посылает инверсные ARP запросы к каждому своему обнаруженному PVC каналу. Маршрутизатор на другом конце PVC возвращает свой IP адрес.

Из таблицы мы видим, что широковещание broadcast разрешены, что позволяет протоколам маршрутизации обмениваться маршрутной

информацией. Используя команду `show ip route`, убедитесь, что из Лондона есть маршрут на Сингапур

London#**show ip route**

```
C      192.168.192.0/24 is directly connected, Serial0
C      192.168.200.0/24 is directly connected, Ethernet0
D      192.168.0.0/24 [110/65] via 192.168.192.1, 00:00:28, Serial0
D      192.168.232.0/24 [110/65] via 192.168.192.4, 00:00:44, Serial0
```

Лондон имеет маршрут на сеть 192.168.232.0/24 Сингапура на адрес 192.168.192.4 через Serial0.

Мы видим, что только с **одной физической связью** к Frame Relay каждый маршрутизатор имеет **по две логических связи**.

Сконфигурируем локальные сети филиалов

PC	IP	Шлюз
PCSJ	192.168.0.2	192.168.0.1
PCL	192.168.200.2	192.168.200.1
PCS	192.168.232.2	192.168.232.1

Таблица 1.

Проверим полную конфигурацию с помощью расширенной команды `ping`. Например мы можем из маршрутизатора в SanJose проверить есть ли связь от филиального компьютера PCL в Лондоне к филиальному компьютеру PCS в Сингапуре.

SanJose#**ping**

Protocol [ip]:

Target IP address: 192.168.232.2

Repeat count [5]: 50

Datagram size [100]:

Timeout in seconds [2]:

Extended commands [n]:y

Source address or interface:192.168.200.2

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 50, 100-byte ICMP Echos to 192.168.232.2, timeout is 2 seconds:

!!

Success rate is 100 percent (50/50), round-trip min/avg/max = 1/2/4 ms Все

Пинги успешны.

2.2. Статическая настройка

Используйте симулятор версии 6.0 Final beta.

Сетевые инженеры могут быть не удовлетворены неявной динамической конфигурацией. Удобства динамического отображения DLCI - IP минимальны из-за статического характера большинства глобальных связей. Когда соединение установлено, оно не меняется годами.

Объединим филиалы в глобальную сеть статически. Для этого следует создать новую конфигурацию заново. Загрузите в симулятор прежнюю созданную топологию и настройте маршрутизаторы

SanJose

Router(config)#**hostname SanJose**

SanJose(config)#**interface Serial0**

SanJose(config-if)#**ip address 192.168.192.1 255.255.255.0**

SanJose(config-if)#**encapsulation frame-relay**

SanJose(config-if)#**no frame-relay inverse-arp**

SanJose(config-if)#**frame-relay map ip 192.168.192.2 102 broadcast**

SanJose(config-if)#**frame-relay map ip 192.168.192.4 103 broadcast**

SanJose (config-if)# **no shut**

SanJose(config-if)#**interface Ethernet0**

SanJose(config-if)#**ip address 192.168.0.1 255.255.255.0**

SanJose (config-if)# **no shut**

SanJose(config-if)#**exit**

SanJose(config)#**router ospf 100**

SanJose(config-router)#**network 192.168.0.0 0.0.255.255 area 1**

London

Router(config)#**hostname London**

London(config-if)#**interface Serial0**

London(config-if)# **ip address 192.168.192.2 255.255.255.0**

London(config-if)# **no frame-relay inverse-arp**

London(config-if)#**encapsulation frame-relay**

London(config-if)# **frame-relay map ip 192.168.192.1 201 broadcast**

London(config-if)#**frame-relay map ip 192.168.192.4 203 broadcast**

London (config-if)# **no shut**

London(config-if)#**interface Ethernet0**

London(config-if)#**ip address 192.168.200.1 255.255.255.0**

London (config-if)# **no shut**

London(config-if)#**exit**

London(config)#**router ospf 100**

London(config-router)#**network 192.168.0.0 0.0.255.255 area 1**

Singapore

Router(config)#**hostname Singapore**

Singapore (config)# **interface Serial0**

Singapore (config-if)# **ip address 192.168.192.4 255.255.255.0**

Singapore (config-if)# **encapsulation frame-relay**

Singapore (config-if)# **no frame-relay inverse-arp**

Singapore (config-if)# **frame-relay map ip 192.168.192.1 301 broadcast**

Singapore (config-if)# **frame-relay map ip 192.168.192.2 302 broadcast**

Singapore (config-if)# **no shut**

Singapore (config-if)# **interface Ethernet0**

Singapore (config-if)# **ip address 192.168.232.1 255.255.255.0**

Singapore (config-if)# **no shut**

Singapore (config-if)# **exit**

Singapore (config)# **router ospf 100**

Singapore (config-router)# **network 192.168.0.0 0.0.255.255 area 1**

Мы использовали команду **no frame-relay inverse-arp**, так как без неё симулятор отображает адреса как статически так и динамически.

Команда **frame-relay map** асоциирует IP адрес следующего хопа с локальным DLCI. Ключевое слово **broadcast** разрешает прохождение широковещательного трафика, например для информации о маршрутах для протоколов маршрутизации.

Введите на всех маршрутизаторах команды “**show ip interface brief**” и “**show interfaces serial0**”. Вы можете увидеть состояния интерфейсов. Введите на обоих маршрутизатора команды “**show frame-relay pvc**” и “**show frame-relay lmi**”. Вы можете увидеть состояния frame relay. Команда **show frame-relay map** покажет статически назначенное отображение «внешний IP адрес»-«локальный dlcі для канала pvc на удалённый маршрутизатор с этим внешним IP адресом». Вы должны увидеть слово **static**. Например

```
SanJose)# show frame-relay map
Serial0 (up): ip 192.168.192.2 dlcі 102(0x66,0x1860), static,
               broadcast,CISCO, status defined, active
Serial0 (up): ip 192.168.192.4 dlcі 103(0x66,0x1860), static,
               broadcast,CISCO, status defined, active
```

Используя команду **show ip route**, убедитесь, что из Лондона есть маршрут на Сингапур. К сожалению у симулятора версии 6.0 Final beta проблемы с маршрутизацией.

Сконфигурируем локальные сети филиалов как в таблице 1.

Проверим полную конфигурацию с помощью расширенной команды **ping**. Например мы можем из маршрутизатора в SanJose проверить есть ли связь от филиального компьютера PCL в Лондоне к филиальному компьютеру PCS в Сингапуре.

Все пинги должны быть успешны. В частности между всеми PC.

2.3 Использование подинтерфейсов

На симуляторе версии 6 final beta не работает динамическая маршрутизация. Работаем с симулятором версии 5.31.

Для улучшения возможностей настройки соединений сетевые инженеры вначале использовать индивидуальные физические связи между каждым двумя филиалами. Однако общая стоимость проекта оказалась чрезмерной. Решено было использовать подинтерфейсы и Frame Relay. Это позволяет создать отдельный логический канал между каждым двумя филиалами. Каждый маршрутизатор имеет по-прежнему одну физическую линию для соединения с Frame Relay, но для каждого последовательного интерфейса настраивается по два подинтерфейса. Становится возможным использование нескольких каналов PVC для каждого физического интерфейса маршрутизатора. Общая стоимость проекта оказывается приемлемой.

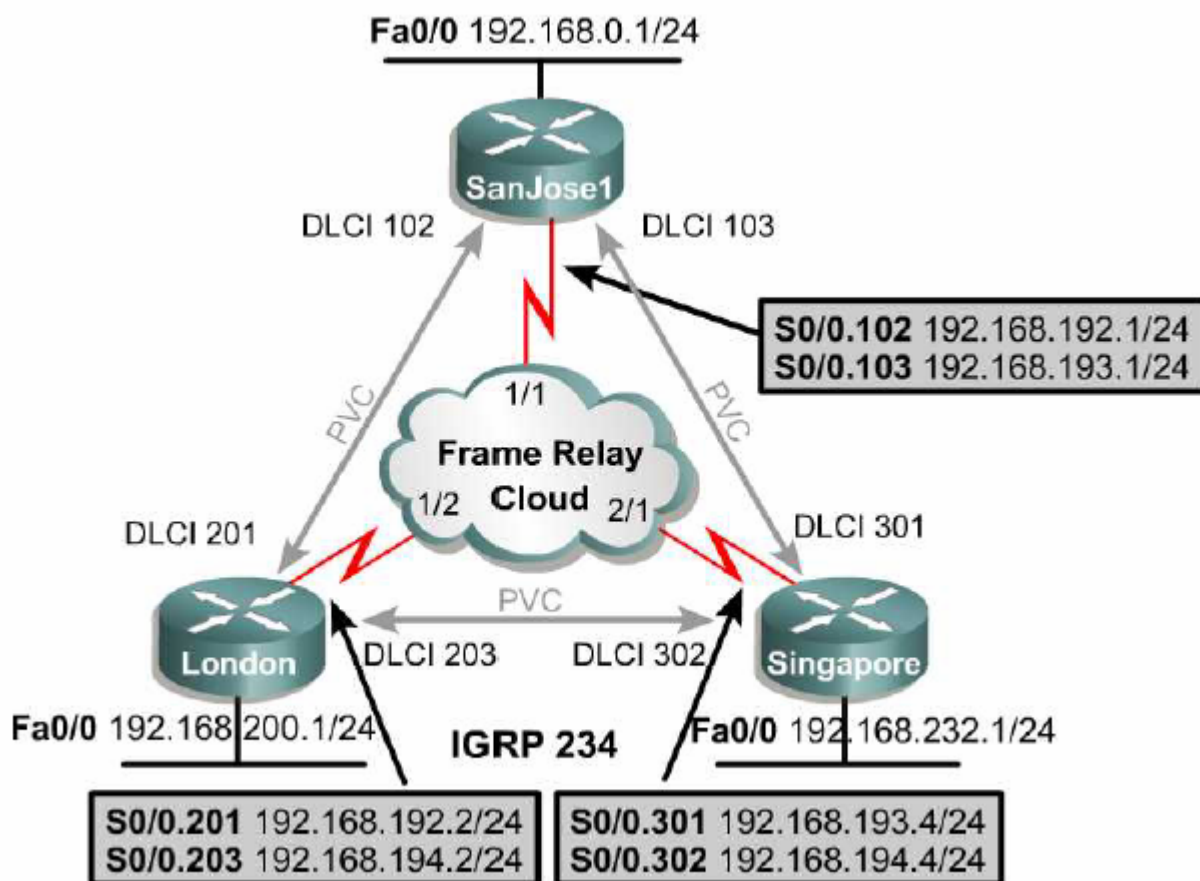


Рис. 13.

Для этого следует создать новую конфигурацию заново. Перегрузите симулятор и загрузите прежнюю созданную топологию (рисунок 11). Настройте на маршрутизаторах подынтерфейсы согласно рисунку 13. Для каждого канала PVC используем отдельную подсеть

PVC	DLCI	Подсеть
SanJose-London	102 - 201	192.168.192.0/24
SanJose-Singapoure	103 - 301	192.168.193.0/24
London-Singapoure	203 - 302	192.168.194.0/24

Таблица 2.

Для этого используйте команды. На маршрутизаторе SanJose1

```
Router(config)#hostname SanJose
```

```
SanJose(config)#interface serial 0
```

```
SanJose(config-if)#encapsulation frame-relay
```

```
SanJose1(config-if)#no shutdown
```

IP адрес на весь интерфейс не назначаем. Адреса назначаем на каждый подинтерфейс.

Интерфейсы Frame Relay могут быть переведены либо в режим точка-точка (point-to-point) либо в режим многоточка (multipoint). Конфигурируем подинтерфейс точка-точка, используя DLCI. Обычной практикой является использование номера DLCI как номера подинтерфейс. Тогда легче определять

какой канал PVC использует каждый подынтерфейс. Например, подынтерфейс 0.103 использует DLCI 103:

```
SanJose(config-if)#interface s0.103 point-to-point
SanJose(config-subif)#ip address 192.168.193.1 255.255.255.0
SanJose(config-subif)#frame-relay interface-dlci 103
```

Тщательно вводите в симуляторе команду `interface s0.103 point-to-point`. Используйте ? для выдачи подсказки после ввода очередного слова команды.

Команда `frame-relay interface-dlci` используется для определения, какой DLCI использует каждый подынтерфейс. Подынтерфейс точка-точка может использовать только один DLCI. Если подынтерфейс для DLCI не определён, то DLCI ассоциируется с главным интерфейсом `s0`.

```
Конфигурируем подынтерфейс для DLCI 102:
SanJose(config)#interface s0.102 point-to-point
SanJose(config-subif)#ip address 192.168.192.1 255.255.255.0
SanJose(config-subif)#frame-relay interface-dlci 102
Конфигурируем Ethernet0 и маршрутизацию
SanJose(config-if)#interface Ethernet0
SanJose(config-if)#ip address 192.168.0.1 255.255.255.0
SanJose(config-if)#exit
SanJose(config)#router ospf 100
SanJose(config-router)#network 192.168.0.0 0.0.255.255 area 1
```

Аналогично, используя рисунок 13, конфигурируем остальные маршрутизаторы. Лондон

```
Router(config)#hostname London
London(config)#interface Serial0
London(config-if)#encapsulation frame-relay
London(config-if)#no shutdown
London(config-if)#interface Serial0.201 point-to-point
London(config-subif)#ip address 192.168.192.2 255.255.255.0
London(config-subif)#frame-relay interface-dlci 201
London(config-subif)#interface Serial0.203 point-to-point
London(config-subif)#ip address 192.168.194.2 255.255.255.0
London(config-subif)#frame-relay interface-dlci 203
London(config-subif)#interface Ethernet0
London(config-if)#ip address 192.168.200.1 255.255.255.0
London(config-if)#no shutdown
London(config-if)#exit
London(config)#router ospf 100
London(config-router)#network 192.168.0.0 0.0.255.255 area 1
```

Сингапур

```
Router(config)#hostname Singapore
Singapore(config)#interface Serial0
Singapore(config-if)#encapsulation frame-relay
```

```

Singapore(config-if)#no shutdown
Singapore(config-if)#interface Serial0.301 point-to-point
Singapore(config-subif)#ip address 192.168.193.4 255.255.255.0
Singapore(config-subif)#frame-relay interface-dlci 301
Singapore(config-subif)#interface Serial0.302 point-to-point
Singapore(config-subif)#ip address 192.168.194.4 255.255.255.0
Singapore(config-subif)#frame-relay interface-dlci 302
Singapore(config-subif)#interface Ethernet0
Singapore(config-if)#ip address 192.168.232.1 255.255.255.0
Singapore(config-if)#no shutdown
Singapore(config-if)#exit
Singapore(config)#router ospf 100
Singapore(config-router)#network 192.168.0.0 0.0.255.255 area 1

```

Проверьте на каждом маршрутизаторе статус PVC командой `show frame-relay pvc`. Команд `show frame-relay map` в случае подинтерфейсов не двёт отображения DLCI на IP адрес. Каждый подинтерфейс рассматривается как отдельный физический интерфейс для каждого канала PVC. В этом случае каждый канал PVC содержит только два хоста и нет нужды идентифицировать следующий хоп.

Для проверки правильности назначения адресов и для того, чтобы убедиться, что все подинтерфейсы находятся в активном состоянии используйте команду **show ip interface brief**. Используя команду **show ip route**, убедитесь, что из Лондона есть маршрут на Сингапур

```
London#show ip route
```

```

Gateway of last resort is not set
C      192.168.192.0/24 is directly connected, 192.168.192.2
C      192.168.200.0/24 is directly connected, Ethernet0
C      192.168.194.0/24 is directly connected, 192.168.194.2
O      192.168.0.0/24 [110/1] via 192.168.192.1, 00:00:38, Serial0.201
O      192.168.193.0/24 [110/1] via 192.168.194.4, 00:00:15, Serial0.203
O      192.168.232.0/24 [110/1] via 192.168.192.1, 00:00:48, Serial0.201

```

Мы видим, что OSPF сработал так, что Лондон имеет маршрут на сеть 192.168.232.0/24 Сингапура через Serial0.201 на адрес 192.168.192.1. Маршрут, через через Serial0.203 на адрес 192.168.194.4 короче. А как у вас? Вопросы оптимизации маршрутов здесь не рассматриваются.

Сконфигурируем локальные сети филиалов согласно таблице 1.

Проверим полную конфигурацию с помощью расширенной команды `ping`. Например мы можем из маршрутизатора в SanJose проверить есть ли связь от филиального компьютера PCL в Лондоне к филиальному компьютеру PCS в Сингапуре.

Все пинги должны быть успешны. В частности между всеми РС.

3. Топология звезда

Трафик между Лондоном и Сингапуром оказался незначительным и было решено отказаться от канала PVC между ними и воспользоваться топологией звезда. Весь трафик между Лондоном и Сингапуром будет проходить через SanJose.

Загрузим в дизайнер старую топологию. Нажав в ней правой кнопкой на облаке Frame Relay, поменяем DLCI согласно рисунку, т.е. уничтожим PVC между Лондоном и Сингапуром.

	SanJose	London	Singapore
SanJose		102	103
London	201		0
Singapore	301	0	

Рис. 14.

Используем подинтерфейсы в режиме point-to-point
SanJose:

```
Router(config)#hostname SanJose
SanJose(config)#interface Serial0
SanJose(config-if)#encapsulation frame-relay
SanJose(config-if)#no shutdown
SanJose(config-if)#interface Serial0.103 point-to-point
SanJose(config-subif)#ip address 192.168.193.1 255.255.255.0
SanJose(config-subif)#frame-relay interface-dlci 103
SanJose(config-subif)#interface Serial0.102 point-to-point
SanJose(config-subif)#ip address 192.168.192.1 255.255.255.0
SanJose(config-subif)#frame-relay interface-dlci 102
SanJose(config-subif)#interface Ethernet0
SanJose(config-if)#ip address 192.168.0.1 255.255.255.0
SanJose(config-if)#no shutdown
SanJose(config-if)#exit
SanJose(config)#router ospf 100
SanJose(config-router)#network 192.168.0.0 0.0.255.255 area 1
```

London:

```
Router(config)#hostname London
London(config)# interface Serial0
London (config-if)#encapsulation frame-relay
London (config-if)#no shutdown
London(config-if)# interface Serial0.201 point-to-point
London(config-subif)# ip address 192.168.192.2 255.255.255.0
London(config-subif)# frame-relay interface-dlci 201
London(config-subif)# interface Ethernet0
```

```

London(config-if)# ip address 192.168.200.1 255.255.255.0
London (config-if)#no shutdown
London (config-if)#exit
London (config)#router ospf 100
London (config-router)#network 192.168.0.0 0.0.255.255 area 1

```

Singapore:

```

Router(config)#hostname Singapore
Singapore (config)#interface Serial0
Singapore (config-if)#encapsulation frame-relay
Singapore (config-if)# no shutdown
Singapore (config-if)# interface Serial0.301 point-to-point
Singapore (config-subif)# ip address 192.168.193.4 255.255.255.0
Singapore (config-subif)# frame-relay interface-dlci 301
Singapore (config-subif)# interface Ethernet0
Singapore (config-if)# ip address 192.168.232.1 255.255.255.0
Singapore (config-if)#no shutdown
Singapore (config-if)#exit
Singapore (config)# router ospf 100
Singapore (config-router)# network 192.168.0.0 0.0.255.255 area 1

```

Введите на всех маршрутизаторах команды **show ip interface brief** и **show interfaces serial0**. Вы можете увидеть состояния интерфейсов. Интерфейсы должны быть активны (UP)

Введите на обоих маршрутизаторах команды **show frame-relay pvc** и **show frame-relay lmi**. Вы можете увидеть состояния frame relay. Отметим отсутствие PVC между Лондоном и Сингапуром. Для подынтерфейсов команда **show frame-relay map** не покажет отображение адрес DLCI . Используйте **show ip interface brief**.

Используя команду **show ip route**, убедитесь, что из Лондона есть маршрут на Сингапур

```

London#show ip route
C      192.168.200.0/24 is directly connected, Ethernet0
C      192.168.192.0/24 is directly connected, 192.168.192.2
O      192.168.193.0/24 [110/1] via 192.168.192.1, 00:00:39, Serial0.201
O      192.168.0.0/24 [110/1] via 192.168.192.1, 00:00:43, Serial0.201
O      192.168.232.0/24 [110/1] via 192.168.192.1, 00:00:47, Serial0.201

```

Мы видим, что Лондон имеет маршрут на адрес 192.168.192.1 через Serial0.201 на сеть 192.168.232.0/24 Сингапура.

Сконфигурируем локальные сети филиалов согласно таблице 1. Проверим полную конфигурацию с помощью расширенной команды **ping**. Например мы можем из маршрутизатора в SanJose проверить есть ли связь от филиального компьютера PCL в Лондоне к филиальному компьютеру PCS в Сингапуре.

Все пинги должны быть успешны. В частности между всеми РС.

Посмотрим путь пакетов от филиального компьютера PCL в Лондоне к филиальному компьютеру PCS в Сингапуре.

PCL#trace 192.168.232.2

```

1 192.168.200.1 0 msec 16 msec 0 msec
2 192.168.192.1 20 msec 16 msec 16 msec
3 192.168.193.4 20 msec 16 msec 16 msec
4 192.168.232.2 20 msec 16 msec *
```

Он лежит через маршрутизатор SanJose (192.168.192.1).

Контрольные вопросы

1. Назовите типы последовательных каналов связи.
2. Где располагается устройство CSU/DSU и какие функции оно выполняет?
3. Что такое DTE и DCE?
4. Как DTE связан с DCE?
5. Сформулируйте идею сетей с коммутацией пакетов СКП.
6. Что такое виртуальные цепи в СКП?
7. Назовите типы СКП?
8. Какие физические каналы связи используют СКП?
9. Как осуществляется доступ потребителя к СКП?
10. Какие технологии передачи данных используются в СКП?
11. В чём различие Frame Relay и X.25?
12. Протокол Frame Relay не исправляет ошибок. Где они тогда исправляются?
13. Что такое АТМ?
14. Назовите протоколы глобальных сетей и для каких типов последовательных каналов связи они используются.
15. Назовите DTE устройства Frame Relay.
16. Назовите DCE устройства Frame Relay.
17. Как связываются во Frame Relay устройства DTE и DCE?
18. Чем отличаются PVC от SVC?
19. Что такое DLCI и как оно соотносится с PVC?
20. Что надо сделать, чтобы передать IP пакет через сеть Frame Relay?
21. Как отобразить DLCI на удалённый IP?
22. Что такое LMI?
23. Опишите работу протокола Инверсный ARP.
24. Как конфигурировать последовательный интерфейс маршрутизатора для подключения к Frame Relay.
25. Когда используется статическое отображение адресов?
26. Как на маршрутизаторе увидеть к каким PVC он подсоединён?
27. Как на маршрутизаторе для интерфейсов увидеть отображения DLCI на удалённый IP?
28. Как на маршрутизаторе для подынтерфейсов увидеть отображения DLCI на удалённый IP?
29. Назовите Топологии Frame Relay.

30. Как осуществляется конфигурирование подынтерфейсов для Frame Relay.
31. Какую роль играет команда `frame-relay interface-dlci` при конфигурировании подынтерфейсов для Frame Relay.
32. Какие ограничения имеет симулятор при работе с Frame Relay?

Порядок выполнения и сдачи работы

1. Изучить теоретическую и практическую часть.
2. Сдать преподавателю теорию работы путём ответа на контрольные вопросы.
3. Выполнить подряд все пункты практической части.
4. Создать все скриншоты, приведенные в практической части.
5. Показать преподавателю результат выполнения пунктов 1, 2 и 3 практической части.
6. Выполните в Boson задание для самостоятельной работы.
7. Показать преподавателю результат выполнения пунктов 1, 2 и 3 задания для самостоятельной работы.
8. Оформите отчёт. Содержание отчёта смотри ниже.
9. Защитите отчёт.

Задание для самостоятельной работы

1. Для полносвязной топологии из четырёх сайтов с локальными сетями (Рисунок 15) спроектируйте Frame Relay сеть с:

- А) динамическим назначением адресов;
- В) подынтерфейсами.

Используйте протокол маршрутизации OSPF.

В вашем распоряжении подсеть `10.v.0.0/16`, где `v` – номер варианта. Адреса сетей и интерфейсов назначьте самостоятельно.

Сделайте скриншоты:

1. 4-х таблиц маршрутизаций для каждого маршрутизатора.
2. Команды расширенного пинга из любого маршрутизатора для двух произвольных пар компьютеров.
3. Результат выполнения команды **`show frame-relay map`** для каждого маршрутизатора

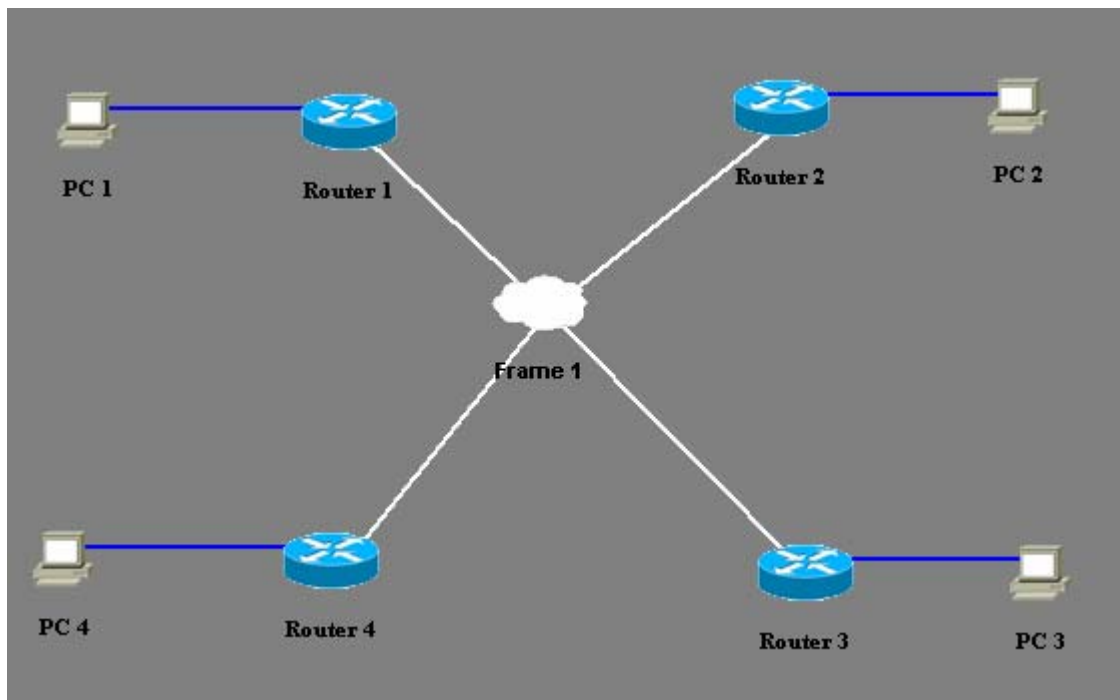


Рис. 15.

2. Неполносвязные топологии

В дизайнере, нажав правой кнопкой на облаке Frame Relay и выбрав пункт Set Frame Relay Parameters, можно отредактировать таблицу DLCI

TO ROUTER					
FROM ROUTER		Router 1	Router 2	Router 3	Router 4
	Router 1		102	103	104
	Router 2	201		203	204
	Router 3	301	302		304
	Router 4	401	402	403	

и получить различные неполносвязные топологии. Рассмотрим четыре

А) Звезда (вилка)

TO ROUTER					
FROM ROUTER		Router 1	Router 2	Router 3	Router 4
	Router 1		102	103	104
	Router 2	201		0	0
	Router 3	301	0		0
	Router 4	401	0	0	

Б) Буква П

TO ROUTER					
FROM ROUTER		Router 1	Router 2	Router 3	Router 4
	Router 1		102	0	0
	Router 2	201		203	0
	Router 3	0	302		304
	Router 4	0	0	403	

В) Буква О (квадрат)

TO ROUTER					
FROM ROUTER		Router 1	Router 2	Router 3	Router 4
	Router 1		102	0	104
	Router 2	201		203	0
	Router 3	0	302		304
	Router 4	401	0	403	

Варианты

Вариант	Топология	Сеть
1	а	1.1.0.0/16
2	б	2.1.0.0/16
3	в	3.1.0.0/16
4	а	4.1.0.0/16
5	б	5.1.0.0/16
6	в	6.1.0.0/16
7	а	7.1.0.0/16
8	б	8.1.0.0/16
9	в	9.1.0.0/16
10	а	10.1.0.0/16
11	б	11.1.0.0/16
12	в	12.1.0.0/16

Адреса сетей и интерфейсов назначьте самостоятельно. Используйте подинтерфейсы. Сделайте скриншоты:

1. 4 таблиц маршрутизаций для каждого маршрутизатора.
2. Команды расширенного пинга из любого маршрутизатора для двух произвольных пар компьютеров.
3. Результат выполнения команды **show frame-relay map** для каждого маршрутизатора

Содержание отчёта.

Отчёт готовится в электронном виде и распечатывается. Отчёт содержит

1. Скриншоты топологий, созданных при выполнении практической части.

2. Все скриншоты, созданные при выполнении практической части.
3. Конфигурации всех из rtr файлов, созданных при выполнении практической части.
4. Конфигурацию маршрутизаторов из rtr файлов, созданных при выполнении заданий для самостоятельной работы.
5. Все скриншоты, указанные в задании для самостоятельной работы.

Лабораторная работа №9. Виртуальные локальные сети VLAN

Теоретическая часть.

Локальные сети в настоящее время принято строить на основании технологии коммутируемого Ethernet. Стремятся минимизировать число используемых концентраторов (хабов-hub) и использовать преимущественно коммутаторы (свичи - switch). В коммутаторе между приёмником и передатчиком на время соединения образуется виртуальный канал (virtual circuit) точка-точка. Такая сеть может быть рассмотрена как совокупность независимых пар приёмник-передатчик, каждая из которых использует всю полосу пропускания. Коммутатор позволяет осуществлять параллельную передачу информации. Коммутация уменьшает вероятность переполнения в сетях Ethernet.

Если коммутатору необходимо передать пакет на какой-то выходной порт, и этот порт занят, то пакет помещается в буферную память. Это позволяет согласовать скорости передатчиков и приёмников пакетов.

Для отправки фрейма через коммутатор используются два метода:

Отправка с промежуточным хранением (store-and-forward). Пакет должен быть принят полностью до того как будет начата его отправка.

Сквозной метод (cut-through). Коммутатор принимает начало пакета, считывает в нём адрес пункта назначения и начинает отправлять пакет ещё до его полного получения

Ethernet-коммутатор узнаёт MAC адреса устройств в сети путём чтения адресов источников в принимаемых пакетах. Коммутатор запоминает в своих внутренних таблицах информацию на какие порты и с каких MAC адресов приходят пакеты. При подключении к одному порту нескольких устройств через концентратор (hub) в таблице одному порту будет соответствовать несколько MAC-адресов. Таблицы хранятся в памяти, адресуемой по смыслу (content-addressable memory, CAM): если адрес отправителя отсутствует в таблице, то он туда заносится. Наряду с парами адрес-порт, в таблице хранится и метка времени. Метка времени в строке таблицы изменяется либо при приходе на коммутатор пакета с таким же адресом, либо при обращении коммутатора по этому адресу. Если строка таблицы не использовалась определённый период времени, то она удаляется. Это позволяет коммутатору поддерживать правильный список адресов устройств для передачи пакетов.

Используя CAM таблицу адресов и содержащийся в пришедшем пакете адрес получателя, коммутатор организует виртуальное соединение порта отправителя с портом получателя и передает пакет через это соединение.

Виртуальное соединение между портами коммутатора сохраняется в течение передачи одного пакета, т.е. для каждого пакета виртуальное соединение организуется заново на основе содержащегося в этом пакете адреса получателя.

Поскольку пакет передается только в тот порт, к которому подключен адресат, остальные устройства подключенные к коммутатору, не получают этот пакет.

В коммутаторах Ethernet передача данных между любыми парами портов происходит независимо и, следовательно, для каждого виртуального соединения выделяется вся полоса канала.

При передаче широковещательного пакета, в коммутаторе образуется «веер» каналов по принципу один ко многим. Примерами источников широковещательного трафика являются ARP и маршрутизирующие протоколы.

Коммутаторы можно соединять друг с другом. При этом группа попарно прямо либо косвенно связанных коммутаторов образует один логический коммутатор с теоретически произвольным числом портов. То есть коммутаторы позволяют создавать теоретически сколь угодно большую локальную сеть. Правильное соединение коммутаторов, то есть выбор топологии сети составляет одну из важнейших задач проектирования локальных сетей.

Рекомендуется осуществлять соединение коммутаторов по слоям (рисунок 1): серверный слой, слой распределения (distribution) и слой доступа (access). Рядовые компьютеры подключаются к слою доступа, а сервера к серверному слою

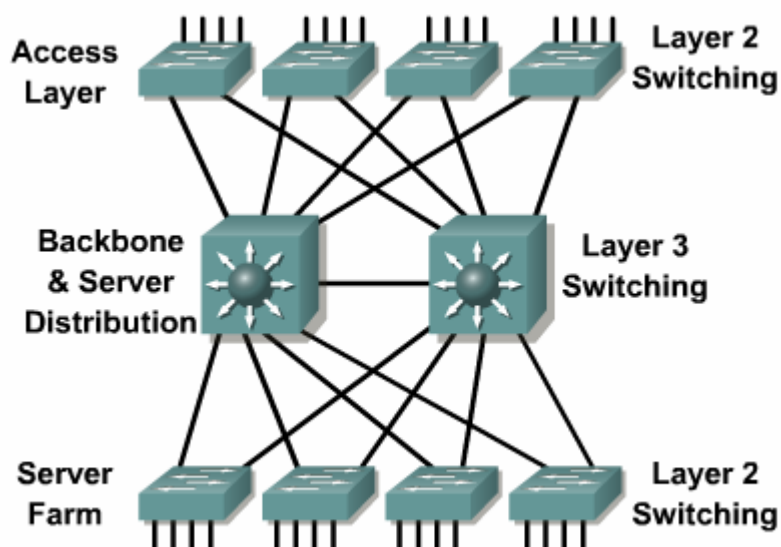


Рис. 1.

Главным препятствием для создания больших локальных сетей с помощью одних только коммутаторов является нелинейный рост объема широковещательного трафика с ростом числа устройств в сети. При числе устройств в сети более, чем 2000 (по другим оценкам 500, по третьим 4000 – всё зависит от топологии сети и класса решаемых задач) объем широковещательного трафика резко возрастает. Добавление новых устройств резко снижает производительность сети.

Например. Если в сети из нескольких тысяч устройств один из компьютеров А впервые осуществляет IP соединение с другим компьютером В в этой сети, то он должен предварительно послать ко всем устройствам сети широковещательный ARP запрос для определения MAC адреса компьютера В.

Локальная сеть, созданная с помощью одних только коммутаторов представляет один домен широковещания. Уменьшить домен широковещания

можно, физически разделив локальную сеть на независимые подсети (независимые группы попарно связанных коммутаторов) и соединить их в единое целое с использованием маршрутизаторов. Такую задачу можно решить только на этапе построения сети, но не в момент её эксплуатации. Здесь на помощь приходят виртуальные локальные сети VLAN (virtual local area network).

Виртуальная локальная сеть VLAN представляет собой совокупность портов одного или более коммутаторов (Рисунок 2).

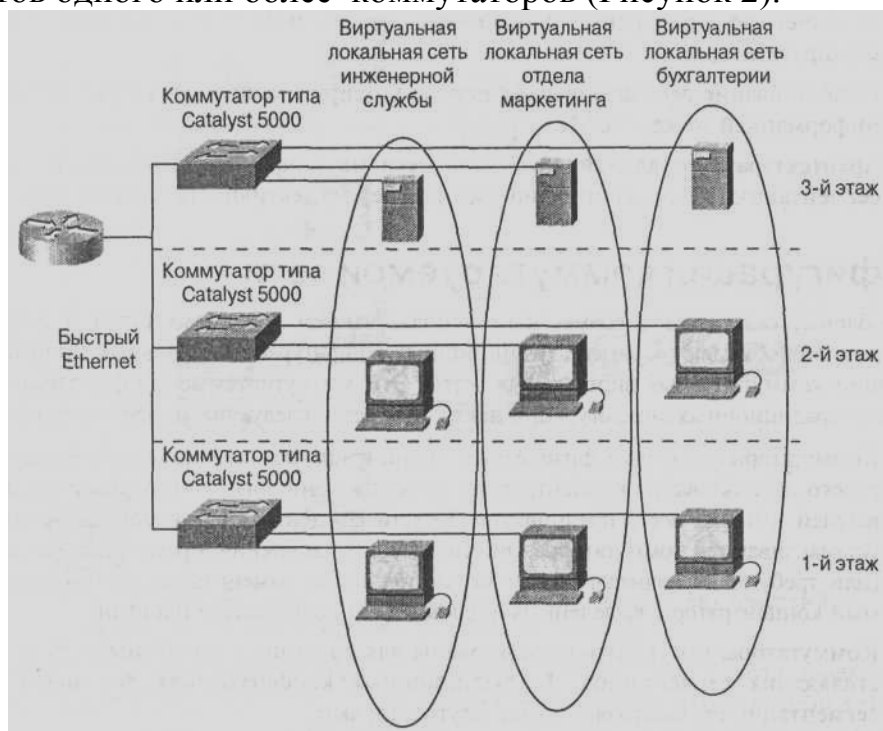


Рис. 2.

VLAN позволяют логически разбить исходную локальную сеть на несколько независимых локальных сетей без физического обрыва сетевых соединений. Для этого администратор сети должен на каждом коммутаторе назначить, какие его порты относятся к каким VLAN. По умолчанию все порты коммутатора относятся к одной VLAN с номером 1. Максимальное число VLAN в коммутаторе равно общему числу его портов. Правильная разбивка локальной сети на VLAN составляет одну из важнейших задач проектирования.

VLAN ведут себя так же, как и физически разделённые локальные сети. То есть после разбивки сети на VLAN мы получим несколько локальных сетей, которые далее необходимо объединить в единое целое с помощью маршрутизации на третьем сетевом уровне.

Концепция VLAN, помимо решения проблемы с широковещательным трафиком даёт также ряд дополнительных преимуществ: формирование локальных сетей не по месту расположения ближайшего коммутатора, а по принадлежности компьютеров к решению той или иной производственной задачи; создание сети по типу потребляемого вычислительного ресурса и требуемой серверной услуги (файл-сервер, сервер баз данных). VLAN

позволяют вести различную политику безопасности для разных виртуальных сетей; переводить компьютер из одной сети в другую без осуществления физического перемещения или переподключения.

Для обмена информацией о VLAN коммутаторы используют магистральный (транковый) протокол. Для осуществления обмена информацией о VLAN между коммутаторами вы должны создать магистральные порты. Магистральный порт это порт, используемый для передачи информации о VLAN в другие сетевые устройства, присоединенные к этому порту. Обычные порты не рекламируют информацию о VLAN, но любой порт может быть настроен для приема/передачи информации о VLAN. Вы должны активизировать магистральный протокол на нужных портах, так как он выключен по умолчанию.

Порт коммутатора работает либо в режиме доступа либо в магистральном режиме. Соответственно связь, подсоединённая к порту является либо связью доступа либо магистральной связью. В режиме доступа порт принадлежит только одной VLAN. Порт доступа присоединяется к оконечному устройству: ПК, рабочей станции, серверу, хабу. Фреймы, проходящие через порт доступа, являются обычными Ethernet фреймами.

Магистральные связи способны поддерживать несколько VLAN. VLAN на различных коммутаторах связываются через магистральный протокол. Магистральные порты не принадлежат определённой VLAN и используются для подсоединения к другим коммутаторам, маршрутизаторам или серверам, имеющим сетевые адаптеры с возможностью для подключения ко многим VLAN.

Магистральные могут расширить VLAN по всей сети. Для магистральных целей назначают высокоскоростные порты коммутаторов: Gigabit Ethernet и 10Gigabit .

Для мультиплексирования трафика VLAN существуют специальные протоколы, позволяющие приёмным портам определить, какому VLAN принадлежит пакет. Для связи между устройствами Cisco используется протокол Inter-Switch Link (ISL). При наличии в сети оборудования нескольких производителей применяется протокол IEEE 802.1Q

Без магистральных связей для поддержки VLAN должно быть организовано по одной связи доступа для каждой VLAN. Такой подход дорог и неэффективен, поэтому магистральные связи абсолютно необходимы при проектировании локальных сетей.

На рисунке 3 порты А и В на коммутаторе Y определены как связи доступа на одной и той же VLAN 200. По определению они могут принадлежать только одной VLAN и не могут получать ethernet фреймы, содержащие идентификатор VLAN. Например, когда Y получает трафик от порта А к порту В, то он не добавляет ISL заголовок в ethernet фреймы.

Порт С на коммутаторе Z также является портом доступа и также принадлежит к VLAN 200. Если порт А пересылает фрейм в порт С, то происходит следующее:

1. Коммутатор Y получает фрейм и, сопоставляя номер порта назначения с номером VLAN, определяет его как трафик, направленный к VLAN 200 на другом коммутаторе,

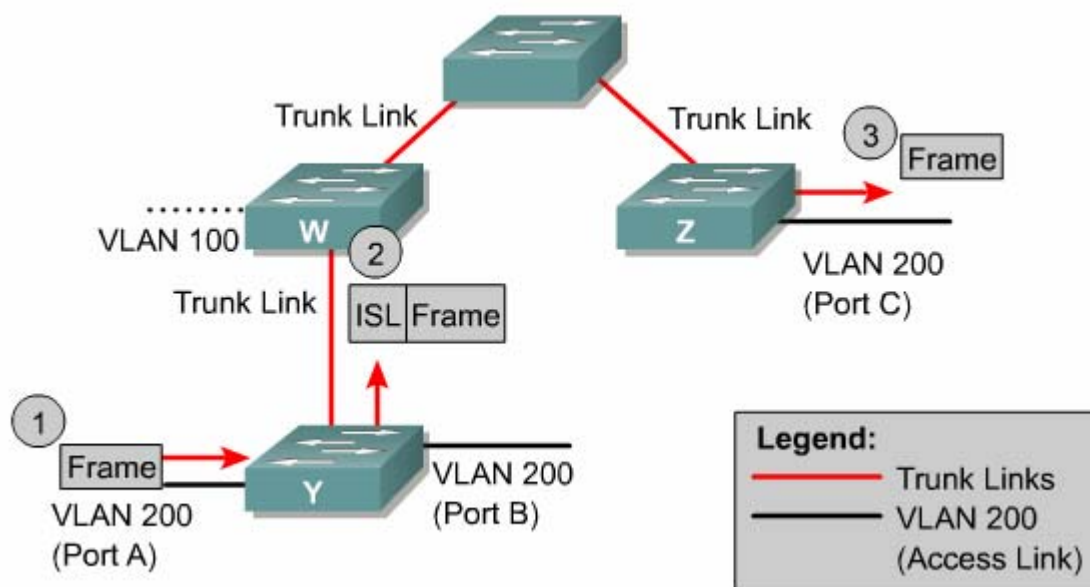


Рис. 3.

2. Коммутатор Y добавляет к фрейму ISL заголовок с номером VLAN и пересылает фрейм через промежуточный коммутатор на магистральную связь.

3. Этот процесс повторяется на каждом коммутаторе по пути фрейма к конечному порту C.

4. Коммутатор Z получает фрейм, удаляет ISL заголовок и направляет фрейм на порт C.

Если порт находится в магистральном режиме, то он может быть настроен или для транспорта всех VLAN или ограниченного множества VLAN. Магистральные связи используются для связи коммутаторов с другими коммутаторами, маршрутизаторами или с серверами, имеющими поддержку VLAN.

Согласно базовой терминологии магистраль это связь точка-точка, поддерживающая несколько VLAN. Целью магистрали является сохранение номеров портов при создании связи между двумя устройствами, образующими VLAN.

Верхняя фигура на рисунке 4 показывает способ создания VLAN путём использования двух физических связей между коммутаторами (по одной на каждую VLAN). Это решение плохо масштабируется: при добавлении третьего VLAN надо пожертвовать ещё двумя портами. Это решение неэффективно и в смысле разделения нагрузки: малый трафик на некоторых связях может не стоить того, что эта связявляется пучком виртуальных связей через одну физическую связь. На нижней фигуре одна физическая связь способна нести трафик для любой VLAN. Для достижения этого коммутатор Sa так оформляет

фреймы, что Sb знает, на какую VLAN они направляется. Для такого оформления пакетов используются либо стандарт IEEE 802.1Q либо Cisco протокол ISL (Inter-Switch Link).

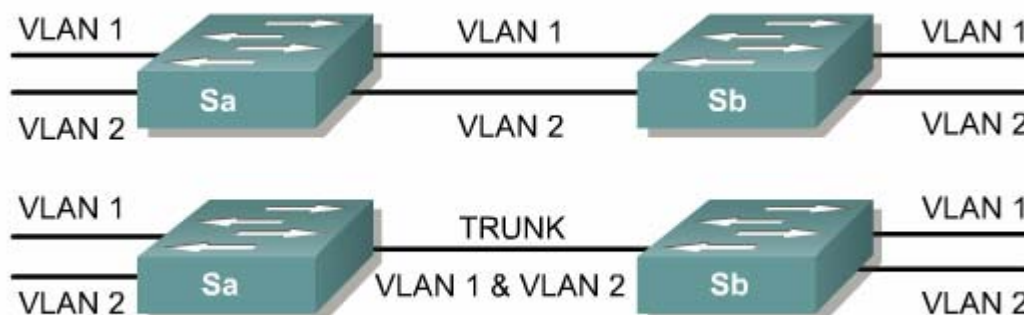


Рис. 4.

Для больших сетей ручная конфигурация VLAN становится весьма трудоёмкой задачей. Cisco VLAN Trunk Protocol (VTP) служит для автоматического обмена информацией о VLAN через магистральные порты. Преимуществом использования VTP является то, что вы можете контролировать добавление, удаление или изменение сетей VLAN из коммутаторов на котором созданы VTP сервера. После настройки ваших коммутаторов как VTP серверов, остальные коммутаторы вашей сети могут быть настроены как клиенты, которые только получают VLAN информацию. Недостатком является ненужный трафик, создаваемый на магистральный портах для устройств, которым возможно не нужна эта информация.

Если ваша сеть будет содержать много коммутаторов, содержащих много виртуальных сетей, расположенных в разных коммутаторах, возможно, имеет смысл использовать VTP. Если ваша сеть останется достаточно статической, и VLAN не будут добавляться или изменяться по отношению к начальной конфигурации, то лучше использовать статическое определение виртуальных сетей.

В топологии локальных сетей возможны циклы (петли). Например, уже три коммутатора соединённых друг с другом по кругу образуют цикл в топологии. Петли приводят к неоднозначности при определении пути от источника пакетов к приёмнику. Для решения этой серьёзной проблемы был разработан протокол связующего дерева STP (spanning tree protocol). Для графа топологии каждой VLAN, которая определена в сети, строится минимальное покрывающее дерево (граф без циклов) с вершиной в некотором коммутаторе. Для физической реализации таких деревьев STP переводит избыточные порты в состояние блокировки. Расчёт деревьев производится параллельно на всех коммутаторах. Далее пакеты во VLAN идут только по путям, определённым в построенных

покрывающих деревьях. При изменении топологии, активации/остановке портов происходит пересчёт покрывающих деревьев.

Для создания топологии связующего дерева существуют специальные фреймы, называемые модулями данных мостового протокола (bridge protocol data units, BPDU). Эти фреймы отправляются и принимаются всеми коммутаторами в сети через равные промежутки времени.

Конфигурирование статических VLAN

1. Статические VLAN это совокупность портов на коммутаторе, которые вручную назначаются командой IOS при конфигурировании интерфейса.

На коммутаторах Cisco серии 1950 пустую VLAN создают командой

```
Switch(config)#vlan #VLAN name <NAME>,
```

где #VLAN –целое число –номер VLAN, <NAME> - последовательность символов – имя VLAN.

Например

```
Switch(config)#vlan 77 name 77
```

Порт во VLAN добавляется в режиме конфигурации интерфейса этого порта, например, команды

```
Switch(config)#interface Ethernet0/3
```

```
Switch(config-if)# vlan-membership static 77
```

Добавляют во VLAN 77 порт 3. Если VLAN 77 до этого не была создана, то она создаётся с именем равным его номеру.

После создания всех VLAN, можно проверить, какие порты принадлежат каким VLAN командой **show vlan-membership**.

Детальную информацию о VLAN с номером #VLAN можно получить командой **Show vlan #VLAN**, например

```
Switch#show vlan 77
```

Для того, чтобы перевести порт в магистральный режим необходимо в режиме конфигурации интерфейса этого порта выполнить команду

```
Switch(config-if)# trunk on
```

Вывести интерфейс из магистрального режима можно командой

```
Switch(config-if)# trunk off
```

Наиболее простым способом перевода всего ethernet соединения точка-точка в магистральный режим является использование команды **trunk on** на обоих интерфейсах на концах соединения.

Для проверки состояния магистралей используется команда

```
Switch# show trunk (a|b).
```

2. Для создания пустой VLAN с номером #VLAN на коммутаторах Cisco серии 2950 используются команды

```
Switch#vlan database
```

```
Switch(vlan)#vlan #VLAN
```

```
SwitchVlan)# exit
```

Например, команды

```
Switch#vlan database
```

```
Switch(vlan)#vlan 33
```

```
SwitchVlan)# exit
```

создадут пустую VLAN с номером 33 и система даст VLAN имя VLAN0033.

Заметим, что команды выполняются не в режиме конфигурации.

Команда **switchport mode** используется для установки интерфейса в динамический режим, режимы доступа или режим магистральной (trunk).

```
Switch(config-if)#switchport mode [access | dynamic | trunk]
```

Хотя режим доступа является режимом по умолчанию, но в ряде случаев устройство, присоединённое к порту коммутатора, может перевести его в магистральный режим. Поэтому рекомендуется все немагистральные порты переводить в режим доступа командой **switchport mode access**.

Для статического помещения текущего интерфейса во VLAN используются команды

```
Switch(config-if)#switchport mode access
```

```
Switch(config-if)#switchport access vlan #number
```

где **#number** – число – номер VLAN.

Команда **interface range** определяет диапазон интерфейсов для последующих конфигураций. Например, порты с первого по шестой могут быть помещены во VLAN 10 командами

```
Switch(config)#interface range fa0/1 – 6
```

```
Switch(config-if-range)#switchport access vlan 10
```

После настройки VLAN проверьте настройку командами **show running-config**, **show vlan** и **show vlan brief**.

При настройке VLAN помните, что по умолчанию все порты находятся во VLAN 1.

Для создания или конфигурирования магистральной VLAN вы должны настроить порт как магистральный

```
Switch(config-if)#switchport mode trunk
```

По умолчанию последняя команда определяет порт как магистральный для всех VLAN в сети. Однако существуют ситуации, когда магистраль не должна поддерживать все VLAN. Типичной является ситуация с подавлением широковещания. Широковещание посылается на каждый порт во VLAN. Магистральная связь выступает как член VLAN и должна пропускать всё широковещание. Если на другом конце магистральной нет портов нужной VLAN, то полоса пропускания и процессорное время устройств тратится попусту.

Если VLAN не используется на другом конце магистральной, нет нужды разрешать эту VLAN на этой магистральной. По умолчанию магистральные порты принимают и передают трафик со всех VLAN в сети. Для сокращения магистрального трафика используйте команду

```
Switch(config-if)#switchport trunk allowed vlan vlan-list
```

Например, команда

```
Switch(config-if)#switchport trunk allowed vlan 3
```

```
Switch(config-if)#switchport trunk allowed vlan 6-10
```

разрешает на магистральной VLAN 3 и затем VLAN с 6 по 10. О том, какие VLAN разрешены на магистральной можно посмотреть командой **show running-config**.

Для удаления большого числа VLANs из магистральной проще вначале удалить все VLAN, а затем выборочно разрешать.

Простейший способ перевести связь в режим доступа это задать на интерфейсах с её двух сторон по команде

SwitchA(config-if)#**switchport mode access**

Простейший способ перевести связь в режим доступа это задать на интерфейсах с её двух сторон по команде

SwitchA(config-if)#**switchport mode trunk**

Практическая часть

1. Соберите топологию на коммутаторах серии 1950 (Рисунок 5). Коммутаторы соедините двумя Fastethernet (fa0/26 и fa0/27) соединениями. Компьютеры подсоедините к интерфейсам согласно таблице 1. Назначьте компьютерам адреса, согласно таблице 1. Все компьютеры входят в одну подсеть 172.16.0.0 255.255.0.0. Маршруты по умолчанию на компьютерах не устанавливайте. Пропингуйте сеть.

Организуем в нашей сети два VLAN. Компьютеры 20_1 и 20_2 поместим во VLAN с номером 20, а компьютеры 30_1 и 30_2 поместим во VLAN с номером 30.

```
1912-1(conf)#interface Ethernet0/2
1912-1(conf-if)#vlan-membership static 20
1912-1(conf-if)#interface Ethernet0/3
1912-1(conf-if)#vlan-membership static 30
1912-2(conf)#interface Ethernet0/2
1912-2(conf-if)#vlan-membership static 20
1912-2(conf-if)#interface Ethernet0/3
1912-2(conf-if)#vlan-membership static 30
```

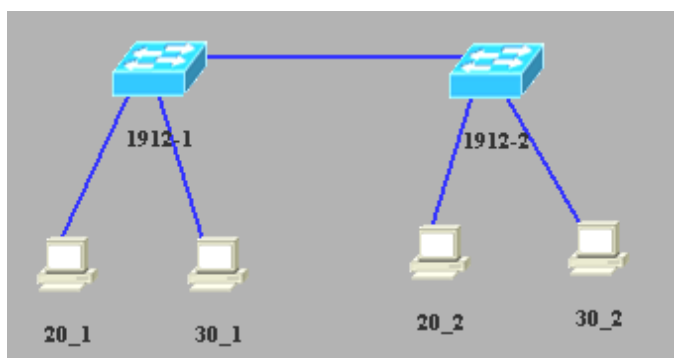


Рис. 5.

	Fa/26	Fa/26	
	Fa/27	Fa/27	

Компьютер	1912-1	1912-2	Адрес
20_1	E0/2		172.16.20.1/16
30_1	E0/3		172.16.20.2/16
20_2		E0/2	172.16.30.1/16
30_2		E0/3	172.16.30.2/16

Т
аблица 1.

На обоих коммутаторах проверьте результаты создания VLAN, например
1912-1#sh vl 20

VLAN Name		Status		Ports					

20	VLAN0020	Enabled		2					

VLAN	Type	SAID	MTU	Parent	RingNo	BridgeNo	Stp	Trans1	Trans2

20	Ethernet	100020	1500	0	1	1	Unkn	0	0

И

1912-1#sh vlan-membership

Port	VLAN	Membership Type

1	1	Static
2	20	Static
3	30	Static
4	1	Static
5	1	Static
6	1	Static
7	1	Static
8	1	Static
9	1	Static
10	1	Static
11	1	Static
12	1	Static
AUI	1	Static
A	1	Static
B	1	Static

Так как нет обмена информации о VLAN между коммутаторами, то компьютеры будут пинговать только самих себя.

Организуем магистрали на коммутаторах. Для этого используем сразу по два Fastethernet порта

```
1912-1(conf)#interface FastEthernet0/26
1912-1(conf-if)#trunk On
1912-1(conf-if)# FastEthernet0/27
1912-1(conf-if)#trunk On
1912-2(conf)#interface FastEthernet0/26
1912-2(conf-if)#trunk On
1912-2(conf-if)# FastEthernet0/27
1912-2(conf-if)#trunk On
```

Проверим состояния магистралей командами **show trunk a** и **show trunk b**. Теперь компьютеры в пределах одной VLAN должны пинговаться, а компьютеры, находящиеся в различных VLAN не должны пинговаться (см. таблицу 2).

Ping из/в	20_1	30_1	20_2	30_2
20_1	Да	Нет	Да	Нет

30_1	Нет	Да	Нет	Да
20_2	Да	Нет	Да	Нет
30_2	Нет	Да	Нет	Да

Таблица 2.

Сохраните конфигурацию. Определитесь с тем какой .rtr файл за какое устройство отвечает.

2. Загрузим дизайнер и изменим топологию, сохранив её в новом файле. В этой топологии маршрутизатор модели 2612 соединён двумя связями с интерфейсами e0/12 коммутаторов (рисунок 6). Загрузим топологию в симулятор. Загрузим поочерёдно в каждое устройство сохранённую конфигурацию. Снова проверьте, что компьютеры в пределах одной VLAN пингуются, а компьютеры, находящиеся в различных VLAN не пингуются.

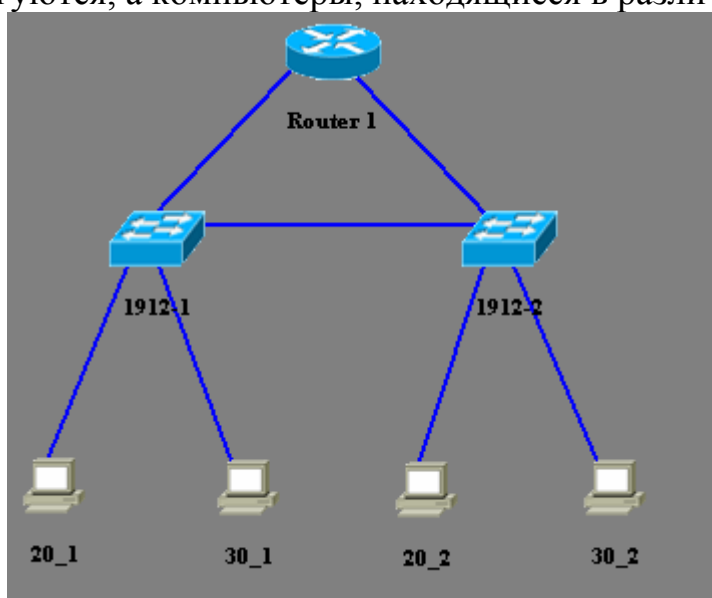


Рис. 6.

Поставим задачу объединения виртуальных сетей с помощью маршрутизатора. Для этого следует разбить нашу сеть 172.16.0.0/16 на две подсети 172.16.20.0/24 и 172.16.30.1/24. Для этого просто поменяем маски у компьютеров на 255.255.255.0.

Теперь компьютеры пингуются в пределах одной VLAN и в пределах одной IP подсети, а это значит только сами на себя.

Введём на коммутаторах интерфейсы, подсоединённые к маршрутизатору в виртуальные сети

```
1912-1(conf)#interface Ethernet0/12
1912-1(conf-if)#vlan-membership static 20
1912-2(conf)#interface Ethernet0/12
1912-2(conf-if)#vlan-membership static 30
```

Настроим IP адреса на маршрутизаторе

```
Router(conf)#interface Ethernet0
Router(conf-if)#ip address 172.16.20.254 255.255.255.0
Router(conf-if)#no shutdown
```

```
Router(conf-if)#interface Ethernet1
Router(conf-if)#ip address 172.16.30.254 255.255.255.0
Router(conf-if)#no shutdown
```

Теперь маршрутизатор маршрутизирует наши две сети 172.16.20.0/24 и 172.16.30.1/24. Добавим на наших компьютерах маршрутизацию по умолчанию на интерфейсы маршрутизатора

```
20_1#ipconfig /dg 172.16.20.254
20_2#ipconfig /dg 172.16.20.254
30_1#ipconfig /dg 172.16.30.254
30_2#ipconfig /dg 172.16.30.254
```

Теперь из всех устройств нашей сети мы можем пинговать все наши IP адреса.

3. Покажем, как с использованием транзитных линий, мы можем сэкономить порты.

Изменим топологию, перебросив магистраль fa0/26 от коммутаторов к интерфейсу fa0/0 маршрутизатора (рисунок 7). Загрузим топологию в симулятор. Загрузим по очереди в каждое устройство, кроме маршрутизатора, сохранённые конфигурации. Заметим, что в конфигурации коммутатора 1912-2 установка магистрали на fa0/26 не нужна, хотя она и не мешает.

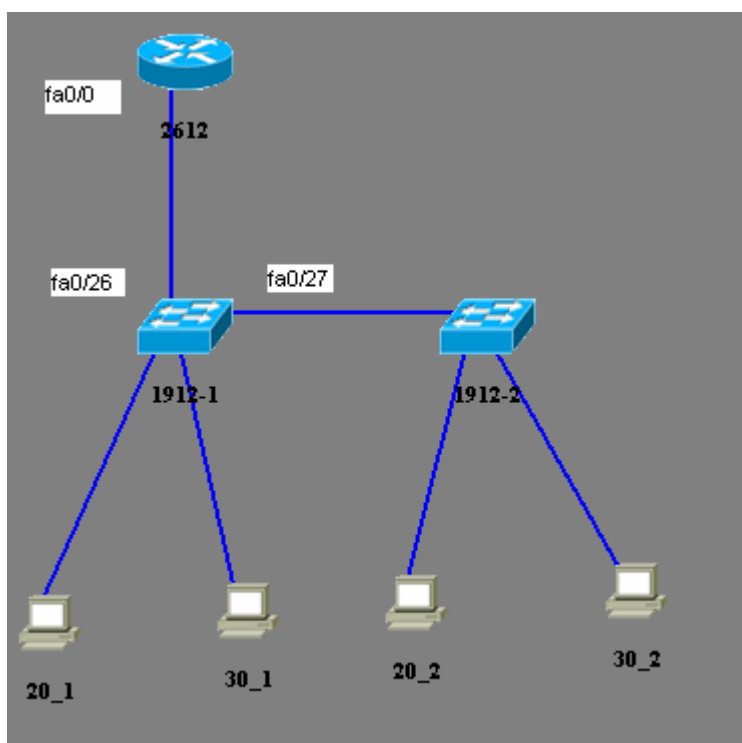


Рис. 7.

На маршрутизаторе разобьём интерфейс fa0/0 на два подинтерфейса fa0/0.20 и fa0/0.30. Определим на них инкапсуляцию isl и поместим их в виртуальные сети 20 и 30, соответственно.

```
Router(conf)#interface FastEthernet0/0.20
Router(conf-subif)#encapsulation isl 20
Router(conf-subif)#ip address 172.16.20.254 255.255.255.0
```

```
Router(conf-subif)#interface FastEthernet0/0.30
Router(conf-subif)#encapsulation isl 30
Router(conf-subif)#ip address 172.16.30.254 255.255.255.0
```

Посмотрим таблицу маршрутов

```
Router#show ip route
```

```
c      172.16.20.0/24 is directly connected, 172.16.20.254
c      172.16.30.0/24 is directly connected, 172.16.30.254
```

Проверьте, что из каждого устройства вы можете пинговать все адреса в сети.

Выполните на Router команду расширенного пинга от адреса 172.16.20.1 компьютера 20_1 из VLAN 20, подключённого к коммутатору 1912_1 к адресу 172.16.30.2 компьютера 30_2 из VLAN 30, подключённого к коммутатору 1912_2

```
Router#ping
Protocol [ip]:
Target IP address: 172.16.30.2
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]:y
Source address or interface:172.16.20.1
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]:

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.30.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

Контрольные вопросы

1. Почему предпочитают строить локальные сети с помощью коммутаторов, а не концентраторов?
2. Какие существуют методы для отправки фрейма через коммутатор?
3. Как коммутатор узнаёт MAC адреса подключенных устройств?
4. Где и как в коммутаторе хранятся адреса подключенных устройств?
5. Что такое виртуальное соединение и как долго оно существует?
6. Сколь большую локальную сеть можно создать с помощью коммутаторов?
7. Как принято строить большие локальные сети?
8. Что является главным препятствием для создания больших локальных сетей с помощью одних только коммутаторов?
9. Что такое домен широковещания?
10. Как уменьшить домен широковещания?
11. Что такое VLAN?
12. Какие проблемы локальных сетей решает VLAN?
13. Как в локальной сети организовать обмен информацией о VLAN?

14. В каких режимах работают порты коммутатора?
15. Какой VLAN принадлежит магистральный порт?
16. Как распространить одну VLAN на несколько коммутаторов?
17. Можно ли организовать несколько VLAN на нескольких коммутаторах без использования магистралей?
18. Зачем нужны протоколы ISL и IEEE 802.1Q?
19. Зачем нужны магистрали в локальной сети?
20. Какие задачи решает VTP?
21. Какие задачи решает STP?
22. Какими командами можно организовать VLAN?
23. Какой командой перевести порт в режим доступа и в режим магистрали?
24. Какой командой можно получить информацию о VLAN?
25. В локальной сети имеется одиннадцать VLAN. Сколько маршрутизаторов надо для объединения всех 11 VLAN в единое целое?

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. Выполните в Boson задание для самостоятельной работы.
5. Для своего варианта предъявите преподавателю возможность выполнить расширенный пинг между любыми адресами для топологий на рисунках 6 и 7.
6. Оформите отчёт. Содержание отчёта смотри ниже.
7. Защитите отчёт.

Задание для самостоятельной работы

Повторить все пункты практической части для IP сети предприятия, согласно вариантам. Практическая часть проделана для сети 172.16.0.0/16.

Вар.	Сеть	Вар.	Сеть	Вар.	Сеть
1	2.1.0.0/16	5	6.1.0.0/16	9	10.1.0.0/16
2	3.1.0.0/16	6	7.1.0.0/16	10	11.1.0.0/16
3	4.1.0.0/16	7	8.1.0.0/16	11	12.1.0.0/16
4	5.1.0.0/16	8	9.1.0.0/16	12	13.1.0.0/16

Создать те же скриншоты, что и при выполнении практической части.

Содержание отчёта.

Отчёт готовится в электронном виде и распечатывается. Отчёт содержит

1. Скриншоты топологий, созданных при выполнении практической части.
2. Конфигурации всех маршрутизаторов и компьютеров из rtr файлов, созданных при выполнении практической части.
3. Конфигурации всех маршрутизаторов и компьютеров из rtr файлов, созданных при выполнении задания для самостоятельной работы.
4. Все скриншоты, указанные в задании для самостоятельной работы

Литература

1. Документация к программе Boson Netsim.
2. Cisco Network Academy.

Содержание

Лабораторная работа №1. Знакомство со средой	3
Лабораторная работа №2. Введение в межсетевую операционную систему IOS компании Cisco	13
Лабораторная работа №3. Статическая маршрутизация	31
Лабораторная работа №4. Динамическая маршрутизация	41
Лабораторная работа №5. Бесклассовая адресация CIDR и маски переменной длины VLSM	57
Лабораторная работа №6. Списки управления доступом ACL	70
Лабораторная работа №7. Преобразование сетевых адресов NAT	87
Лабораторная работа №8. Удалённый доступ. Frame Relay	98
Лабораторная работа №9. Виртуальные локальные сети VLAN	130