

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по выполнению лабораторных работ
по дисциплине «Основы инфокоммуникационных систем и сетей»
для студентов направления подготовки
11.03.02 Инфокоммуникационные технологии и системы связи
Направленность (профиль)
«Интеллектуальная обработка данных в инфокоммуникационных
системах и сетях»

СОДЕРЖАНИЕ

1. Создание прямых и перекрещенных кабелей «неэкранированная витая пара».	
2. Изучение программного симулятора работы сети Cisco Packet Tracer	
3. Определение параметров сетевого адаптера и настроек протокола IP	
4. Планирование подсетей.....	
5. Моделирование сетей с использованием топологических структур общая шина и звезда	
6. Исследование коммуникационного оборудования Switch 2960	
7. Списки управления доступом ACL	
8. Преобразование сетевых адресов NAT	
9. Виртуальные локальные сети VLAN	
10. Статическая маршрутизация	
11. Динамическая маршрутизация	
12. Исследование протокола PPP	
13. Исследование технологии Frame Relay	
14. Исследование технологии ATM	
15. Сравнение коммутаторов серий 2960 и 3560.	
16. Проектирование локальной сети. Отработка комплексных практических навыков.	
17. Настройка сетей VLAN, VTP и DTP.	
18. Поиск и устранение неполадок с VTP и DTP.....	
19. Исследование проектирования с резервированием.	
20. Настройка PVST+.	
21. Настройка Rapid PVST+.	
22. Настройка EtherChannel.	
23. Поиск и устранение неполадок в работе EtherChannel.	
24. Отладка последовательных интерфейсов.	
25. Концепции WAN. Отработка комплексных практических навыков	
26. Настройка GRE.	
27. Отладка GRE	
28. Настройка расширенных списков контроля доступа.	
29. Поиск и устранение неполадок. Документирование сети	
30. Изучение сети ЭВМ с использованием системных утилит	

Лабораторная работа №1.

Создание прямых и перекрещенных кабелей «неэкранированная витая пара»

Учебные вопросы:

1. Создание и тестирование прямого соединительного кабеля Ethernet.
2. Создание и тестирование перекрещенного кабеля Ethernet.
3. Тестирование и определение характера неисправности кабелей Ethernet.

Цели

Создать и протестировать прямой и перекрещенный неэкранированный кабель «неэкранированная витая пара» для сети Ethernet.

Общие положения

На этом лабораторном занятии требуется сделать прямые соединительные кабели и перекрещенные кабели для Ethernet. В прямом кабеле цвет провода контакта 1 на одном конце совпадает с цветом провода контакта 1 на другом конце кабеля. Для остальных семи контактов ситуация аналогична.

Кабель будет создан на основе стандарта TIA/EIA T568A или T568B для Ethernet, который определяет цвета проводов для каждого контакта. Прямые соединительные кабели обычно используются для прямого подключения узла к концентратору или компьютеру, либо к настенному креплению в офисе.

В перекрещенном кабеле вторая и третья пары на разъеме RJ-45 на одном конце кабеля перекрещены по отношению к другому концу. Выводы данного кабеля соответствуют стандарту T568A на одном конце и стандарту T568B на другом. Перекрещенные кабели обычно используются для подключения к концентраторам и коммутаторам или прямого соединения двух узлов для создания простой сети.

Потребуются следующие ресурсы: два куска кабеля длиной 0,6 – 0,9 м (2 – 3 фута) категории 5 или 5e; не менее четырех разъемов RJ-45 (может потребоваться больше при неправильном соединении проводов); обжимные клещи RJ-45; тестер кабелей Ethernet; кусачки.

Таблица 1. Стандарт T568A

Стандарт T568A			
№ контакта	№ пары	Цвет провода	Функция
1	2	Белый/зеленый	Передача
2	2	Зеленый	Передача
3	3	Белый/оранжевый	Прием
4	1	Синий	Не используется
5	1	Белый/синий	Не используется
6	3	Оранжевый	Прием
7	4	Белый/коричневый	Не используется
8	4	Коричневый	Не используется

Таблица 2. Стандарт T568B

Стандарт T568B			
№ контакта	№ пары	Цвет провода	Функция
1	2	Белый/оранжевый	Передача
2	2	Оранжевый	Передача
3	3	Белый/зеленый	Прием
4	1	Синий	Не используется
5	1	Белый/синий	Не используется
6	3	Зеленый	Прием
7	4	Белый/коричневый	Не используется
8	4	Коричневый	Не используется

1. Создание и тестирование прямого соединительного кабеля Ethernet

Шаг 1: получение и подготовка кабеля

а. Определите требуемую длину кабеля. Это может быть расстояние от устройства, такого как компьютер, до устройства, к которому он подключается (например, концентратор или коммутатор), или между устройством и штепсельным разъемом RJ-45. Добавьте к этому расстоянию не менее 30,48 см (1 фут). Согласно стандарту TIA/EIA максимальная длина кабеля равна 5 м (16,4 фута). Стандартные длины кабелей Ethernet: 6 м (20 футов), 1,83 м (6 футов)

или 3,05 м (10 футов).

б. Отрежьте кусок кабеля требуемой длины. Для соединительных кабелей (кабели между конечными сетевыми устройствами, такими как ПК, и разъемом RJ-45) обычно используется скрученный кабель «неэкранированная витая пара», так как такие кабели дольше служат при многократном сгибании. Они называются скрученными, так как каждый из проводов в кабеле сделан из нескольких жил из чистой меди, а не из одного одножильного провода. Одножильный провод используется для трасс кабелей между разъемом RJ-45 и монтажным блоком.

в. С помощью инструмента для снятия изоляции очистите от оболочки оба конца кабеля на 5,08 см.

Шаг 2: подготовка и вставка проводов

а. Определите, какой стандарт будет использоваться.

[T568A | T568B]

б. Определите правильную таблицу на основе используемого стандарта соединений проводов.

в. Разверните пары кабелей и упорядочите их примерно в требуемом порядке на основе выбранного стандарта.

г. Расплетите короткий участок пар и упорядочите их в порядке, точно соответствующем стандарту.

Очень важно расплести как можно меньший участок. Скручивание очень важно, так как оно обеспечивает подавление помех.

д. Выпрямите и разгладьте провода между своими большим и указательным пальцами.

е. Убедитесь, что порядок проводов кабеля все еще соответствует стандарту.

ж. Обрежьте прямой участок кабеля на расстоянии 1,25 – 1,9 см (1/2 – 3/4") от края кабельной оболочки. Если оставить более длинный участок, кабель будет восприимчив к перекрестным помехам (помехам, создаваемым соседними проводами).

з. При вставке проводов выступ (штырь, торчащий из разъема RJ-45) должен находиться с обратной стороны и быть направленным вниз. Вставьте провода в разъем RJ-45 до упора, все провода должны заходить в разъем на максимальную длину.

Шаг 3: осмотр, сгибание и повторный осмотр кабеля

а. Осмотрите кабель и проверьте правильность цветового кода при подключении проводов к правильным номерам контактов.

б. Осмотрите конец разъема. Восемь проводов должны быть плотно сжаты на конце разъема RJ-45. Некоторая часть оболочки кабеля должна заходить в разъем. Это ослабляет напряжение кабеля. Если оболочка заходит в разъем недостаточно глубоко, то в конечном итоге это может привести к повреждению кабеля.

в. Если все правильно выровнено и вставлено, вставьте разъем RJ-45 и кабель в обжимные клещи. Обжимные клещи обожмут разъем RJ-45 двумя поршнями.

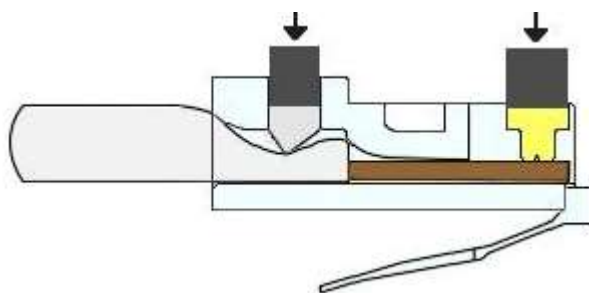


Рисунок 1. Разъем RJ-45 и кабель в обжимных клещах

г. Повторно осмотрите разъем. В случае неправильной установки обрежьте конец и повторите процесс.

Шаг 4: обработка другого конца кабеля

а. Выполните ранее описанные действия, чтобы установить разъем RJ-45 на другом конце кабеля.

б. Повторно осмотрите разъем. В случае неправильной установки обрежьте конец и повторите процесс.

Шаг 5: тестирование кабеля

а. С помощью кабельного тестера проверьте работоспособность прямого кабеля. Если кабель не прошел тест, заново выполните данную лабораторную работу.

2. Создание и тестирование перекрещенного кабеля Ethernet

Шаг 1: получение и подготовка кабеля

а. Определите требуемую длину кабеля. Это может быть расстояние между концентраторами, от концентратора до коммутатора, между коммутаторами, от компьютера до маршрутизатора или между компьютерами. Добавьте к этому расстоянию не менее 30,48 см (1 фут).

б. Отрежьте кусок кабеля требуемой длины и с помощью инструмента для снятия изоляции очистите от оболочки оба конца кабеля на 5,08 см.

Шаг 2: подготовка и вставка проводов в соответствии со стандартом T568A

а. Обратитесь к таблице T568A.

б. Разверните пары кабелей и упорядочите их примерно в требуемом порядке на основе стандарта T568A.

в. Расплетите короткий участок пар и упорядочите их в порядке, точно соответствующем стандарту. Очень важно расплести как можно меньший участок. Скручивание очень важно, так как оно обеспечивает подавление помех.

г. Выпрямите и разгладьте провода между своими большим и указательным пальцами.

д. Убедитесь, что порядок проводов кабеля соответствует стандарту.

е. Обрежьте прямой участок кабеля на расстоянии 1,25 – 1,9 см от края кабельной оболочки. Если оставить более длинный участок, кабель будет восприимчив к перекрестным помехам (помехам, создаваемым соседними проводами).

ж. При вставке проводов выступ (штырь, торчащий из разъема RJ-45) должен находиться с обратной стороны и быть направленным вниз. Вставьте провода в разъем RJ-45 до упора, все провода должны заходить в разъем на максимальную длину.

Шаг 3: осмотр, сгибание и повторный осмотр кабеля

а. Осмотрите кабель и проверьте правильность цветового кода при подключении проводов к правильным номерам контактов.

б. Осмотрите конец разъема. Восемь проводов должны быть плотно сжаты на конце разъема RJ-45. Некоторая часть оболочки кабеля должна заходить в разъем. Это ослабляет напряжение кабеля, которое в конечном итоге может привести к повреждению кабеля.

в. Если все правильно выровнено и вставлено, вставьте разъем RJ-45 и кабель в обжимные клещи. Обжимные клещи обожмут разъем RJ-45 двумя поршнями.

Шаг 4: обработка другого конца кабеля в соответствии со стандартом T568B

а. Выполните ранее описанные действия (но используя таблицу и стандарт T568B), чтобы установить разъем RJ-45 на другом конце кабеля.

б. Повторно осмотрите разъем. В случае неправильной установки обрежьте конец и повторите процесс.

Шаг 5: тестирование кабеля

а. С помощью кабельного тестера проверьте работоспособность перекрещенного кабеля. Если кабель не прошел тест, заново выполните данную лабораторную работу.

3. Тестирование и определение характера неисправности кабелей Ethernet.

Определить характер неисправности предлагаемого для исследования кабелей. Описать характер неисправностей.

Вопросы:

1. Какой стандарт [T568A | T568B] лучше использовать дома, если есть домашняя сеть или требуется ее создать?
2. Какая часть процесса создания таких кабелей оказалась наиболее трудной? Сравните свое мнение с мнением сокурсников.
3. Какая длина кабеля выбрана и почему?
4. Все ли четыре пары кабелей скручены одинаково? Обсудите причины в каждом случае.
5. Многие технические специалисты включают перекрещенный кабель в свой инструментарий. Как вы думаете, когда следует использовать перекрещенный кабель и в каких ситуациях такой кабель используется сетевым техником?

Лабораторная работа № 2

Изучение программного симулятора работы сети Cisco Packet Tracer

Учебные вопросы:

1. Изучение возможностей Packet Tracer.
2. Работа с файлами в симуляторе.

Теоретическая часть

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

Для запуска Cisco Packet Tracer необходимо вызвать исполняемый файл, PacketTracer5.exe. Общий вид программы представлен на рис. 1.1.

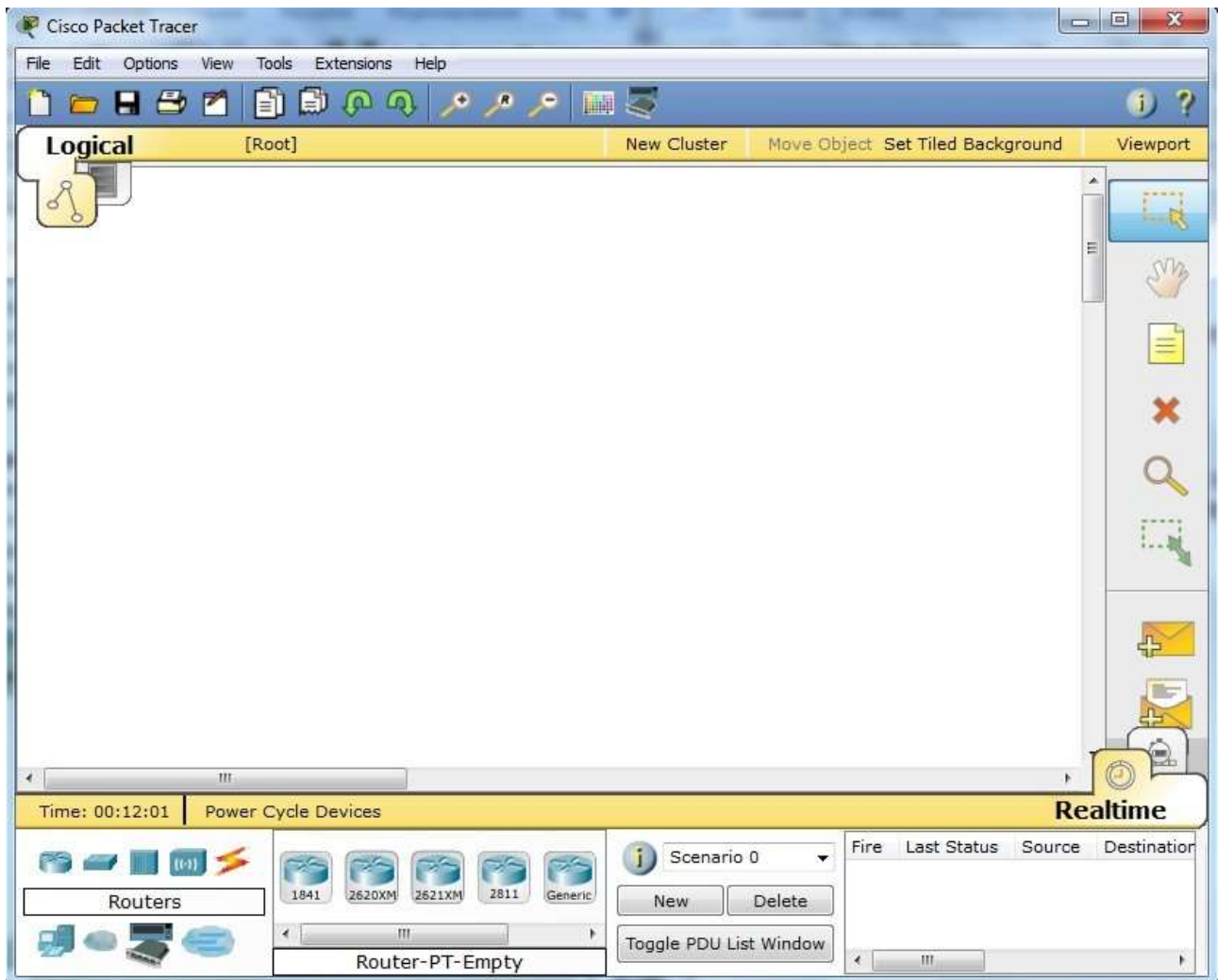


Рисунок 1.1. Общий вид программы Packet Tracer

Рабочая область окна программы состоит из следующих элементов:

- 1 **Menu Bar** – панель, которая содержит меню File, Edit, Options, View, Tools, Extensions, Help.
- 2 **Main Tool Bar** – содержит графические изображения ярлыков для доступа к командам меню File, Edit, View и Tools, а также кнопку Network Information.
- 3 **Common Tools Bar** – панель, обеспечивающая доступ к наиболее используемым инструментам программы: Select, Move Layout, Place Note, Delete, Inspect, Add Simple PDU и Add Complex PDU.
- 4 **Logical/Physical Workspace and Navigation Bar** – панель, которая дает возможность переключать рабочую область: физическую или логическую, а также позволяет перемещаться между уровнями кластера.

5 **Workspace** – область, в которой происходит создание сети, проводятся наблюдения за симуляцией и просматривается разная информация и статистика.

6 **Realtime/Simulation Bar** – с помощью закладок этой панели можно переключаться между режимом Real time и режимом Simulation. Она также содержит кнопки, относящиеся к Power Cycle Devices, кнопки Play Control и переключатель Event List в режиме Simulation.

7 **Network Component Box** – это область, в которой выбираются устройства и связи для размещения их на рабочем пространстве. Она содержит области Device-Type Selection и Device-Specific Selection.

8 **Device-Type Selection Box** – область содержит доступные типы устройств и связей в Packet Tracer. Область Device-Specific Selection изменяется в зависимости от выбранного устройства.

9 **Device-Specific Selection Box** – область используется для выбора конкретных устройств и соединений, необходимых для постройки в рабочем пространстве сети.

10 **User Created Packet Window** – это окно управляет пакетами, которые были созданы в сети во время симуляции сценария.

Для создания топологии необходимо выбрать устройство из панели Network Component, а затем из панели Device-Type Selection выбрать тип устройства. После этого нужно нажать левую кнопку мыши в поле рабочей области программы (Workspace). Также можно переместить устройство прямо из области Device-Type Selection, но при этом будет выбрана модель устройства по умолчанию.

Для быстрого создания нескольких экземпляров одного и того же устройства нужно, удерживая кнопку Ctrl, нажать на устройство в области Device-Specific Selection и отпустить кнопку Ctrl. После этого нужно несколько раз кликнуть мышью в рабочей области для добавления копий устройства.

В Packet Tracer представлены следующие типы устройств:

- маршрутизаторы;
- коммутаторы (в том числе и мосты);
- хабы и повторители;
- конечные устройства – ПК, серверы, принтеры, IP-телефоны;
- беспроводные устройства: точки доступа и беспроводный маршрутизатор;
- устройства – облако, DSL-модем и кабельный модем.

Добавим необходимые элементы в рабочую область программы так, как показано на рис.1.2.

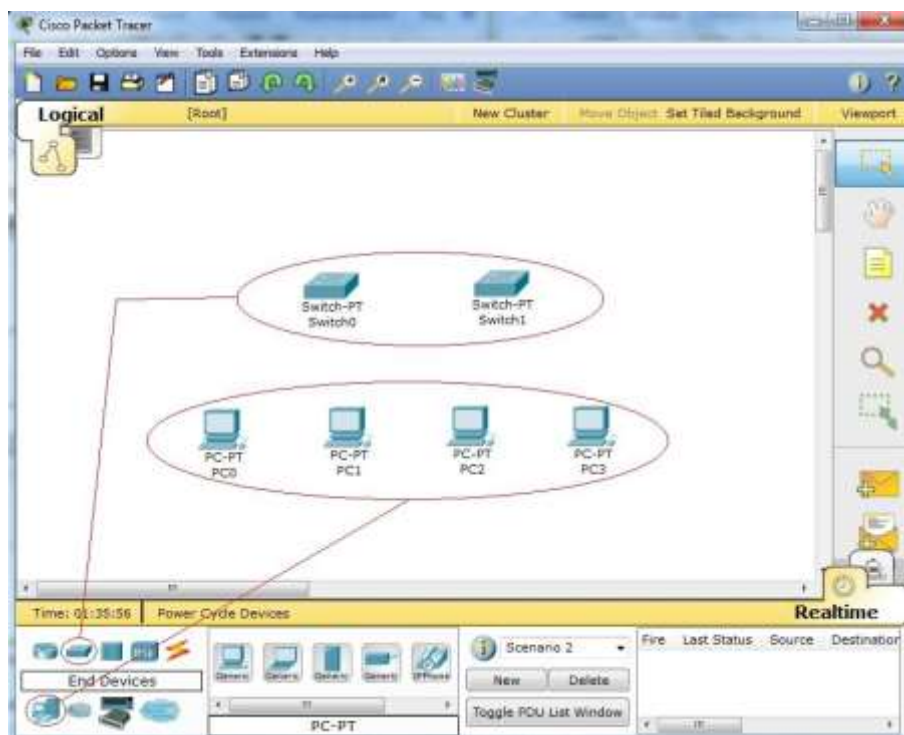


Рисунок 1.2. Добавление элементов сети

При добавлении каждого элемента пользователь имеет возможность дать ему имя и установить необходимые параметры. Для этого необходимо нажать на нужный элемент левой кнопкой мыши (ЛКМ) и в диалоговом окне устройства перейти к вкладке **Config**.

Диалоговое окно свойств каждого элемента имеет две вкладки:

- **Physical** – содержит графический интерфейс устройства и позволяет симулировать работу с ним на физическом уровне.
- **Config** – содержит все необходимые параметры для настройки устрой-

ства и имеет удобный для этого интерфейс.

Также в зависимости от устройства, свойства могут иметь дополнительную вкладку для управления работой выбранного элемента: Desktop (если выбрано конечное устройство) или CLI (если выбран маршрутизатор) и т.д.

Для удаления ненужных устройств с рабочей области программы используется кнопка Delete (Del).

Добавленные элементы свяжем с помощью соединительных связей. Для этого необходимо выбрать вкладку **Connections** из панели Network Component Box. Мы увидим все возможные типы соединений между устройствами. Выберем подходящий тип кабеля. Указание мыши изменится на курсор “connection” (имеет вид разъема). Кликнем мышью на первом устройстве и выберем соответствующий интерфейс, к которому нужно выполнить соединение, затем кликнем мышью на втором устройстве, выполнив ту же операцию. Можно также соединить элементы сети с помощью **Automatically Choose Connection Type** (автоматически соединяет элементы в сети). Выберем и нажмем на каждом из устройств, которые нужно соединить. Между устройствами появится кабельное соединение, а индикаторы на каждом конце покажут статус соединения (для интерфейсов, которые имеют индикатор).



Рисунок 1.3. Поддерживаемые в Packet Tracer типы кабелей

Packet Tracer поддерживает широкий диапазон сетевых соединений (см. табл. 1). Каждый тип кабеля может быть соединен лишь с определенными типами интерфейсов.

Таблица 1. Типы соединений в Packet Tracer

Тип кабеля	Описание
Console 	Консольное соединение может быть выполнено между ПК и маршрутизаторами или коммутаторами. Должны быть выполнены некоторые требования для работы консольного сеанса с ПК: скорость соединения с обеих сторон должна быть одинаковой, должно быть 7 бит данных (или 8 бит) для обеих сторон, контроль четности должен быть одинаковым, должно быть 1 или 2 стоповых бита для обеих сторон.
Copper Straight- through 	Этот тип кабеля является стандартной средой передачи Ethernet для соединения устройств, который функционирует на разных уровнях OSI. Он должен быть соединен со следующими типами портов: медный 10 Мбит/с (Ethernet), медный 100 Мбит/с (Fast Ethernet) и медный 1000 Мбит/с (Gigabit Ethernet).
Copper Cross- over 	Этот тип кабеля является средой передачи Ethernet для соединения устройств, которые функционируют на одинаковых уровнях OSI. Он может быть соединен со следующими типами портов: медный 10 Мбит/с (Ethernet), медный 100 Мбит/с (Fast Ethernet) и медный 1000 Мбит/с (Gigabit Ethernet).
Fiber 	Оптоволоконная среда используется для соединения между оптическими портами (100 Мбит/с или 1000 Мбит/с).
Phone 	Соединение через телефонную линию может быть осуществлено только между устройствами, имеющими модемные порты. Стандартное представление модемного соединения - это конечное устройство (например, ПК), дозванивающееся в сетевое облако.
Coaxial 	Коаксиальная среда используется для соединения между коаксиальными портами, такие как кабельный модем, соединенный с облаком Packet Tracer.
Serial DCE and DTE 	Соединения через последовательные порты, часто используются для связей WAN. Для настройки таких соединений необходимо установить синхронизацию на стороне DCE-устройства. Синхронизация DTE выполняется по выбору. Сторону DCE можно определить по маленькой иконке “часов” рядом с портом. При выборе типа соединения Serial DCE, первое устройство, к которому применяется соединение, становится DCE-устройством, а второе - автоматически станет стороной DTE. Возможно и обратное расположение сторон, если выбран тип соединения Serial DTE.

После создания сети ее нужно сохранить, выбрав пункт меню File -> Save или иконку Save на панели **Main Tool Bar**. Файл сохраненной топологии имеет расширение *.pkt.

Packet Tracer дает возможность симулировать работу с интерфейсом командной строки (ИКС) операционной системы IOS, установленной на всех коммутаторах и маршрутизаторах компании Cisco.

Подключившись к устройству, мы можем работать с ним так, как за консолью реального устройства. Симулятор обеспечивает поддержку практически всех команд, доступных на реальных устройствах.

Подключение к ИКС коммутаторов или маршрутизаторов можно произвести, нажав на необходимое устройство и перейдя в окне свойств к вкладке CLI.

Для симуляции работы командной строки на оконечном устройстве (компьютере) необходимо в свойствах выбрать вкладку Desktop, а затем нажать на ярлык Command Prompt.

Работа с файлами в симуляторе

Packet Tracer дает возможность пользователю хранить конфигурацию некоторых устройств, таких как маршрутизаторы или свитчи, в текстовых файлах. Для этого необходимо перейти к свойствам необходимого устройства и во вкладке Config нажать на кнопку “Export...” для экспорта конфигурации Startup Config или Running Config. Таким образом, получим диалоговое окно для сохранения необходимой конфигурации в файл, который будет иметь расширение *.txt. Текст файла с конфигурацией устройства running-config.txt (имя по умолчанию) представляется аналогичным к тексту информации, полученному при использовании команды show running-config в IOS устройствах.

Необходимо отметить, что конфигурация каждого устройства сохраняется в отдельном текстовом файле. Пользователь также имеет возможность изменять конфигурацию в сохраненном файле вручную с помощью произвольного текстового редактора. Для предоставления устройству сохраненных или

отредактированных настроек, нужно во вкладке Config нажать кнопку “Load...” для загрузки необходимой конфигурации Startup Config или кнопку “Merge...” для загрузки конфигурации Running Config.

Практическая часть

1. Добавим в рабочую область программы 2 коммутатора Switch-PT. По умолчанию они имеют имена – Switch0 и Switch1.
2. Добавим в рабочее поле четыре компьютера с именами по умолчанию PC0, PC1, PC2, PC3.
3. Соединим устройства в сеть Ethernet, как показано на рис.1.4.
4. Сохраним созданную топологию, нажав кнопку Save (в меню File -> Save).
5. Откроем свойства устройства PC0 кликнув мышью по его изображению. Перейдем к вкладке Desktop и симулируем работу run нажав Command Prompt.
6. Список команд получим, если введем команду ? и нажмем Enter. Для конфигурирования компьютера воспользуемся командой ipconfig из командной строки, например, для компьютера PC0:

```
ipconfig 192.168.1.2 255.255.255.0
```

IP адрес и маску сети также можно вводить в удобном графическом интерфейсе устройства (см. рис.1.4, и рис.1.5). В поле DEFAULT GATEWAY вводится адрес шлюза, в данном случае эта информация не важна, так как создаваемая сеть не требует маршрутизации.

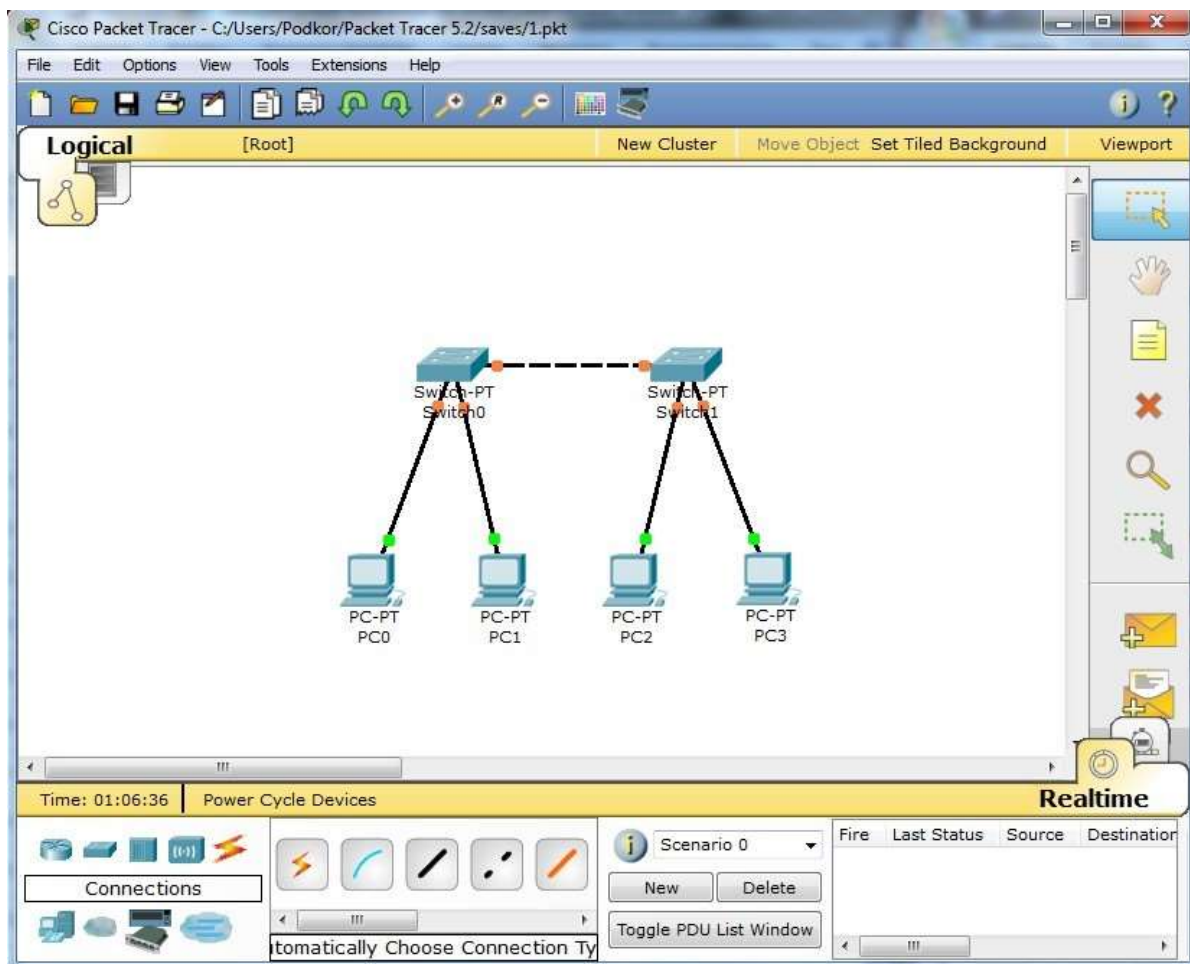


Рисунок 1.4. Созданный фрагмент сети

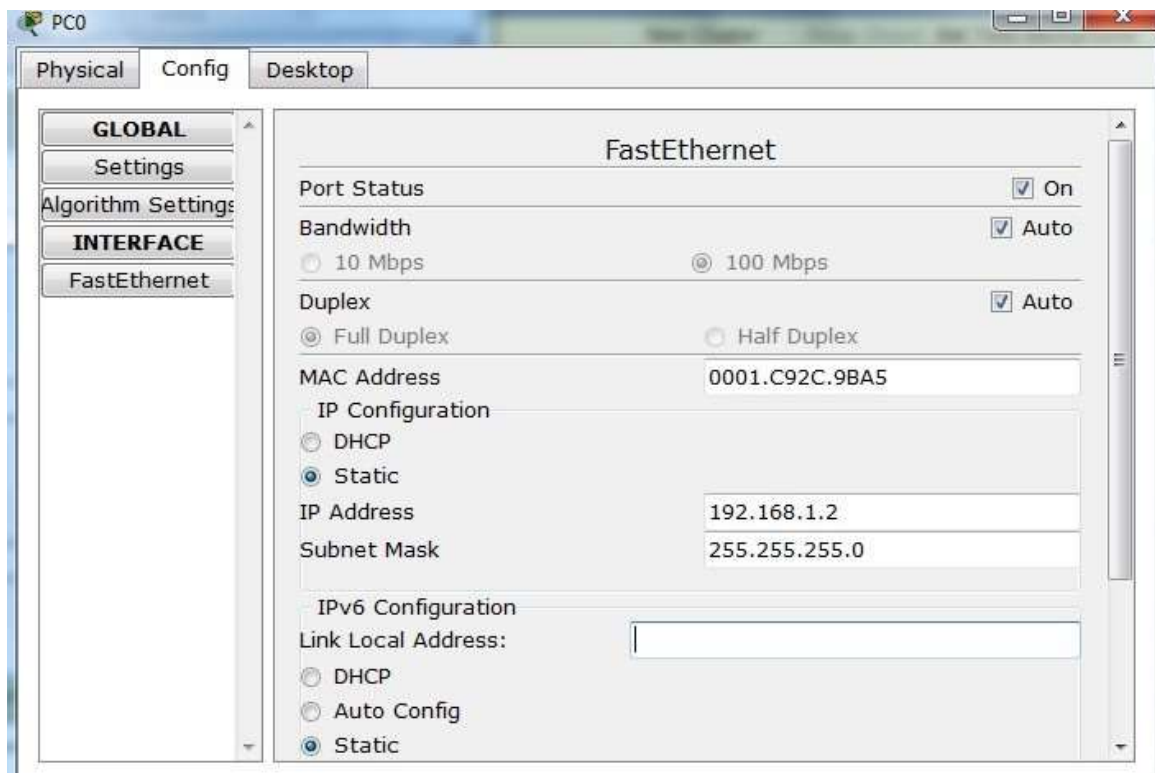


Рисунок 1.5. Настройка PC0

Таким же путем настроим каждый компьютер рис. 1.4.

Таблица 2. IP-адреса и маски компьютеров

Устройство	IP ADDRESS	SUBNET MASK
PC0	192.168.1.2	255.255.255.0
PC1	192.168.1.3	255.255.255.0
PC2	192.168.1.4	255.255.255.0
PC3	192.168.1.5	255.255.255.0

7. На каждом компьютере посмотрим назначенные адреса командой ipconfig без параметров. Если все сделано правильно, мы сможем проверить соединение компьютеров с использованием команды ping. Например, зайдем на компьютер PC3 и пропингуем его с компьютером PC0. Мы должны увидеть отчет о пинге, подобный рис. 1.6.

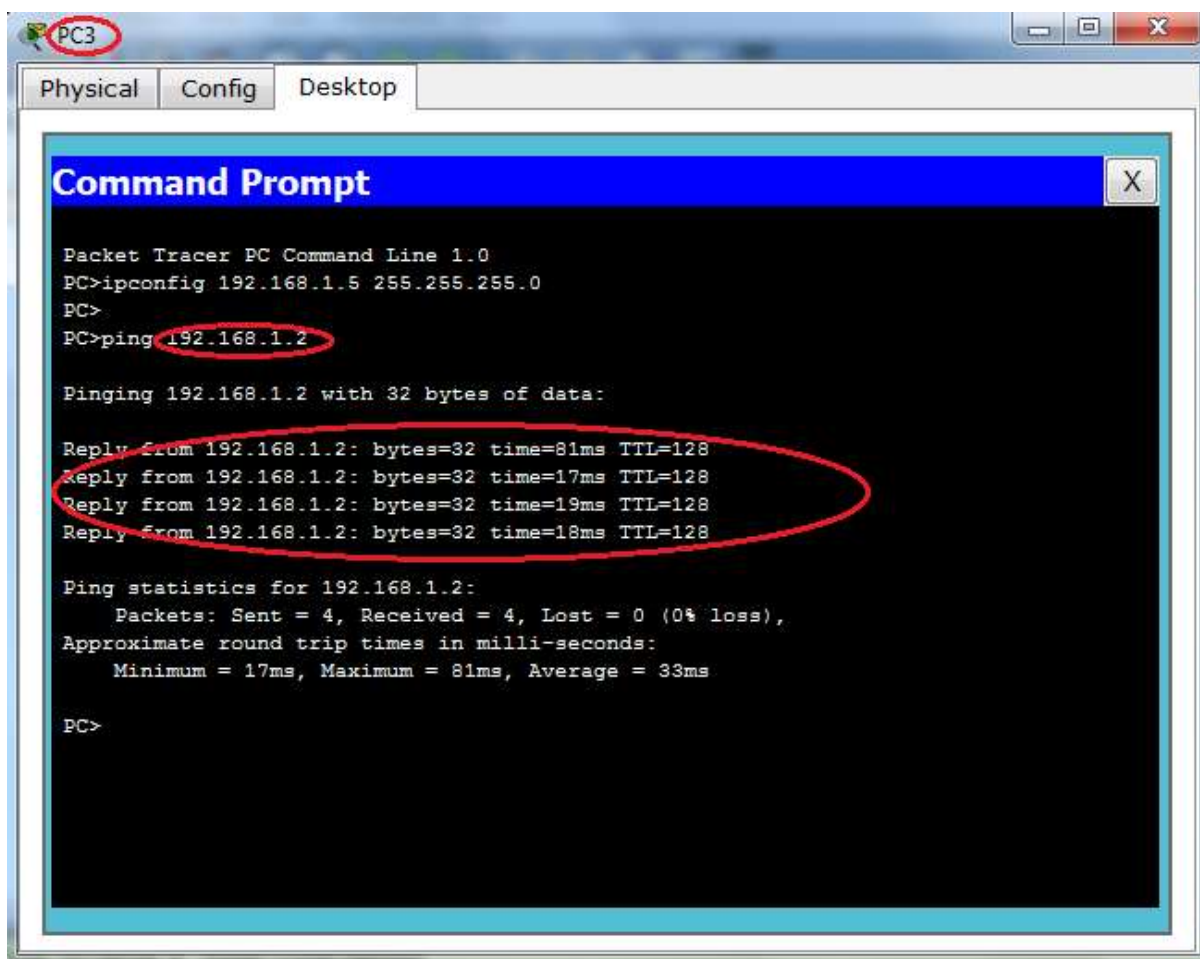


Рисунок 1.6. Отчет о пинге PC3

Задание для самостоятельной работы

1. Создайте топологию

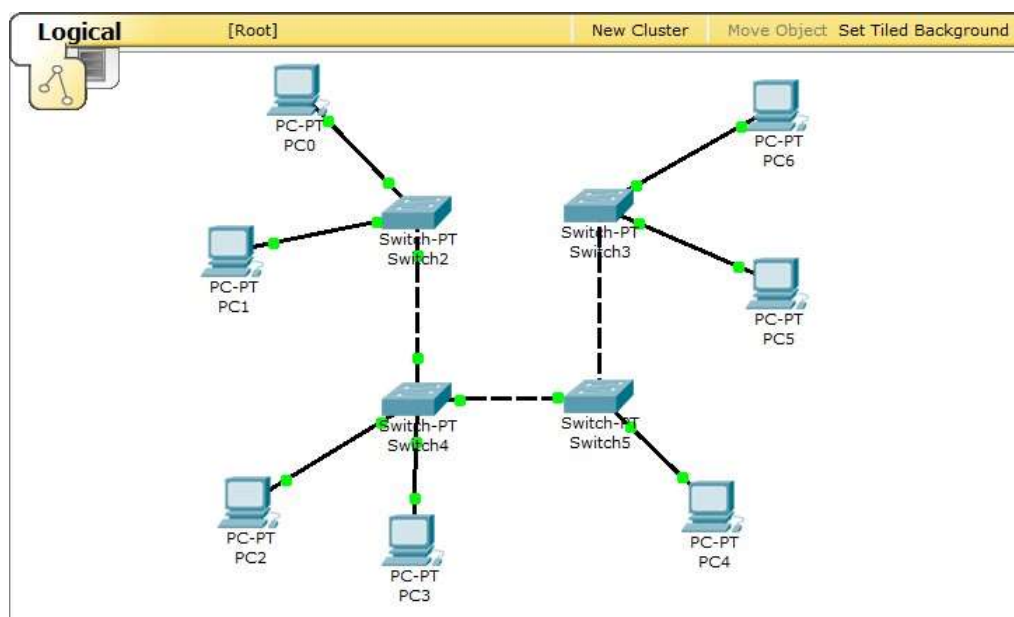


Рис 1.7.

2. Назначьте компьютерам адреса, согласно своему варианту

Таблица 3. IP адреса и маски компьютеров

Устройство	IP ADDRESS	SUBNET MASK
PC1	192.168.1.1	255.255.255.0
PC2	192.168.1.2	255.255.255.0
PC3	192.168.1.3	255.255.255.0
PC4	192.168.1.4	255.255.255.0
PC5	192.168.1.5	255.255.255.0
PC6	192.168.1.6	255.255.255.0
PC0	192.168.1.7	255.255.255.0

3. Назначьте компьютерам разные имена в командной строке.

4. Если сделано всё правильно, Вы сможете пропинговать любой компьютер из любого другого компьютера.

Таблица 4. Варианты заданий

Вариант №	Пинг из	Пинг в	Вариант №	Пинг из	Пинг в
1	PC1	PC6	7	PC0	PC5
2	PC2	PC0	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

Предполагается, создание реального фрагмента сети рис. 1.7 и демонстра-

ция команды ping преподавателю согласно своему варианту.

Порядок выполнения и сдачи лабораторной работы

1. Изучить теоретическую и практическую часть лабораторной работы.
2. Сдать преподавателю теорию работы путём ответа на контрольные вопросы.
3. Выполнить в Packet Tracer практическую часть лабораторной работы согласно своему варианту.
4. Предъявите преподавателю результат выполнения задания согласно своему варианту. Продемонстрируйте, что любой компьютер пингуется из любого компьютера.
5. Оформите и защитите отчет.
6. Отчёт должен содержать скриншоты созданной топологии, а также выполнение команды ping, согласно своему варианту.

Контрольные вопросы:

1. Какое максимальное количество устройств в сети поддерживает Packet Tracer?
2. Какие типы сетевых устройств и соединений можно использовать в Packet Tracer?
3. Каким способом можно перейти к интерфейсу командной строки устройства.
4. Как конфигурировать устройства из другого компьютера?
5. Как добавить в топологию и настроить новое устройство?
6. Как сохранить конфигурацию устройства в .txt файл?

Лабораторная работа 3. Определение параметров сетевого адаптера и настроек протокола IP

Цель работы: Получение навыков в настройке сетевых подключений в MS Windows XP и в работе с консолью.

Задание

1. Определите параметры сетевого адаптера ПК
2. Определить настройки протокола IP своего ПК.
3. Определение расширенных свойств сетевых подключений своего ПК и других ПК подсети.

Краткие теоретические сведения

IP-адрес (айпи-адрес, сокращение от англ. Internet Protocol Address) — уникальный сетевой адрес узла в компьютерной сети, построенной по протоколу IP. При связи через сеть Интернет требуется глобальная уникальность адреса, в случае работы в локальной сети требуется уникальность адреса в пределах сети. Имеет длину 4 байта. Обычно первый и второй байты определяют адрес сети, третий байт определяет адрес подсети, а четвертый - адрес компьютера в подсети

IP-адрес состоит из двух частей: номера сети и номера узла. В случае изолированной сети её адрес может быть выбран администратором из специально зарезервированных для таких сетей блоков адресов (192.168.0.0/16, 172.16.0.0/12 или 10.0.0.0/8). Если же сеть должна работать как составная часть Интернета, то адрес сети выдаётся провайдером.

Номер узла в протоколе IP назначается независимо от локального адреса узла. Маршрутизатор по определению входит сразу в несколько сетей. Поэтому каждый порт маршрутизатора имеет собственный IP-адрес. Конечный узел также может входить в несколько IP-сетей. В этом случае компьютер должен иметь несколько IP-адресов, по числу сетевых связей.

Таким образом, IP-адрес характеризует не отдельный компьютер или маршрутизатор, а одно сетевое соединение.

IP-адрес называют статическим (постоянным, неизменяемым), если он прописывается в настройках устройства пользователем, либо если назначается автоматически при подключении устройства к сети, но используется в течение неограниченного промежутка времени и не может быть присвоен другому устройству.

IP-адрес называют динамическим (непостоянным, изменяемым), если он назначается автоматически при подключении устройства к сети и используется в течение ограниченного промежутка времени, как правило, до завершения сеанса подключения.

Динамические IP-адреса также бывают виртуальными, обслуживание виртуального IP-адреса производится по технологии NAT: пользователям предоставляется возможность беспрепятственно получать информацию из сети Интернет, при этом теряется всякая возможность иного доступа к компьютеру из сети, так, например, компьютер с таким ip не может использоваться в качестве веб-сервера.

Не виртуальные ip называют реальными, прямыми, внешними, общественными или публичными, «белыми», все таковые ip являются статическими.

MAC-адрес (от англ. Media Access Control — управление доступом к среде, также Hardware Address) — это уникальный идентификатор, присваиваемый каждой единице оборудования компьютерных сетей. Большинство сетевых протоколов канального уровня используют одно из трёх пространств MAC-адресов, управляемых IEEE: MAC-48, EUI-48 и EUI-64. Адреса в каждом из пространств теоретически должны быть глобально уникальными. Не все протоколы используют MAC-адреса, и не все протоколы, использующие MAC-адреса, нуждаются в подобной уникальности этих адресов.

В широкополосных сетях (таких, как сети на основе Ethernet) MAC-адрес позволяет уникально идентифицировать каждый узел сети и доставлять данные только

этому узлу. Таким образом, MAC-адреса формируют основу сетей на канальном уровне, которую используют протоколы более высокого (сетевого) уровня. Для преобразования MAC-адресов в адреса сетевого уровня и обратно применяются специальные протоколы (например, ARP и RARP в сетях TCP/IP).

Адреса типа MAC-48 наиболее распространены; они используются в таких технологиях, как Ethernet, Token ring, FDDI, WiMAX и др. Они состоят из 48 бит, таким образом, адресное пространство MAC-48 насчитывает 248 (или 281 474 976 710 656) адресов.

ARP (англ. Address Resolution Protocol — протокол определения адреса) — использующийся в компьютерных сетях протокол низкого уровня, предназначенный для определения адреса канального уровня по известному адресу сетевого уровня. Наибольшее распространение этот протокол получил благодаря повсеместности сетей IP, построенных поверх Ethernet, поскольку практически в 100 % случаев при таком сочетании используется ARP.

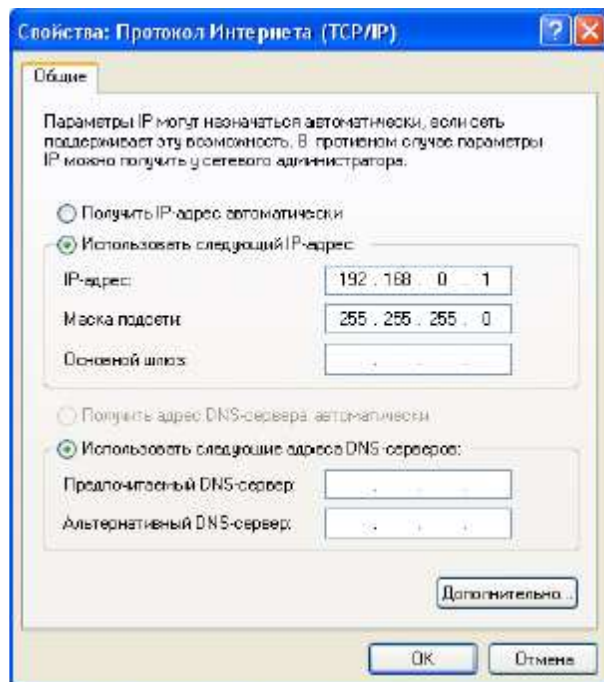
Существуют следующие типы сообщений ARP: запрос ARP (ARP request) и ответ ARP (ARP reply). Система-отправитель при помощи запроса ARP запрашивает физический адрес системы-получателя. Ответ (физический адрес узла-получателя) приходит в виде ответа ARP. Перед тем как передать пакет сетевого уровня через сегмент Ethernet, сетевой стек проверяет кэш ARP, чтобы выяснить, не зарегистрирована ли в нём уже нужная информация об узле-получателе. Если такой записи в кэше ARP нет, то выполняется широковещательный запрос ARP. Этот запрос для устройств в сети имеет следующий смысл: «Кто-нибудь знает физический адрес устройства, обладающего следующим IP-адресом?» Когда получатель с этим IP-адресом примет этот пакет, то должен будет ответить: «Да, это мой IP-адрес. Мой физический адрес следующий: ...» После этого отправитель обновит свой кэш ARP и будет способен передать информацию получателю. Записи в кэше ARP могут быть статическими и динамическими. Пример, данный выше, описывает динамическую запись кэша. Можно также создавать статические записи в таблице ARP. Это можно сделать при помощи команды:

```
arp -s <IP-адрес> <MAC-адрес>
```

Записи в таблице ARP, созданные динамически, остаются в кэше в течение 2-х минут. Если в течение этих двух минут произошла повторная передача данных по этому адресу, то время хранения записи в кэше продлевается ещё на 2 минуты. Эта процедура может повторяться до тех пор, пока запись в кэше просуществует до 10 минут. После этого запись будет удалена из кэша, и будет отправлен повторный запрос ARP.

Указания к выполнению работы

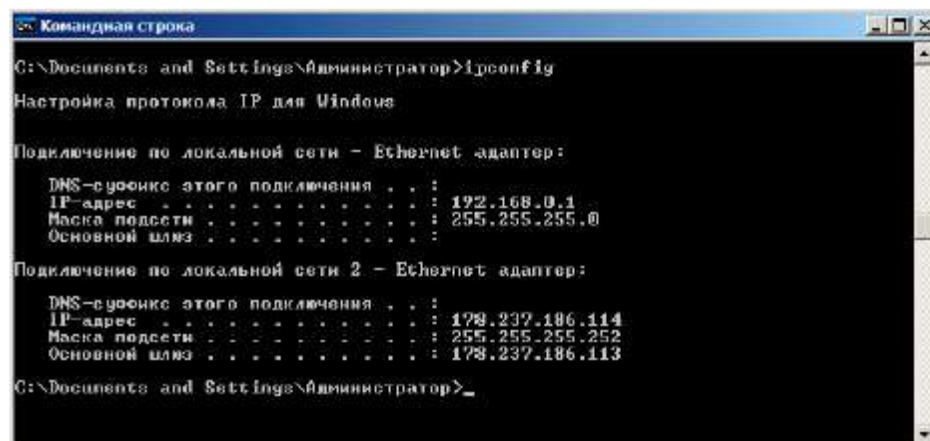
Задание 1. Определите параметры сетевого адаптера ПК, для чего командой Пуск-Настройки-Панель управления откройте окно Панель управления, затем щелкните на ярлыке Сетевые подключения. В окне Сетевые подключения выберите сетевое подключение и, щелкнув правой кнопкой мыши, выберите в контекстном меню команду Свойства.



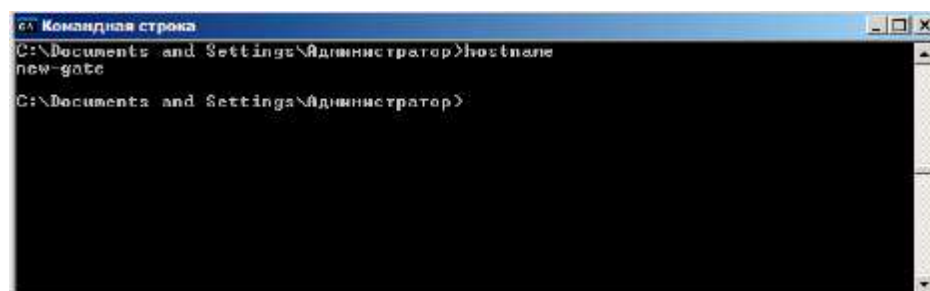
В окне Подключение по локальной сети - свойства укажите в списке компонент Протокол Интернета TCP/IP и щелкните на кнопке «Свойства». После этого откроется окно Свойства: Протокол Интернета (TCP/IP). Запишите IP-адрес ПК в отчет.

Задание 2. Определите настройки протокола IP вашего компьютера.

Для этого запустите программу ipconfig. Чтобы получить доступ к командной строке, выберите в меню Пуск команду Программы-Стандартные-Командная строка. В окне Командная строка введите команду ipconfig.



Для того чтобы определить имя локального компьютера, задайте команду Hostname.



Занесите в отчет сведения об адресации всех сетевых интерфейсов ПК, а также его символическое имя.

Задание 3. Получение расширенных свойств сетевых подключений ПК.

В окне командной строки наберите команду `ipconfig /all`. Результатом ее выполнения является таблица расширенных свойств всех сетевых адаптеров ПК, включая их MAC-адреса. Данные адреса необходимо отразить в отчетах.

```
cmd Командная строка
C:\Documents and Settings\Администратор>ipconfig /all

Настройка протокола IP для Windows

Имя компьютера . . . . . : new-gate
Основной DNS-суффикс . . . . . : 
Тип узла . . . . . : неизвестный
IP-маршрутизация включена . . . . . : нет
WINS-прокси включен . . . . . : да

Подключение по локальной сети - Ethernet адаптер:

DNS-суффикс этого подключения . . . : 
Описание . . . . . : Realtek PCIe GBE Family Controller
Физический адрес . . . . . : E0-CB-4E-B4-55-D1
DHCP включен . . . . . : нет
IP-адрес . . . . . : 192.168.0.1
Маска подсети . . . . . : 255.255.255.0
Основной шлюз . . . . . : 

Подключение по локальной сети 2 - Ethernet адаптер:

DNS-суффикс этого подключения . . . : 
Описание . . . . . : Realtek RTL8139 Family PCI Fast Ethernet
NIC
Физический адрес . . . . . : 00-80-48-22-EF-8F
DHCP включен . . . . . : нет
IP-адрес . . . . . : 178.237.186.114
Маска подсети . . . . . : 255.255.255.252
Основной шлюз . . . . . : 178.237.186.113
DNS-серверы . . . . . : 91.207.136.36
91.207.136.55

C:\Documents and Settings\Администратор>
```

Для определения MAC и IP-адресов ПК, находящихся в одной подсети с исследуемым ПК, необходимо выполнить последовательно две консольных команды: `ping [IP]` и `arp -a`. Результатом выполнения должна стать таблица с данными об IP и MAC-адресах

```
cmd Командная строка
C:\Documents and Settings\Администратор>arp -a

Интерфейс: 192.168.0.1 --- 0x3
IP-адрес          Физический адрес      Тип
192.168.0.12      f4-6d-04-99-30-a5     динамический
192.168.0.19      00-1d-92-3b-53-7e     динамический
192.168.0.21      6c-f0-49-a3-a1-e9     динамический
192.168.0.22      00-18-f3-0a-d7-71     динамический
192.168.0.23      00-24-1d-c8-8a-24     динамический
192.168.0.28      00-19-db-b7-57-8a     динамический
192.168.0.29      00-25-22-01-6a-14     динамический
192.168.0.37      00-21-85-3c-71-51     динамический
192.168.0.39      00-19-db-c3-b6-6b     динамический
192.168.0.41      00-1b-11-b2-03-56     динамический
192.168.0.46      00-25-22-85-44-c7     динамический
192.168.0.47      00-1b-11-b2-03-4d     динамический
192.168.0.48      00-1d-7d-e9-d1-d6     динамический
192.168.0.49      00-21-85-05-71-50     динамический
192.168.0.50      00-21-85-06-7b-40     динамический
192.168.0.71      00-21-85-06-7b-44     динамический
192.168.0.72      00-21-85-06-77-0f     динамический
192.168.0.141     00-19-66-e8-9f-6f     динамический
192.168.0.142     70-e6-ba-30-be-0f     динамический
192.168.0.181     00-1c-f0-d2-86-04     динамический

Интерфейс: 178.237.186.114 --- 0x20002
IP-адрес          Физический адрес      Тип
178.237.186.113   00-26-cb-9d-20-1b     динамический

C:\Documents and Settings\Администратор>
```

Занесите сведения об адресации других ПК подсети в отчеты.

Контрольные вопросы

1. Из каких частей состоит IP-адрес компьютера (IPv4)?
2. Что такое MAC-адрес ПК?.
3. Перечислите основные способы определения IP и MAC адресов сетевых интерфейсов.

4. Охарактеризуйте типы сообщений протокола ARP.

Лабораторная работа №4. Планирование подсетей

Цель и содержание:

Цель

Изучить методы определения хостов и выделения подсетей

Содержание

1. Понятие IP-адресации.
2. Выделение подсетей.
3. Маски подсетей.
4. Использование калькулятора Windows для решения сетевых задач.

Теоретическое обоснование:

Что такое IP-адресация?

Одной из наиболее важных тем при обсуждении TCP/IP является IP-адресация. *IP-адрес* представляет собой числовой идентификатор, присваиваемый каждому компьютеру сети IP. Он отражает расположение устройства в сети. IP-адрес является программным, а не аппаратным адресом – последний «зашит» в компьютере или плате сетевого интерфейса.

Иерархическая схема IP-адресации

IP-адрес содержит 32 бита информации, которые разделяются на четыре однобайтовые (восемьбитовые) секции, иначе называемые октетами. Существуют три способа представления IP-адресов:

- представление десятичными числами, разделенными точками, например, 130.57.30.56;
- двоичное представление, например, 10000010.00111001.00011110.00111000;
- шестнадцатеричное представление, например, 82 39 1E 38.

Здесь показаны три формы представления одного и того же IP-адреса. 32-битовый IP-адрес имеет иерархическую, а не плоскую (неиерархическую) структуру. Можно пользоваться любой схемой адресации, но иерархическая – более привлекательна.

Хороший пример неиерархической схемы адресации – нумерация карточек социального страхования. В этих номерах нет какого-либо деления на секции, содержимое которых отражало бы местожительство или иные характеристики обладателя карточки. Достоинством плоской схемы является большое число возможных адресов, например, пространство 32-битовых адресов, в каждой позиции которых может быть 0 или 1, содержит 2 или 4.2 миллиарда адресов. Неудобство такой схемы состоит в невозможности осуществления маршрутизации, именно по этой причине плоская схема не используется для IP-адресации. Если все адреса различны, маршрутизаторам Интернета придется запоминать адреса всех машин, подключенных к Интернету, что не позволит добиться эффективной маршрутизации, даже при использовании лишь части адресного пространства.

Поэтому для адресации выбрана иерархическая схема с тремя уровнями иерархии: сеть, подсеть и хост. Для примера рассмотрим структуру телефонного номера. Первая его часть (код региона) описывает обширную географическую область. Вторая часть (префикс) сужает эту область до зоны действия локальной телефонной станции. Последний сегмент (собственно номер телефона) определяет конкретное соединение. При IP-адресации также используется схема с тремя уровнями. Вместо того чтобы рассматривать 32-битовую комбинацию как единый идентификатор, в адресе выделяются части для адреса сети и для адреса узла.

Адрес сети однозначно определяет сеть. В IP-адресах всех машин, подключен или к одной сети, указывается один и тот же адрес сети. Например, в IP-адресе 172.16.30.56 адресом сети может быть 172.16.

Адрес узла присваивается каждой машине сети. В отличие от адреса сети, описывающего группу устройств, адрес узла уникален и однозначно определяет конкретную машину сети. Адрес узла называют также адресом хоста. В приведенном примере адрес узла имеет вид 30.56.

Разработчики Интернета решили создать классы сетей, причем сеть относится к тому или иному классу в зависимости от своего размера. Для небольшого количества сетей со значительным числом узлов создан *класс А*. *Класс С* зарезервирован для многочисленных сетей с малым количеством узлов. К сетям *класса В* относятся сети промежуточного размера. Способ разделения IP-адреса на поля адресов сети и узла определяется принадлежностью сети к тому или иному классу. В таблице 1.1 приведены сведения о трех классах сетей.

Таблица 1.1 – Характеристика классов сетей

Класс	Формат	Начальная битовая комбинация	Диапазон десятичных значений первого байта адреса сети	Максимальное число сетей	Максимальное число узлов в сети
A	Сеть.Узел.Узел.Узел	0	1-127	127	16 777 214
B	Сеть.Сеть.Узел.Узел	10	128-191	16384	65 534
C	Сеть.Сеть.Сеть.Узел	110	192-223	2097152	254

Для эффективной маршрутизации разработчики Интернета определяют обязательную начальную часть адреса для каждого из классов. Например, зная, что адрес сети класса А всегда начинается с 0, маршрутизатор может ускорить передачу пакета по нужному маршруту после распознавания содержимого первого бита адреса. На рисунке 1.1 показано соответствие между классом сети и содержимым начальных битов адреса.

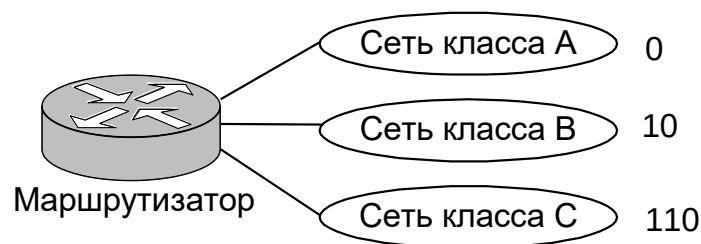


Рисунок 1.1 – Начальные битовые комбинации сети

Некоторые IP-адреса зарезервированы для специального применения, и администратор сети не вправе приписывать их узлам сети. Зарезервированные адреса перечислены в таблице 1.2.

Таблица 1.2 – Зарезервированные IP-адреса

Адрес	Назначение
Адрес сети, состоящий из нулей	Интерпретируется как «текущая сеть» или «текущий сегмент»
Адрес сети, состоящий из единиц	Интерпретируется как «все сети».
Сеть 127	Зарезервировано для кольцевых тестов. Обозначает локальный узел и используется, когда узел посылает тестовый пакет самому себе без создания дополнительного трафика
Адрес узла, состоящий из нулей	Интерпретируется как «текущий узел»
Адрес узла, состоящий из единиц	Интерпретируется как «все узлы» определенной сети; например, адрес 128.2.255.255 описывает все узлы сети 128.2 (адрес класса В).
Нулевой IP-адрес	Используется маршрутизаторами Cisco для обозначения маршрута, выбранного по умолчанию.
IP-адрес, состоящий из единиц (или 255.255.255.255)	Широковещательный адрес (адрес сообщений, направляемых всем узлам текущей сети)

Сети класса А

В IP-адресе сетей класса А первый байт занимает адрес сети, а в трех последующих байтах размещается адрес узла. Формат IP-адреса сети класса А:

Сеть.Узел.Узел.Узел.

Например, в IP-адресе 49.22.102.70 адрес сети равен 49, а адрес узла – 22.102.70. Каждая машина этой сети должна иметь адрес сети, равный 49.

Адрес сети класса А имеет длину 1 байт, причем его первый бит зарезервирован (см. таблицу 1.1). Это означает, что можно создать не более 128 сетей класса А. Так как каждый из семи оставшихся битов может принимать значение 0 или 1, т. е. существует 2 или 128 различных комбинаций. Было решено, что нулевой адрес сети (0000 0000) резервируется для обозначения маршрута, выбранного по умолчанию (см. таблицу 1.2). Поэтому фактическое число используемых адресов сетей класса А составляет $128 - 1 = 127$.

Вы можете определить это число самостоятельно. Начните со строки с двоичным нулем и спускайтесь по строкам вниз, обращая внимание на значение первого самого левого) бита. Десятичное число в строке, где этот бит впервые становится 1, равно 127. Иначе говоря, номера сетей класса А выбираются из диапазона от 0 до 127. Так как нулевой адрес зарезервирован, диапазон становится уже: от 1 до 127. В таблице 1.2 указано, что сеть с номером 127 также является особой. Таким образом, точное число сетей класса А равно 126.

Под адрес узла в IP-адресе сетей класса А отведено 3 байт (24 бит). В них можно разместить 2^{24} или 16 777 216 различных двоичных комбинаций или адресов узлов. Поскольку адреса, состоящие только из нулей и только из единиц, зарезервированы, точное число узлов в сети класса А составляет 16 777 214.

Сети класса В

В IP-адресе сетей класса В первые два байта занимает адрес сети, а в двух последующих байтах размещается адрес узла. Формат IP-адреса сети класса В:

Сеть.Сеть.Узел.Узел

Например, в IP-адресе 172.16.30.56 адрес сети равен 172.16, а адрес узла – 30.56.

Для адреса сети, состоящего из 16 бит, имеется 2^{16} возможных комбинаций. Разработчики Интернета решили, что адрес сети класса В

должен начинаться с комбинации 10. Поэтому свободными для формирования адреса остаются лишь 14 бит; это означает, что может существовать 2 или 16 384 сетей класса В.

Под адрес узла в IP-адресе сетей класса В отведено 2 байта. Поскольку адреса, состоящие только из нулей и только из единиц, зарезервированы, точное число сетей класса В равно $2^{16} - 2 = 65\,534$.

Сети класса С

Первые три байта в IP-адресе сетей класса С занимает адрес сети, и всего байт остается для адреса узла. Формат IP-адреса сети класса С:

Сеть.Сеть.Сеть.Узел

Например, в IP-адресе 192.168.100.102 адрес сети равен 192.168.100, а адрес Узла – 102.

Первые три бита адреса сети класса С занимает комбинация 110. Поэтому для формирования адреса остается лишь $24 - 3 = 21$ бит. Таким образом, может существовать 2^{21} или 2 097 152 сетей класса С.

Дополнительные классы сетей

Еще одним классом сетей является класс D. Диапазон адресов этого класса (от 224.0.0.0 до 239.255.255.255) используется для групповой рассылки пакетов.

Существует также класс сетей E с диапазоном адресов от 240.0.0.0 до 255.255.255.255. Эти адреса зарезервированы для использования в будущем.

Если вы не хотите усложнять себе жизнь, не присваивайте узлам своей сети адреса из классов D и E.

Подсети

Если в организации много компьютеров или они далеко стоят друг от друга, имеет смысл разбить большую сеть на несколько более мелких и соединить их через маршрутизаторы. К достоинствам такого подхода относятся:

- сокращенный сетевой трафик. Все ощутят снижение трафика любого типа. При этом сети остаются однотипными. Без маршрутизаторов

трафик грозит почти полностью затормозить, работу сети. А при их использовании большая часть трафика остается в локальной сети; через маршрутизатор будут передаваться лишь пакеты, предназначенные потребителям из других сетей;

- оптимизация производительности сети. Это премия за сокращение сетевого трафика;
- упрощенное управление. В группе небольших сетей, связанных друг с другом, гораздо легче выявить и решить возникающие проблемы, чем в одной большой сети;
- упрощенный охват больших географических пространств. Связи глобальных сетей намного медленнее и более дорогостоящие по сравнению со связями локальных сетей. Объединение многочисленных мелких сетей делает всю систему более эффективной.

Если организация с многочисленными сетями получает в Сетевом информационном центре только один сетевой адрес, возникают различные проблемы. Первые разработчики протокола IP брали в расчет значительно меньшую по объему Интернет, состоящую из десятка сетей и сотен хостов. В их схеме адресации для каждой физической сети предусматривался отдельный адрес.

Такой подход привел к появлению некоторых проблем. Вот первая: один и тот же адрес присваивается нескольким физическим сетям. Однако организация может потребовать отдельный адрес для каждой физической сети. Если бы все подобные запросы удовлетворялись, проблема адресации была бы решена.

Следующая проблема относится к применению маршрутизаторов. Если бы каждый маршрутизатор в Интернете должен был знать о каждой существующей физической сети, таблицы маршрутизации невероятно разрослись бы. Для поддержки таких таблиц потребовались бы дополнительные расходы на администрирование и увеличение физических

затрат на маршрутизаторы (время цикла процессора, память, дисковое пространство и т. д.).

Нежелательным последствием явился бы огромный рост сетевого трафика, поскольку маршрутизаторы обмениваются друг с другом информацией о маршрутизации.

Как же организовать подсети? Чтобы в единой сети IP выделить несколько более мелких логических подсетей, можно воспользоваться особенностью программной реализации TCP/IP. Особенность состоит в том, что часть битов из адреса хоста в IP-адресе отводится под так называемый адрес подсети.

Методика и порядок выполнения работы:

Методика

Выделение подсетей

Выделение подсетей реализуется за счет присвоения некоторого адреса подсети каждой машине заданной физической сети. Например, на рисунке 1.2 каждая машина подсети 4 имеет адрес подсети, равный 4.

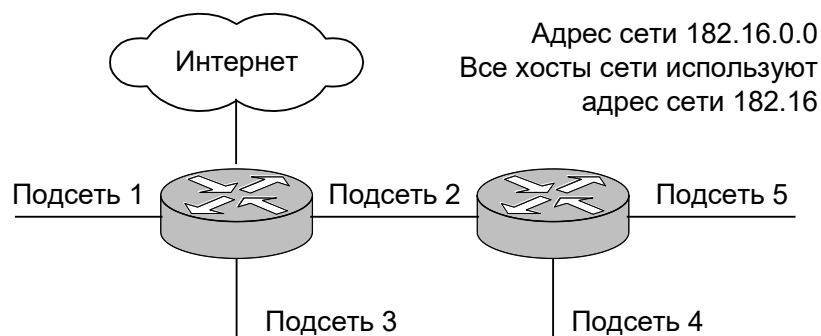


Рисунок 1.2 – Использование подсетей

Рассмотрим порядок размещения адреса подсети в IP-адресе. Часть IP-адреса, содержащую адрес сети, изменять нельзя, поскольку все машины сети используют его совместно. Например, все машины сети Асте имеют адрес сети, равный 182.16. Поэтому изменять можно только вторую часть IP-адреса. В схеме станции с подсетями часть адреса хоста рассматривается

как адрес подсети. На рисунке 1.3 показано, как в IP-адрес может быть встроен адрес подсети.

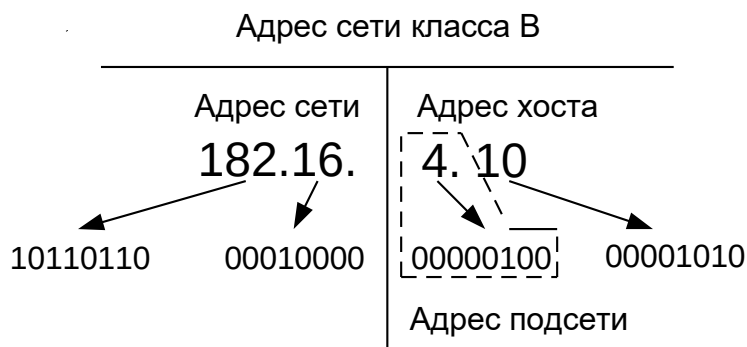


Рисунок 1.3 – Адрес подсети может быть встроен в IP-адрес только в части, относящейся к адресу хоста

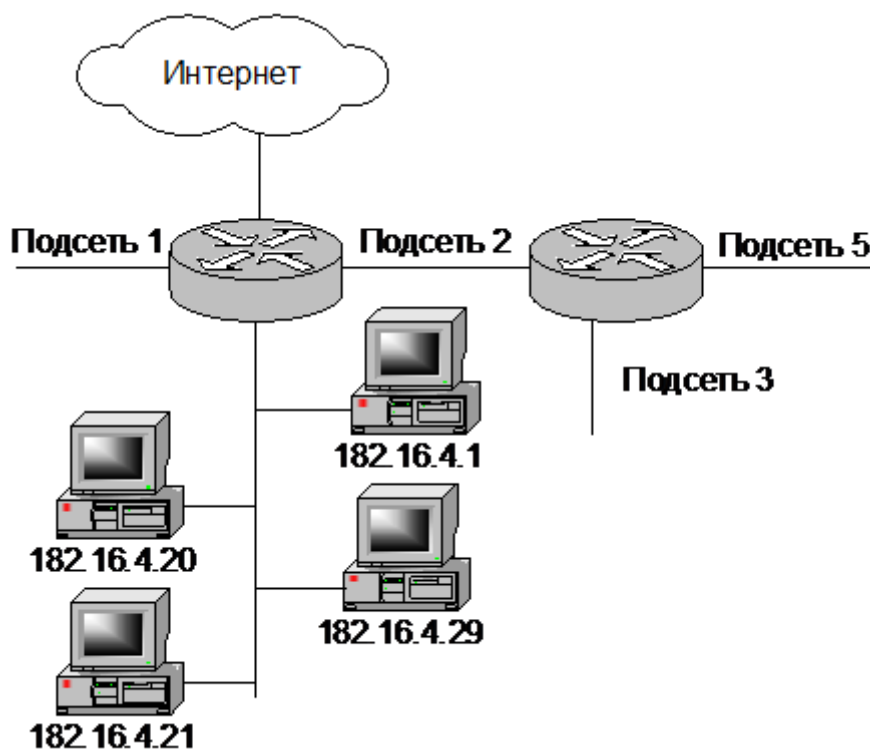


Рисунок 1.4 – Адрес сети и адрес подсети

Поскольку сеть Аспе относится к классу В, в первых двух байтах IP-адреса содержится адрес сети, совместно используемый всеми машинами сети, независимо от того, к каким подсетям они принадлежат. Адрес каждой машины подсети 4 должен иметь в третьем байте число 0000 0100. Четвертый байт (адрес хоста) содержит уникальное число, идентифицирующее конкретный хост. Таким образом, в состав IP-адреса входят три поля: адрес сети, адрес подсети и адрес хоста.

На рисунке 1.4 показано, как используются адреса сети и подсети. Этот подход применим к любой подсети, выделенной в составе сети.

Маски подсетей

При применении схемы адресации с подсетями каждая машина сети должна знать, какая часть адреса хоста занята адресом подсети. Для этого на каждой машине создается маска подсети.

Кодирование разрядов маски подсети

Единицами помечено положение адреса сети и подсети
Нулями помечено положение адреса хостов

Маска подсети для компании Acme Inc.

11111111 . 11111111	11111111	00000000
Позиция адреса сети	Позиция маски подсети	Позиция адреса хоста

Рисунок 1.5 – Маска подсети

Администратор сети создает 32-битовую маску подсети, состоящую из 0 и 1. Единицы в маске подсети помечают позиции, относящиеся к адресам сети и подсети. Нули заносятся в позиции, отведенные под адрес хоста. Эта идея проиллюстрирована на рисунке 1.5.

В примере первые два байта маски подсети заполнены единицами, поскольку сети Acme относится к классу В (формат Сеть.Сеть.Узел.Узел). Третий байт, обычно относящийся к адресу хоста, представляет собой адрес подсети. Поэтому все его биты заполнены единицами. И только в четвертом байте хранится уникальный адрес хоста.

Маска подсети может быть также представлена в десятичном формате. Двоичная комбинация 1111 1111 соответствует десятичному числу 255. Таким образом, маску подсети нашего примера можно отобразить двумя способами. Эта идея проиллюстрирована на рисунке 1.6

Двоичное представление 11111111.11111111.11111111.00000000 Десятичное представление: 255.255.255.0

Рисунок 1.6 – Просмотр IP-адреса с помощью маски подсети

Совсем необязательно, чтобы в состав сети входили подсети, т. е. маска подсети может и не использоваться. В подобных случаях говорят, что существует маска подсети, принятая по умолчанию. Маски, принятые по умолчанию для различных классов сетей (см. таблицу 1.3), изменять нельзя. Иначе говоря, применять в качестве маски подсети маску 255.0.0.0 не разрешается. Хост, которому вы попытаетесь присвоить такую маску, посчитает ее некорректной и не позволит завершить данную операцию. Для сетей класса А нельзя изменять первый байт маски подсети, поэтому она должна начинаться с числа 255. Кроме того, недопустима маска 255.255.255.255, поскольку эта конфигурация обозначает широковещательный адрес. Маска подсети для сетей класса В должна начинаться 255.255, а для сетей класса С – с 255.255.255.

Таблица 1.3 – Маски подсетей, принятые по умолчанию

Класс сети	Формат адреса	Маска подсети
А	Сеть.Узел.Узел.Узел	255.0.0.0
В	Сеть.Сеть.Узел.Узел	255.255.0.0
С	Сеть.Сеть.Сеть.Узел	255.255.255.0

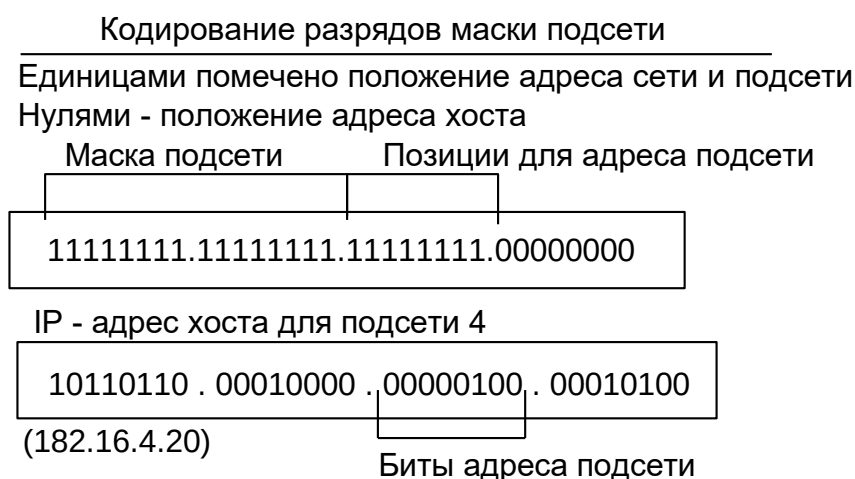


Рисунок 1.7 – Просмотр IP-адреса с помощью маски подсети

После того как администратор сети создаст маску подсети для каждой машины, программное обеспечение протокола IP будет просматривать IP-

адреса, используя маску подсети для определения адреса подсети, Соответствующая операция представлена на рисунке 1.7.

В этом примере по маске подсети программно определяется, что третий байт IP-адреса больше не относится к адресу хоста, а занят адресом подсети, равным 0000 0100.

Если для размещения адреса подсети использовать третий байт IP-адреса класса В, то задание и определение адреса подсети облегчается. Например, если в сети Acme Inc. необходимо определить подсеть 12, в третий байт адресов всех машин этой подсети следует вставить 0000 1100.

При использовании всего третьего байта IP-адреса класса В для адреса подсети можно создать довольно много подсетей. Действительно, так как байт состоит из восьми битов, можно определить $2^8 = 256$ различных адресов. Поскольку адрес не может состоять только из единиц или нулей, число возможных подсетей уменьшается до 254. Таким образом, в сети Acme Inc. можно выделить до 254 подсетей, каждая из которых будет содержать до 254 хостов.

Хотя в официальной спецификации протокола IP применение нулевого адреса подсети ограничено, в некоторых программных продуктах такой адрес все же используется. Например, добавление специальных команд для маршрутизатора Cisco позволяет вводить подсети с нулевым адресом, увеличивая число возможных подсетей до 255. Однако по умолчанию такое изменение невозможно.

IP-адрес: 182.16.52.10	Помеченные биты
Маска подсети: 11111111 . 11111111 . 111 00000 . 00000000	
255 . 255 . 224 . 0	

$2^{(\text{число помеченных битов})} - 2 = \text{число подсетей}$

$2^3 - 2 = 6 \text{ подсетей}$

$2^{(\text{число непомеченных битов})} - 2 = \text{число хостов}$

$2^{13} - 2 = 8190 \text{ хостов в подсети}$

Рисунок 1.8 – Формулы для подсчета числа подсетей и хостов

Приведем формулы для подсчета максимального числа подсетей и хостов в подсети:

$2^{(\text{число помеченных битов в маске подсети})} - 2 = \text{максимальное число подсетей};$

$2^{(\text{число непомеченных битов в маске подсети})} - 2 = \text{максимальное число хостов в подсети}.$

Помеченными считаются биты с единичным значением, а *непомеченными* – нулевые биты. На рисунке 1.8 приведен пример применения этих формул.

Таким образом, если число помеченных битов равно 3, согласно приведенным формулам можно создать до шести подсетей. Для адресации хостов остается 5 бит в третьем байте и 8 бит в четвертом. При 13-битовых адресах к подсети можно подключить до $2^{13} - 2 = 8190$ хостов.

При возврате к применению полного байта адреса узла в качестве адреса подсети (маска 255.255.255.0) возможное количество узлов в каждой подсети сократится. Без выделения подсетей адрес класса В может иметь 65 534 различных значений, которые разрешается использовать в качестве адреса узлов.

Если адрес подсети занимает полный байт адреса узла, остается только один байт для адресов хостов, т. е. в подсети можно разместить до 254 хостов. Когда в какой-нибудь подсети потребуется увеличить это число,

возникают затруднения. Для их разрешения необходимо либо укоротить поле адреса подсети в маске и тем самым увеличить длину адреса хоста, либо добавить вторичный IP-адрес к интерфейсу маршрутизатора, что также позволит увеличить число хостов в подсети. Однако побочным эффектом такого решения будет уменьшение количества образуемых подсетей.

Использование калькулятора Windows для решения сетевых задач

Сетевые техники работают на компьютерах и сетевых устройствах с двоичными, десятичными и шестнадцатеричными числами. Требуется:

с помощью калькулятора Windows преобразовывать числа в двоичную, десятичную и шестнадцатеричную системы счисления;

с помощью степенной функции определять число узлов, которым можно назначить адреса, на основе доступного числа разрядов.

Шаг 1: включение калькулятора Windows и определение его режима работы

а. В меню «Пуск» выберите «**Все программы > Стандартные**», а затем – «**Калькулятор**» рис 1.9.



Рисунок 1.9 – Внешний вид калькулятора

Альтернативный метод запуска приложения «Калькулятор» – открыть меню «**Пуск**», выбрать команду «**Выполнить**», ввести **calc** и нажать клавишу «**Enter**». Попробуйте оба метода.

б. После запуска приложения «Калькулятор» откройте меню «**Вид**».

в. Определите, какой режим [Обычный | Инженерный] является активным в данный момент?

г. Выберите режим «Стандартный». Это основной режим для простых вычислений. Посмотрите, сколько математических функций доступно в этом режиме? (Восемь)

д. В меню «Вид» выберите пункт режим «Инженерный».

Шаг 2: переход между системами счисления

а. Перейдите в режим «**Инженерный**». Обратите внимание на доступные режимы систем счисления – Hex (шестнадцатеричная), Dec (десятичная), Oct (восьмеричная) и Bin (двоичная).

б. Посмотрите, какая система счисления используется в данный

момент? (Dec)

в. Уясните, что цифры (0 – 9) на цифровой клавиатуре активны в десятичном режиме.

г. Установите переключатель **Bin** (двоичная). Какие цифры на цифровой клавиатуре активны в данный момент? (0 и 1)

д. Почему другие цифры недоступны?

е. Установите переключатель **Hex** (шестнадцатеричная).

ж. Убедитесь, что символы 0 – 9, а также A, B, C, D, E и F на цифровой клавиатуре активны в данный момент. В шестнадцатеричной системе счисления (основание 16) используется 16 цифр.

з. Установите переключатель **Dec**. С помощью кнопки мыши щелкните цифру **1**, а затем – цифру **5** на цифровой клавиатуре. В поле введено десятичное число 15. Установите переключатель **Bin**.

и. Что стало с числом 15 в текстовом поле наверху окна?

к. Переключая режимы, числа преобразуются из одной системы счисления в другую. Снова перейдите в режим **Dec**. Число в окне вернется к десятичному виду. Выберите режим **Hex**.

л. Какая шестнадцатеричная цифра (0 – 9 или A – F) соответствует десятичному числу 15?

м. Очистите окно от числа 15. Снова перейдите в режим **Dec**. Для ввода цифр можно использовать не только мышь, но и вспомогательную цифровую клавиатуру и клавиши с цифрами. С помощью вспомогательной цифровой клавиатуры, расположенной справа от клавиши **Enter**, введите число **22**. Если число не появляется в поле калькулятора, нажмите клавишу «**Num Lock**», чтобы включить вспомогательную цифровую клавиатуру. После появления числа 22 в текстовом поле калькулятора с помощью клавиш с цифрами наверху клавиатуры добавьте **0** к числу 22 (после чего в поле должно отображаться 220). Установите переключатель **Bin**.

н. Какое двоичное число соответствует десятичному числу 220?

о. Очистите окно от числа 220. В режиме двоичной системы счисления

введите следующее двоичное число: **11001100**. Установите переключатель **Дес.**

п. Определите, какое десятичное число соответствует двоичному числу 11011100?

р. Преобразуйте следующие десятичные числа в двоичные. Убедитесь, что полученные результаты соответствуют приведенным ниже числам

Десятичное число	Двоичное число
86	1010110
175	10101111
204	11001100
19	10011

с. Преобразуйте следующие двоичные числа в десятичные. Проверьте полученный результат.

Двоичное число	Десятичное число
11000011	195
101010	42
111000	56
10010011	147

Шаг 3: преобразование IP-адресов узлов

а. У сетевых компьютеров обычно есть два адреса: IP-адрес и MAC-адрес Ethernet. Для удобства пользователей IP-адрес обычно представляется в виде группы десятичных чисел, разделенных десятичной точкой, например, 135.15.227.68. Каждый десятичный октет в адресе или маске можно преобразовать в 8 двоичных разрядов. Помните, что компьютер понимает только двоичные разряды. Если все 4 октета преобразовать в двоичную форму, сколько разрядов получится? (IP-адрес является 32-разрядным, 4x8).

б. IP-адреса обычно представляются в виде четырех десятичных чисел, которые принимают значения в пределах от 0 до 255 и разделены точкой. Преобразуйте 4 части IP-адреса 192.168.10.2 к двоичному виду.

Десятичное число	Двоичное число
192	11000000
168	10101000
10	1010
2	10

в. В предыдущей задаче обратите внимание на то, что число 10

преобразуется только в четыре цифры, а 2 – в две цифры. Так как в каждой позиции IP-адреса может быть указано любое число от 0 до 255, то для предоставления каждого такого числа обычно используется восемь цифр. В предыдущем примере для представления чисел 192 и 168 в двоичной форме требовалось восемь цифр, однако для представления чисел 10 и 2 столько цифр не требуется. Обычно к каждому преобразованному к двоичной форме числу IP-адреса слева добавляются нули (0), чтобы получить восемь двоичных цифр. Число 10 должно быть представлено в виде 00001010. Перед четырьмя значащими двоичными цифрами добавлено четыре нуля.

г. В текстовом поле калькулятора в двоичном режиме введите цифры **00001010** и установите переключатель **Дес**.

д. Какому десятичному числу соответствует 00001010?

е. Влияют ли «лидирующие» нули на число?

ж. Как должно быть представлено число 2 (в предыдущем примере) восемью цифрами?

Шаг 4: преобразование масок IP-подсетей узлов

а. Маски подсетей, такие как 255.255.255.0, также представлены в виде десятичных чисел с разделительными точками. Маска подсети всегда состоит из четырех 8-разрядных октетов, каждый из которых представляется десятичным числом. За исключением десятичного числа 0 (все 8 двоичных разрядов – нули) и десятичного числа 255 (все 8 двоичных разрядов – единицы), у каждого октета будет некоторое количество единиц слева и некоторое число нулей справа. Преобразуйте 8 возможных десятичных значений октетов маски подсети в двоичную форму.

Десятичное число	Двоичное число
0	00000000
128	10000000
192	11000000
224	11100000
240	11110000
248	11111000
252	11111100
254	11111110

255	11111111
-----	----------

б. Преобразуйте четыре компонента маски подсети 255.255.255.0 к двоичному виду.

Десятичное число	Двоичное число
255	11111111
255	11111111
255	11111111
0	0 (8 нулей)

Шаг 5: преобразование широковещательных адресов

а. Сетевые компьютеры и устройства используют широковещательные адреса для отправки сообщений целым группам узлов. Преобразуйте следующие широковещательные адреса.

Адрес	Двоичное число
Широковещательный IP-адрес 255.255.255.255	11111111 11111111 11111111 11111111
Широковещательный MAC-адрес FF:FF:FF:FF:FF:FF	11111111 11111111 11111111 11111111 11111111 11111111

Шаг 6: определение числа узлов сети с помощью степеней числа 2

а. Для представления двоичных чисел используются две цифры, 0 и 1. При вычислении количества возможных узлов в подсети используются степени числа 2 из-за двоичного представления. В качестве примера рассмотрим маску подсети, в которой остается шесть разрядов в части IP-адреса, соответствующей узлу. В этом случае число узлов в сети равно 2 в 6-ой степени минус 2 (так как одно число требуется для представления сети, другое – для достижения всех узлов сети, т.е. для широковещательного адреса). Всегда используется число 2, так как работа ведется в двоичной системе счисления. Число 6 определяет количество разрядов, соответствующих узлам.

б. В поле калькулятора в режиме **Dec** введите число 2. Нажмите кнопку «x^y», которая соответствует возведению числа в некоторую степень.

Введите число 6. Нажмите кнопку «=», а затем – клавишу «**Enter**» или «=**=**» на клавиатуре – это эквивалентные способы получения конечного результата. В результате появится число 64. Чтобы вычесть два, нажмите кнопку с

минусом (-), затем – кнопку 2, а в конце – кнопку «=». В результате появится число 62.

Это означает, что может использоваться 62 узла.

в. С помощью описанного выше процесса определите число узлов, если для представления узлов используется следующее число разрядов.

№ разрядов для узлов	Число узлов
5	30
14	16382
24	16777214
10	1022

г. С помощью освоенного метода определите, чему равняется 10 в 4-ой степени.

д. Закройте калькулятор Windows.

Шаг 7: определение номера сети и числа узлов на основе маски подсети

а. Задан сетевой IP-адрес 172.16.203.56 и маска подсети 255.255.248.0. Определите сетевую часть адреса и, на основе оставшегося для узлов числа разрядов, вычислите, сколько можно создать узлов.

б. Начните с преобразования 4 октетов десятичного IP-адреса в двоичную форму, а затем преобразуйте к двоичному виду десятичную маску подсети. При преобразовании к двоичной форме не забудьте добавить лидирующие нули, чтобы получить 8 разрядов для каждого октета.

IP-адрес и маска подсети в десятичной форме	IP-адрес и маска подсети в двоичной форме
172.16.203.56	10101100. 00010000. 11001011. 00111000
255.255.248.0	11111111. 11111111. 11111000. 00000000

в. Выровняйте 32 разряда маски подсети с 32 разрядами IP-адреса и сравните их. Разряды IP-адреса, соответствующие разрядам с единицами в маске подсети, представляют номер сети. Каков двоичный и десятичный номер сети для данного IP-адреса? Сначала определите двоичный адрес (включите все 32 разряда), а затем преобразуйте его к десятичному виду.

Сетевой адрес в двоичной форме: 10101100. 00010000. 11001000.
00000000

Сетевой адрес в десятичной форме: 172.16.200.0

г. Сколько разрядов с единицами в данной маске подсети?

_____ 21

д. Сколько разрядов осталось для создания узлов?

_____ 11

е. Сколько узлов можно создать с оставшимся числом разрядов?

$$2^{11} = 2\,048 - 2 = 2\,046$$

Порядок выполнения работы:

- 1) изучение теоретического материала лабораторной работы;
- 2) разбор исходных примеров, приведенных в данной работе;
- 3) выполнение заданий, предлагаемых в работе.

Задания

1. В сети 172.16.0.0 необходимо выделить подсети так, чтобы к каждой подсети можно было подключить до 600 хостов. Какую маску подсети следует выбрать, чтобы допустить рост числа подсетей в будущем?
2. Сеть 172.16.0.0 содержит 8 подсетей. Вам необходимо подключить к подсети максимально возможное число хостов. Какую маску подсети следует выбрать?
3. В сети 192.168.55.0 необходимо выделить максимальное число подсетей так, чтобы к каждой подсети можно было подключить 25 хостов. Какую маску подсети следует выбрать?
4. Ваша сеть класса А содержит 60 подсетей. В следующие два года вам необходимо организовать еще 40 подсетей, причем так, чтобы к каждой из них можно было подключить максимальное число хостов. Какую маску подсети следует выбрать?
5. У вас имеется сеть класса С с адресом 192.168.19.0, содержащая четыре подсети. Вам необходимо установить максимально возможное число хостов на сегменте. Какую маску подсети следует выбрать?

6. У вас есть сеть класса В, разделенная на 30 подсетей. Вы хотите добавить 25 новых подсетей в ближайшие два года. При этом вам потребуется подключить к каждому сегменту до 600 хостов. Какую Маску подсети следует выбрать?

7. Сеть 192.168.1.0 требуется разделить на 9 подсетей. При этом необходимо подключить к каждому сегменту максимально возможное число хостов. Какую маску подсети следует выбрать?

8. У вас имеется сеть класса С с тремя подсетями. Вам необходимо добавить 2 новые подсети в ближайшие два года. Каждая сеть должна содержать 25 хостов. Какую маску подсети следует выбрать?

9. В имеющемся у вас сетевом адресе класса С 192.168.88.0 необходимо выделить максимально возможное число подсетей, в каждой из которых должно быть до 12 хостов. Какую маску подсети следует выбрать?

10. Вы выбрали маску подсети 255.255.255.248. Сколько подсетей и хостов вы получите?

11. У вас есть IP-адрес 172.16.13.5 и маска подсети 255.255.255.128. Укажите класс адреса, адрес подсети и широковещательный адрес.

12. У вас есть 22-битовая маска подсети. Сколько подсетей и хостов вы получите?

13. Преобразование IP- и MAC-адреса узла

а. Нажмите кнопку «Пуск», выберите команду «Выполнить», введите команду **cmd** и нажмите клавишу «Enter». В командной строке введите **ipconfig /all**.

б. Отметьте IP-адрес и физический адрес (также известный как MAC-адрес).

IP-адрес:

MAC-адрес:

в. С помощью калькулятора преобразуйте четыре числа IP адреса к двоичному виду.

Десятичное число	Двоичное число
------------------	----------------

	Зависит от компьютера
	Зависит от компьютера
	Зависит от компьютера
	Зависит от компьютера

MAC-адрес, или физический адрес, обычно представлен 12 шестнадцатеричными цифрами, сгруппированными в пары и разделенными тире (-). Физические адреса на компьютерах под управлением Windows обычно отображаются в формате xx-xx-xx-xx-xx-xx, где x – цифра 0 – 9 или латинская буква от A до F. Каждую шестнадцатеричную цифру в адресе можно преобразовать в 4-разрядное двоичное число, воспринимаемое компьютером. Если все 12 шестнадцатеричные цифры преобразовать в двоичную форму, сколько разрядов получится? (MAC-адрес является 48 разрядным, 12x4).

д. Преобразуйте каждую шестнадцатеричную пару в двоичную форму. Например, если группа чисел CC-12-DE-4A-BD-88-34 соответствует физическому адресу, приведите шестнадцатеричное число к двоичному виду (11001100). Затем преобразуйте к двоичному виду шестнадцатеричное число 12 (00010010) и так далее. Убедитесь, что добавлено необходимое число лидирующих нулей, чтобы получить 8 двоичных разрядов для каждой пары шестнадцатеричных чисел.

Шестнадцатеричное число	Двоичное число
	Зависит от компьютера
	Зависит от компьютера
	Зависит от компьютера
	Зависит от компьютера
	Зависит от компьютера
	Зависит от компьютера

Варианты заданий представлены в следующей таблице:

Номера вариантов	Номера заданий
1	1,3,5,13
2	2,4,6,13
3	1,7,5,13
4	3,8,9,13

5	2,10,12,13
6	4,7,11,13
7	1,6,12,13
8	8,9,10,13
9	3,6,4,13
10	2,5,10,13
11	4,8,12,13
12	1,9,11,13

Требуется решить задачи согласно Вашему варианту, причем в задачах 1-12 записать результаты решения в двоичной, десятичной и шестнадцатеричной системах счисления.

Содержание отчета и его форма:

Отчет должен иметь форму согласно оформлению простого реферата. Титульный лист должен включать: название дисциплины, название лабораторной работы, фамилию и инициалы сдающего студента, номер группы, фамилию и инициалы принимающего преподавателя. Основная часть работы должна содержать:

- а) ответы на контрольные вопросы, предложенные в лабораторной работе;
- б) подробное рассмотрение предложенных примеров лабораторной работы
- в) описание каждого этапа выполнения задания;
- г) выводы по проделанной работе.

Задание должно быть выполнено в соответствии с приведенными примерами.

Защита работы

Защита работы заключается:

- 1) в выполнении задания и оформлении отчета в соответствии с подпунктом «Содержание отчета и его форма»;
- 2) в ответах на контрольные вопросы по лабораторной работе;
- 3) в ответах на дополнительные вопросы.

Контрольные вопросы

1. Понятие об IP-адресе.
2. Способы представления IP-адресов.
3. Пример иерархической схемы адресации.
4. Характеристика классов сетей.
5. Классы сетей.
6. Что такое подсети? С какой целью они используются?
7. Причины появления подсетей.
8. Выделение подсетей.
9. Основное назначение маски подсети.
10. Маски подсетей, принятые по умолчанию для классов сетей.
11. Связь маски подсети с IP-адресом.
12. Формулы для подсчета числа подсетей и хостов.
13. Примеры определения хостов и подсетей.
14. Использование калькулятора Windows для решения сетевых задач.

Лабораторная работа 5

Моделирование сетей с использованием топологических структур общая шина и звезда

Цель и содержание:

Цель

1. Построение локальной вычислительной сети с использованием топологии общая шина.
2. Построение с локальной вычислительной сети использованием топологии звезда.
3. Снятие статистических характеристик, полученных проектов локальной вычислительной сети и их исследование.

Теоретическая часть

Метод случайного доступа CSMA/CD

В сетях Ethernet используется метод доступа к среде передачи данных, называемый методом коллективного доступа с опознаванием несущей и обнаружением коллизий (CSMA/CD).

Этот метод применяется исключительно в сетях с логической общей шиной (к которым относятся и радиосети, породившие этот метод). Все компьютеры такой сети имеют непосредственный доступ к общей шине, поэтому она может быть использована для передачи данных между любыми двумя узлами сети. Одновременно все компьютеры сети имеют возможность немедленно (с учетом задержки распространения сигнала по физической среде) получить данные, которые любой из компьютеров начал передавать на общую шину (рис. 3). Простота схемы подключения – это один из факторов, определивших успех стандарта Ethernet. Говорят, что кабель, к которому подключены все станции, работает в режиме коллективного доступа.



Этапы доступа к среде

Все данные, передаваемые по сети, помещаются в кадры определенной структуры и снабжаются уникальным адресом станции назначения.

Чтобы получить возможность передавать кадр, станция должна убедиться, что разделяемая среда свободна. Это достигается прослушиванием основной гармоник сигнала, которая также называется несущей частотой. Признаком незанятости среды является отсутствие на ней несущей частоты, которая при манчестерском способе кодирования равна 5-10 МГц, в зависимости от последовательности единиц и нулей, передаваемых в данный момент.

Если среда свободна, то узел имеет право начать передачу кадра. Этот кадр изображен на рис. 3 первым. Узел 1 обнаружил, что среда свободна, и начал передавать свой кадр. В классической сети Ethernet на коаксиальном кабеле сигналы передатчика узла 1 распространяются в обе стороны, так что все узлы сети их получают. Кадр данных всегда сопровождается *преамбулой*, которая состоит из 7 байт, состоящих из значений 10101010, и 8-го байта, равного 10101011 (начальный ограничитель кадра). Преамбула нужна для вхождения приемника в побитовый и побайтовый синхронизм с передатчиком.

Все станции, подключенные к кабелю, могут распознать факт передачи кадра, и та станция, которая узнает собственный адрес в заголовках кадра, записывает его содержимое в свой внутренний буфер, обрабатывает полученные данные, передает их вверх по своему стеку, а затем посылает по кабелю кадр-ответ. Адрес станции-источника содержится в исходном кадре, поэтому станция-получатель знает, кому нужно послать ответ.

Узел 2 во время передачи кадра узлом 1 также пытался начать передачу своего кадра, однако обнаружил, что среда занята - на ней присутствует несущая частота, - поэтому узел 2 вынужден ждать, пока узел 1 не прекратит передачу кадра.

После окончания передачи кадра все узлы сети обязаны выдержать технологическую паузу в 9,6 мкс. Эта пауза, называемая межкадровым интервалом, нужна для приведения сетевых адаптеров в исходное состояние, а также для предотвращения монопольного захвата среды одной станцией. После окончания технологической паузы узлы имеют право начать передачу своего кадра, так как среда свободна. Из-за задержек распространения сигнала по кабелю не все узлы строго одновременно фиксируют факт окончания передачи кадра узлом 1.

В приведенном примере узел 2 дождался окончания передачи кадра узлом 1, сделал паузу в 9,6 мкс и начал передачу своего кадра.

Возникновение коллизии

При описанном подходе возможна ситуация, когда две станции одновременно пытаются передать кадр данных по общей среде. Механизм прослушивания среды и пауза между кадрами не гарантируют от возникновения такой ситуации, когда две или более станции одновременно решают, что среда свободна, и начинают передавать свои кадры. Говорят, что при этом происходит **коллизия**, так как содержимое обоих кадров сталкивается на общем кабеле и происходит искажение информации - методы кодирования, используемые в Ethernet, не позволяют выделять сигналы каждой станции из общего сигнала.

Коллизия - это нормальная ситуация в работе сетей Ethernet. В примере, изображенном на рис. 3, коллизию породила одновременная передача данных узлами 3 и 1. Для возникновения коллизии не обязательно, чтобы несколько станций начали передачу абсолютно одновременно, такая ситуация маловероятна. Гораздо вероятней, что коллизия возникает из-за того, что один узел начинает передачу раньше другого, но до второго узла сигналы первого просто не успевают дойти к тому времени, когда второй узел решает начать передачу своего кадра. То есть коллизии - это следствие распределенного характера сети.

Чтобы корректно обработать коллизию, все станции одновременно наблюдают за возникающими на кабеле сигналами. Если передаваемые и наблюдаемые сигналы отличаются, то фиксируется **обнаружение коллизии**. Для увеличения вероятности скорейшего обнаружения

коллизии всеми станциями сети, станция, которая обнаружила коллизию, прерывает передачу своего кадра (в произвольном месте, возможно, и не на границе байта) и усиливает ситуацию коллизии посылкой в сеть специальной последовательности из 32 бит, называемой **jam-последовательностью**.

После этого обнаружившая коллизию передающая станция обязана прекратить передачу и сделать паузу в течение короткого случайного интервала времени. Затем она может снова предпринять попытку захвата среды и передачи кадра. Случайная пауза выбирается по следующему алгоритму:

Пауза = $L \cdot$ (интервал отсрочки),

где интервал отсрочки равен 512 битовым интервалам (в технологии Ethernet принято все интервалы измерять в битовых интервалах; битовый интервал обозначается как bt и соответствует времени между появлением двух последовательных бит данных на кабеле; для скорости 10 Мбит/с величина битового интервала равна 0,1 мкс или 100 нс);

L представляет собой целое число, выбранное с равной вероятностью из диапазона $[0, 2^N]$, где N — номер повторной попытки передачи данного кадра: 1, 2, ..., 10.

После 10-й попытки интервал, из которого выбирается пауза, не увеличивается. Таким образом, случайная пауза может принимать значения от 0 до 52,4 мс.

Если 16 последовательных попыток передачи кадра вызывают коллизию, то передатчик должен прекратить попытки и отбросить этот кадр.

Из описания метода доступа видно, что он носит вероятностный характер, и вероятность успешного получения в свое распоряжение общей среды зависит от загруженности сети, то есть от интенсивности возникновения в станциях потребности в передаче кадров. При разработке этого метода в конце 70-х годов предполагалось, что скорость передачи данных в 10 Мбит/с очень высокая, по сравнению с потребностями компьютеров во взаимном обмене данными, поэтому загрузка сети будет всегда небольшой. Это предположение остается иногда справедливым и по сей день, однако уже появились приложения, работающие в реальном масштабе времени с мультимедийной информацией, которые очень загружают сегменты Ethernet. При этом коллизии возникают гораздо чаще. При значительной интенсивности коллизий полезная пропускная способность сети Ethernet резко падает, так как сеть почти постоянно занята повторными попытками передачи кадров. Для уменьшения интенсивности возникновения коллизий нужно либо уменьшить трафик, сократив, например, количество узлов в сегменте или заменив приложения, либо повысить скорость протокола, например, перейти на Fast Ethernet. Следует отметить, что метод доступа CSMA/CD вообще не гарантирует станции, что она когда-либо сможет получить доступ к среде. При небольшой нагрузке сети вероятность такого события невелика, но при коэффициенте использования сети, приближающемся к 1, такое событие становится очень вероятным. Этот недостаток метода случайного доступа - плата за его чрезвычайную простоту, которая сделала технологию Ethernet самой недорогой. Другие методы доступа - маркерный доступ сетей Token Ring и FDDI, метод Demand Priority сетей 100VG-AnyLAN свободны от этого недостатка.

Время двойного оборота и распознавание коллизий

Четкое распознавание коллизий всеми станциями сети является необходимым условием корректной работы сети Ethernet. Если какая-либо передающая станция не опознает коллизию и решит, что кадр данных ею передан верно, то этот кадр данных будет утерян. Из-за наложения сигналов при коллизии информация кадра исказится, и он будет отбракован принимающей станцией (возможно, из-за несовпадения контрольной суммы). Скорее всего, искаженная информация будет повторно передана каким-либо протоколом верхнего уровня, например, транспортным или прикладным, работающим с установлением соединения. Но повторная передача сообщения протоколами верхних уровней произойдет через значительно более длительный интервал времени (иногда даже через несколько секунд) по сравнению с микросекундными интервалами, которыми оперирует протокол Ethernet. Поэтому если коллизии

не будут надежно распознаваться узлами сети Ethernet, то это придет к заметному снижению полезной пропускной способности данной сети.

Для надежного распознавания коллизий должно выполняться следующее соотношение:

$$T_{\min} \geq \text{PDU},$$

Где $T_{\min}$ — время передачи кадра минимальной длины, а PDU, — время, за которое сигнал коллизии успевает распространиться до самого дальнего узла сети. Так как в худшем случае сигнал должен пройти дважды между наиболее удаленными друг от друга станциями сети (в одну сторону проходит неискаженный сигнал, а на обратном пути распространяется уже искаженный коллизией сигнал), то это время называется временем двойного оборота.

При выполнении этого условия передающая станция должна успевать обнаружить коллизию, которую вызвал переданный ее кадр, еще до того, как она закончит передачу этого кадра.

Очевидно, что выполнение этого условия зависит, с одной стороны, от длины минимального кадра и пропускной способности сети, а с другой стороны, от длины кабельной системы сети и скорости распространения сигнала в кабеле (для разных типов кабеля эта скорость несколько отличается).

Все параметры протокола Ethernet подобраны таким образом, чтобы при нормальной работе узлов сети коллизии всегда четко распознавались. При выборе параметров учитывается и приведенное выше соотношение, связывающее между собой минимальную длину кадра и максимальное расстояние между станциями в сегменте сети.

В стандарте Ethernet принято, что **минимальная длина поля данных кадра составляет 46 байт** (что вместе со служебными полями дает минимальную длину кадра 64 байт, а вместе с преамбулой - 72 байт или 576 бит). Отсюда может быть определено ограничение на расстояние между станциями.

В 10-мегабитном Ethernet время передачи кадра минимальной длины равно 575 битовых интервалов, следовательно, время двойного оборота должно быть меньше 57,5 мкс. Расстояние, которое сигнал может пройти за это время, зависит от типа кабеля. Для толстого коаксиального кабеля расстояние равно примерно 13 280 м. Учитывая, что за это время сигнал должен пройти по линии связи дважды, расстояние между двумя узлами не должно быть больше 6 635 м. В стандарте величина этого расстояния выбрана существенно меньше, с учетом других, более строгих ограничений.

Одно из таких ограничений связано с предельно допустимым затуханием сигнала. Для обеспечения необходимой мощности сигнала при его прохождении между наиболее удаленными друг от друга станциями сегмента кабеля, максимальная длина непрерывного сегмента толстого коаксиального кабеля, с учетом вносимого им затухания, выбрана в 500 м. На кабеле в 500 м, условия распознавания коллизий будут выполняться с большим запасом для кадров любой стандартной длины, в том числе и 72 байт (время двойного оборота по кабелю 500 м составляет всего 43,3 битовых интервала). Поэтому минимальная длина кадра могла бы быть установлена еще меньше. Однако разработчики технологии не стали уменьшать минимальную длину кадра, имея в виду многосегментные сети, которые строятся из нескольких сегментов, соединенных повторителями.

Повторители увеличивают мощность передаваемых с сегмента на сегмент сигналов, в результате затухание сигналов уменьшается и можно использовать сеть гораздо большей длины, состоящую из нескольких сегментов. В коаксиальных реализациях Ethernet разработчики ограничили максимальное количество сегментов в сети пятью, что в свою очередь ограничивает общую длину сети 2500 метрами. Даже в такой многосегментной сети условие обнаружения коллизий по-прежнему выполняется с большим запасом (сравним полученное из условия допустимого затухания расстояние в 2500 м с вычисленным выше максимально возможным по времени распространения сигнала расстоянием 6635 м). Однако в действительности временной запас является существенно меньше, поскольку в многосегментных сетях сами повторители вносят в распространение

сигнала дополнительную задержку в несколько десятков битовых интервалов. Небольшой запас был сделан также для компенсации отклонений параметров кабеля и повторителей.

В результате учета всех этих и некоторых других факторов было тщательно подобрано соотношение между минимальной длиной кадра и максимально возможным расстоянием между станциями сети, которое обеспечивает надежное распознавание коллизий. Это расстояние называют также максимальным диаметром сети.

С увеличением скорости передачи кадров, что имеет место в новых стандартах, максимальное расстояние между станциями сети уменьшается пропорционально увеличению скорости передачи. В стандарте Fast Ethernet оно составляет около 210 м, в стандарте Gigabit Ethernet оно было бы ограничено 25 метрами, если бы разработчики стандарта не предприняли некоторых мер по увеличению минимального размера пакета.

В табл. 1 приведены значения основных параметров процедуры передачи кадра стандарта 802.3, которые не зависят от реализации физической среды. Важно отметить, что каждый вариант физической среды технологии Ethernet добавляет к этим ограничениям свои, часто более строгие ограничения, которые также должны выполняться.

Таблица 1. Параметры уровня MAC Ethernet

Параметры	Значения
Битовая скорость	10 Мбит/с
Интервал отсрочки	512 битовых интервала
Межкадровый интервал	9,6 мкс
Максимальное число попыток передачи	16
Максимальное число возрастания диапазона паузы	10
Длина jam-последовательности	32 бита
Максимальная длина кадра (без преамбулы)	1518 байт
Минимальная длина кадра (без преамбулы)	64 байт (512 бит)
Длина преамбулы	64 бит
Минимальная длина случайной паузы после коллизии	0 битовых инт.
Максимальная длина случайной паузы после коллизии	524000 бит. инт
Максимальное расстояние между станциями сети	2500 м
Максимальное число станций в сети	1024

Методика и порядок выполнения работы

Методика

1. Построение локальной вычислительной сети с использованием топологии общая шина.

Для этого необходимо выполнить следующие действия:

1. Запустите NetCracker Professional выбав Programs==> NetCracker Professional 3.2 ==> NetCracker Professional из меню «Пуск».
2. Из меню Файл выберите пункт Новый.
3. В браузере устройств найдите пункт «GenericLANs».

4. Переместите требуемые устройства в рабочую область, следуя следующим инструкциям:
 - а. Выберите Thin Ethernet Segment в подокне образов и переместите в рабочую область.
 - б. Измените размеры изображения устройства в рабочей области по вашему усмотрению.
 - с. Увеличьте размер шрифта подписи к устройству, выбрав из контекстного меню пункт «Свойства».
 5. Добавьте четыре рабочие станции в окно рабочей области.
 - а. Найдите пункт «LAN workstations», затем «Workstations» и выберите папку «Generic Devices».
 - с. Вытащите Ethernet Workstation.
- Получится следующее (рисунок 2.1):

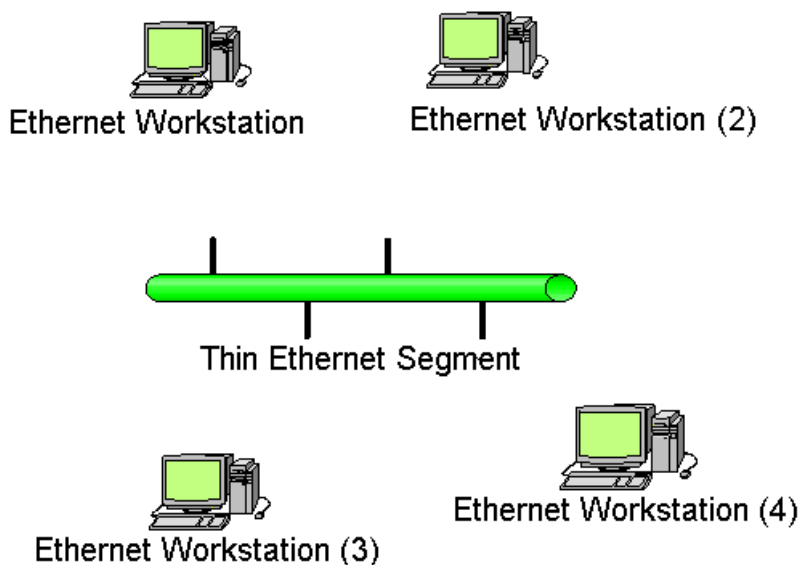


Рисунок 2.1 – Внешний вид рабочих станций

Способ определения совместимых устройств:

1. Щелкните мышью на устройстве.
2. Выберите пункт «Найти совместимые» (Find Compatible) из меню «Объекты» (Objects).
7. Соединение шины и рабочих станций.
 - а. В панели режимов нажмите кнопку .
 - б. Нажмите левой кнопкой мыши сначала на Ethernet Workstation, а потом на Thin Ethernet Segment. Появится такой диалог (рисунок 2.2):

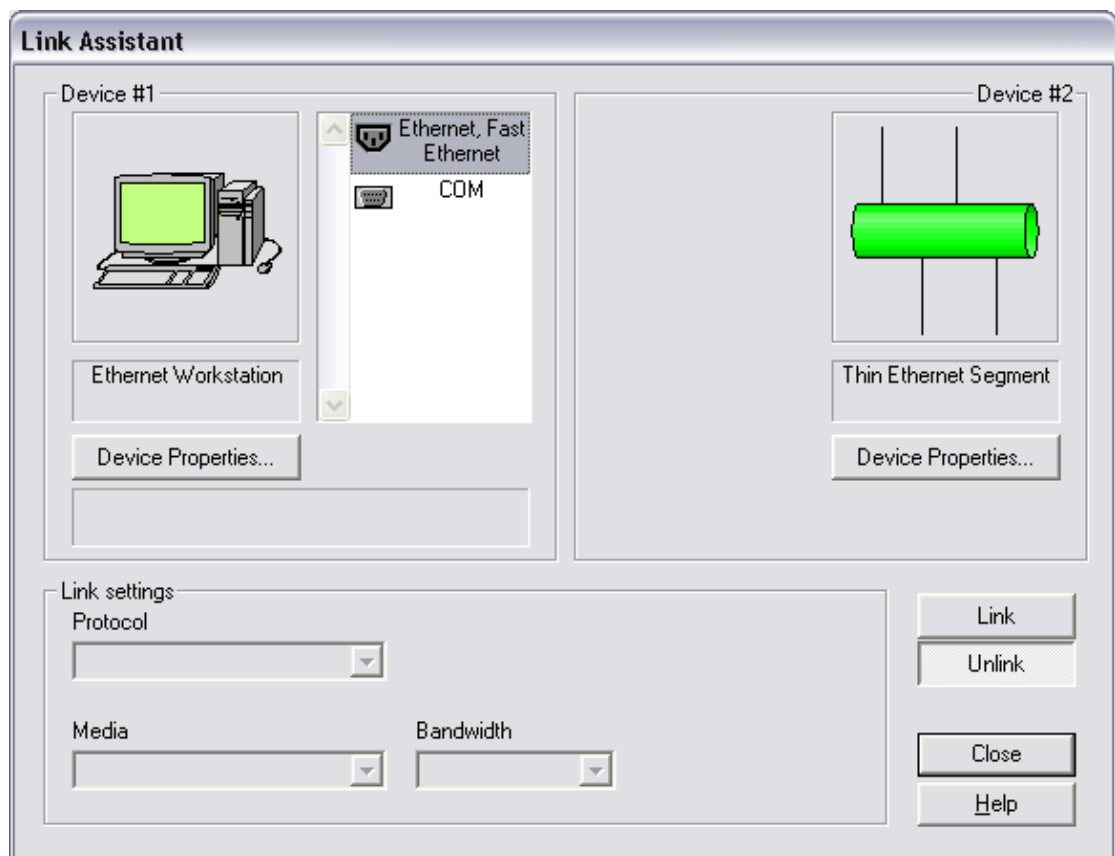


Рисунок 2.2 – Внешний вид панели Link Assistant

- с. Нажмите кнопки «Link», затем «Close», чтобы установить связь и закрыть окно.
- d. Для быстрой установки связи держите клавишу Shift, нажмите мышью на Ethernet Workstation, а потом на шину. Появится связь без обращения к диалоговому окну.
8. Выбор типа среды передачи.
 - a. Из меню «View» – «Вид» выберите пункт «Легенды». Черный цвет связи в вашем проекте означает коаксиальную линию.
 - b. Закройте диалог кнопкой «Close».
9. Выбор типа трафика между рабочими станциями.
 - a. Нажмите кнопку «Set Traffics» .
 - b. Нажмите левой кнопкой мыши на Ethernet Workstation, затем на Ethernet Workstation (4). Появится такой диалог (рисунок 2.3):

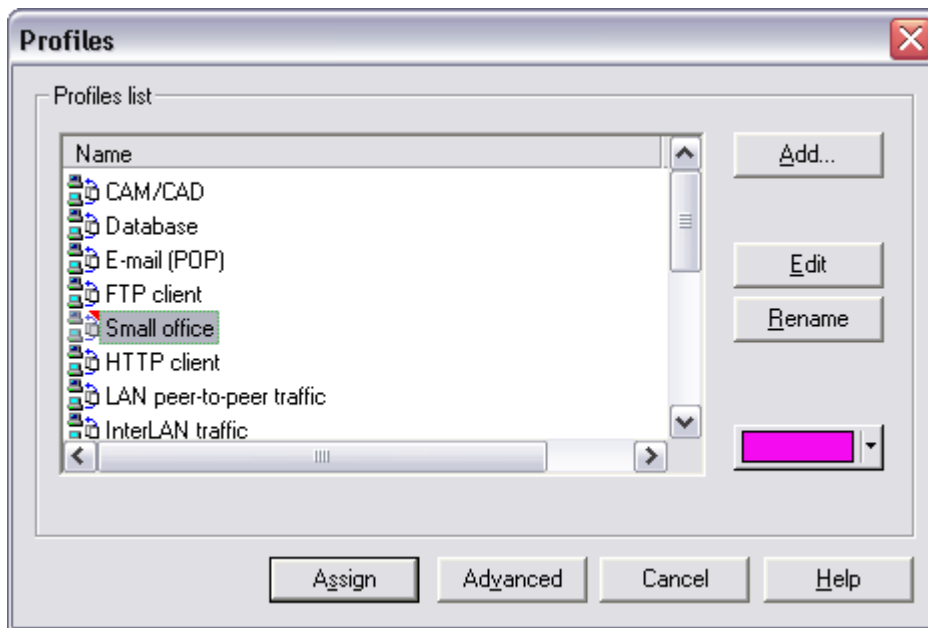


Рисунок 2.3 – Внешний вид панели Profiles

- с. Выберите тип трафика «Small Office» между рабочими станциями.
 - д. Нажмите кнопку «Assign», чтобы установить данный тип трафика.
 - е. Повторите шаги 9b-d, но теперь нажмите левой клавишей мыши вначале на Ethernet Workstation (4), а затем на Ethernet Workstation.
 - ф. Повторите шаги 9b-e для Ethernet Workstation (2) и Ethernet Workstation (3).
 10. Проверка работоспособности созданного проекта сети. Нажмите на управляющей панели кнопку «Start». Через сегмент побегут пакеты между рабочими станциями.
 11. Регулирование интенсивности взаимодействия информационного обмена.
 - а. Нажмите кнопку «Animation Setup» , и в появившемся диалоге переместите регулятор «Packet intensity».
 - б. Нажмите кнопку ОК. Через несколько секунд интенсивность трафика должна измениться.
 12. Изменение скорости перемещения пакетов.
 - а. Нажмите кнопку «Animation Setup» , и в появившемся диалоге переместите регулятор «Packet speed».
 - б. Нажмите кнопку ОК. Через несколько секунд скорость пакетов должна измениться.
 13. Изменение размеров изображений пакетов.
 - а. Нажмите кнопку «Animation Setup» , и в появившемся диалоге переместите регулятор «Packet size».
 - б. Нажмите кнопку ОК. Размер изображений пакетов должен измениться.
 14. Посмотрите на использованные Вами устройства и выберите закладку «Recently used» – «Используемые» в подокне образов.
 15. Расположение карты как фона рабочего окна.
 - а. Из контекстного меню рабочего окна выберите «Site Setup» – «Настройки окна».
 - б. На закладке «Background» поставьте галочку «Map».
- С помощью кнопки «Browse» выберите файл карты.
- Чтобы понять, о чём речь, посмотрите на эту панель (рисунок 2.4):

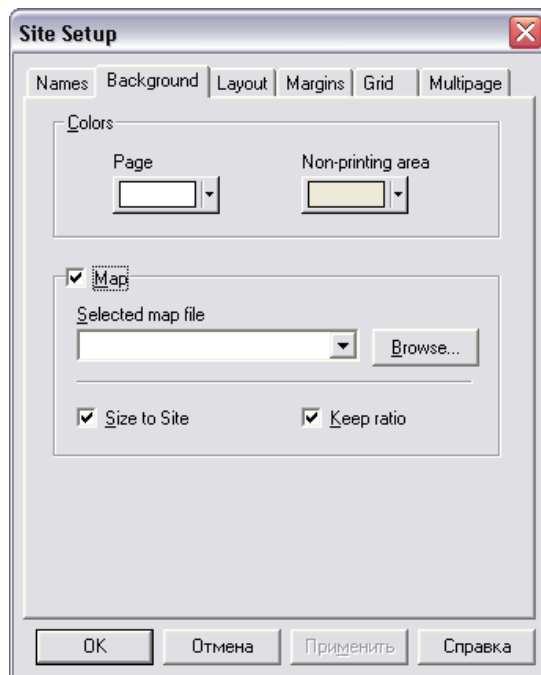


Рисунок 2.4 – Внешний вид панели Site Setup

16. Раскраска фона. Из контекстного меню рабочего окна выберите «Site Setup» – «Настройки окна».

- a. На закладке «Background» снимите галочку «Map».
- b. В списке «Page» выберите цвет по вкусу.
- c. В списке «Non-printing» выберите другой цвет.
- d. Нажмите ОК.

Можно добавить цвет фона и в окно, содержащее уже изображение карты.

17. Профили трафика. Из меню «Global» – «Все», выберите пункт «Data flow» – «Потоки данных» (см. рисунок 2.5).

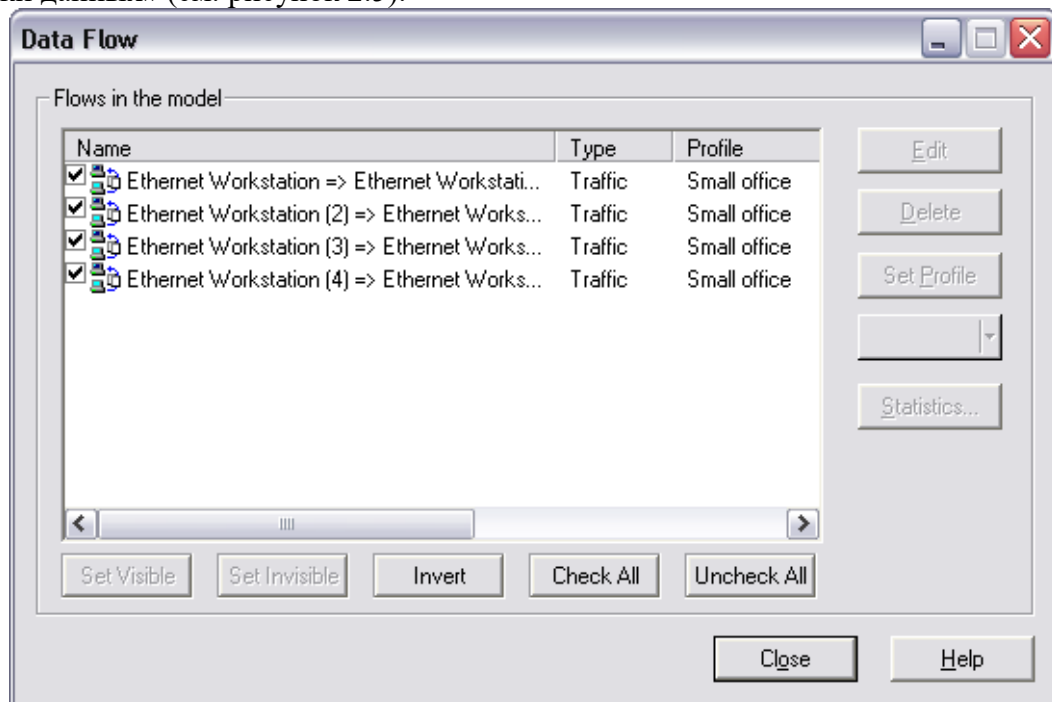


Рисунок 2.5 – Внешний вид панели Data Flow

В появившемся диалоге отобразятся профили уже выбранных Вами типов трафика.

18. Увеличение сети.

а. Добавьте в рабочее окно еще один сегмент Thin Ethernet Segment и еще несколько рабочих станций Ethernet Workstation. Соедините их в сеть повторив шаги 9b-e для новых рабочих станций.

б. Попробуйте создать соединение между двумя сегментами Thin Ethernet Segment и Thin Ethernet Segment (2). В результате вы получите следующее сообщение (см. рисунок 2.6)

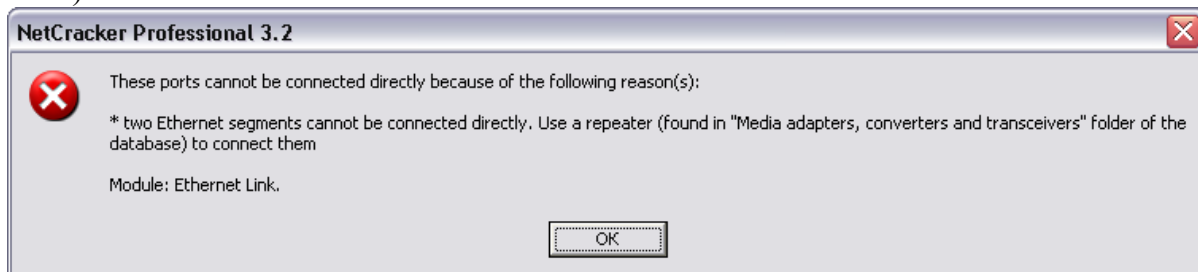


Рисунок 2.6 – Внешний вид окна при попытке соединения двух сегментов напрямую. Для соединения двух сегментов в одну сеть вам необходимо поставить между ними адаптер.

с. Выберите стандартный курсор кнопкой «Standard» в панели режимов.

д. В браузере устройств найдите список «Media adapters and converters and transceivers», затем «Generic Devices».

е. В подокне образов выберите «Ethernet Media Converter», переместите на свободное место в рабочем окне.

ф. Создайте соединение между Thin Ethernet Segment и Ethernet Media Converter. Повторите процесс соединения для Ethernet Media Converter и Thin Ethernet Segment (2).

г. Теперь вы можете назначить трафик к примеру для Ethernet Workstation и Ethernet Workstation (6).

h. Запустите проект и проверьте его работоспособность.

2. Построение с локальной вычислительной сети использованием топологии звезда.

1) Из меню Файл выберите пункт Новый.

2) В браузере устройств найдите пункт «Switches». Затем разверните пункты Workgroup, Ethernet, D-Link Systems.

а) Переместите в рабочую область DES-2212 Fast Ethernet/Ethernet 12-port Switch выбрав его в подокне образов.

б) Откройте свойства данного свича, закладку Ports, на основе этого свича можно создать сеть из 8 рабочих станций с использованием сетевых карт Ethernet и 4 соединения при использовании адаптеров Fast Ethernet.

3) Добавьте несколько рабочих станции в окно рабочей области.

а. Найдите пункт «LAN workstations», затем «PCs» и выберите папку «Generic Devices».

с. Вытащите PC. Должно получиться следующее (рисунок 2.7):

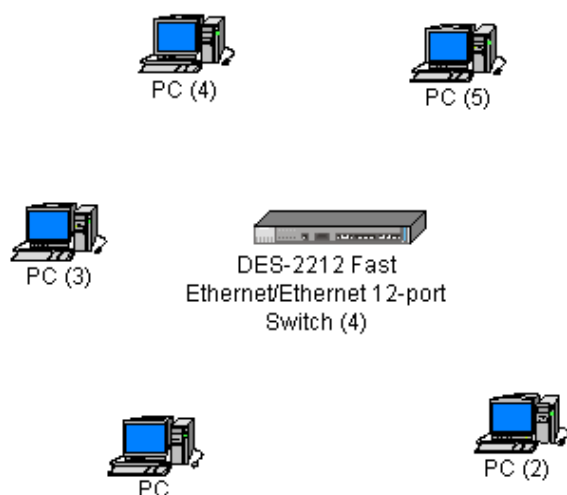


Рисунок 2.7 – Внешний вид рабочих станций

- 4) Добавьте сетевые адаптеры в рабочие станции.
 - а. Соберите пункт «LAN workstation».
 - б. Найдите «LAN adapters», «Ethernet», «Generic Devices».
 - с. Добавьте по одному Ethernet adapter в рабочие станции PC, PC(2), PC(3). И по одному Fast Ethernet adapter в PC(4), PC(5).
 - д. Создайте соединение между свитчем и станциями. При подключении обратите внимание на совместимость между различными портами и сетевыми картами.
 - е. Назначьте трафик между рабочими станциями как это было описано в вопросе 1, пункт 9.
- 5) Добавьте в рабочую область еще один свитч. Выберите DES-5024 Fast Ethernet/Ethernet 24-port Modular Switch из «Switches», Workgroup, Ethernet, D-Link Systems.
 - а) Перенесите еще несколько рабочих станций, добавьте к ним адаптеры и назначьте трафик, повторив действия пунктов 3 и 4.

Полученная сеть будет иметь топологию звезда состоящую из двух сегментов.

3. Снятие статистических характеристик, полученных проектов локальной вычислительной сети и их исследование.

- а. Правой кнопкой мыши кликните на любой из линий соединения и в контекстном меню выберите пункт Statistics. В результате чего появится окно (см. рисунок 2.8)

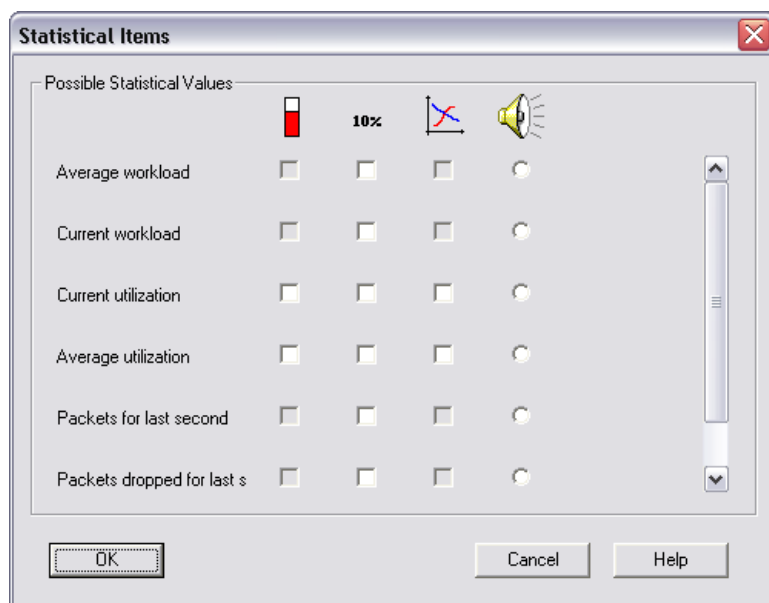


Рисунок 2.8 – Внешний вид окна статистики

- b. Поставьте галочки, например, напротив Current utilization в первом столбце и напротив Average utilization во втором. Нажмите Ок.
- c. Теперь вы можете увидеть статистику выбранной вами линии (см. рисунок 2.9)

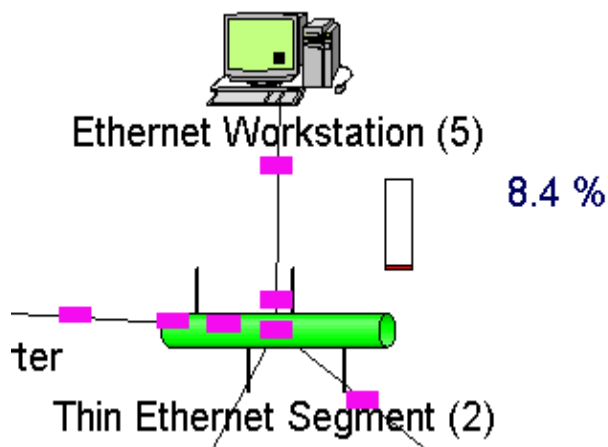


Рисунок 2.9 – Статистика

- d. Попробуйте поменять интенсивность передачи пакетов и посмотреть, что получится.

В NetCracker также возможно создание графиков.

- a. Правой кнопкой мыши кликните на любой из линий соединения и в контекстном меню выберите пункт Statistics.

- b. Поставьте галочку, например, напротив Average utilization в третьем столбце. Нажмите Ок.

- c. На экране график отражающий работу выбранной линии (см. рисунок 2.10).

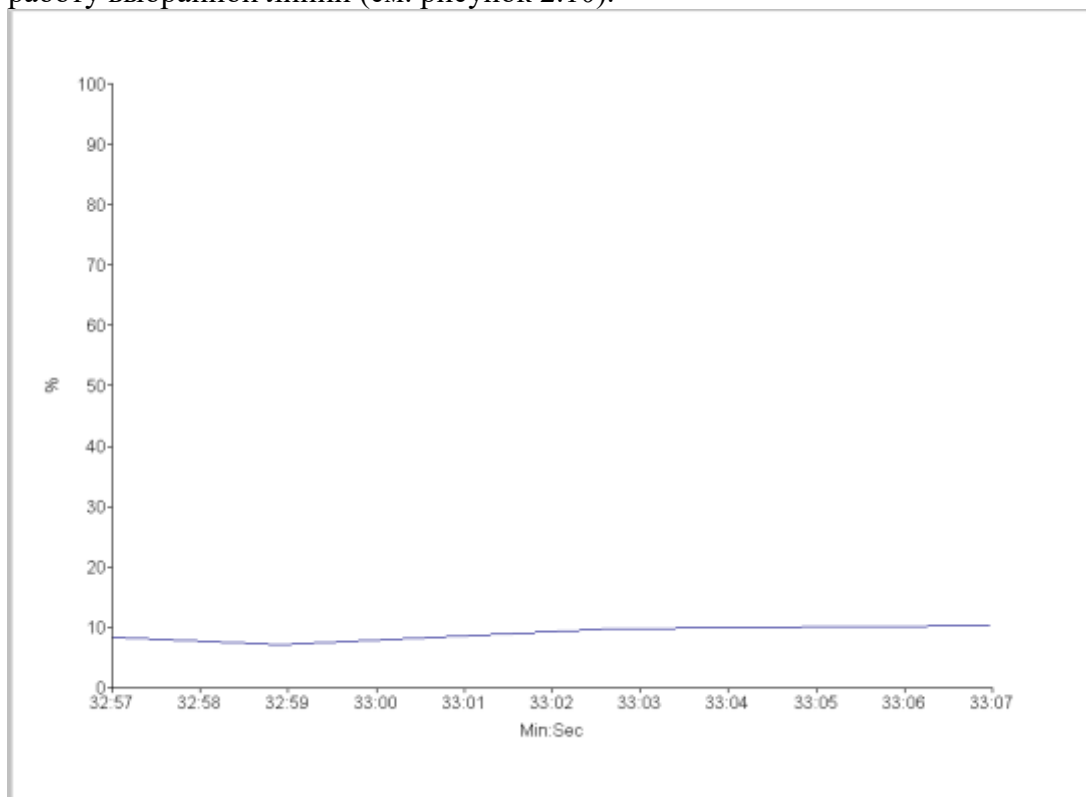


Рисунок 2.10 – График статистики

- a. Из меню Файл выберите команду Сохранить.
- b. По умолчанию имя файл будет Net1.net. Вы можете изменить его на такое, какое считаете необходимым. Расширение .NET добавится автоматически.

- c. Закройте проект, предварительно остановив его.
- d. Постройте одноуровневый сетевой проект согласно Вашему варианту задания, руководствуясь правилами, изложенными выше.

Варианты заданий:

- 1. Построить сетевой проект ЛВС Ethernet, имеющий топологию общая шина.
- 2. Построить сетевой проект ЛВС Ethernet, имеющий топологию звезда.
- 3. Построить сетевой проект ЛВС Ethernet, имеющий топологию иерархическая звезда.
- 4. Снятие статистических характеристик, полученных проектов локальной вычислительной сети и их исследование.

Порядок выполнения работы:

- 1) изучение теоретического материала лабораторной работы;
- 2) разбор исходных примеров, приведенных в данной работе;
- 3) выполнение задания, предлагаемого в лабораторной работе.

Контрольные вопросы:

- 1. В каких топологических структурах используется метод доступа к среде передачи данных CSMA/CD?
- 2. Назовите этапы доступа метода CSMA/CD к среде передачи данных.
- 3. Назовите назначение преамбулы и ее содержание.
- 4. Для чего необходим межкадровый интервал и какое его значение?
- 5. Что такое коллизия? Охарактеризуйте ее.
- 6. Для чего необходима jam-последовательность?
- 7. Как формируется случайная пауза?
- 8. Как осуществляется распознавание коллизий?
- 9. Назовите минимальную длину поля данных.
- 10. Назовите параметры уровня MAC Ethernet.

Лабораторная работа 9

Исследование коммуникационного оборудования Switch 2960

Цель и содержание:

Цель

Изучить основные принципы конфигурирования коммуникационного оборудования Switch 2960.

Содержание

1. Регистрация на Switch.
2. Установка даты и времени на коммутатор
3. Установка паролей.
4. Изменение имени в устройстве Switch 2960
5. Установка IP-адреса
6. Конфигурирование портов

Теоретическое обоснование:

Switch – сетевое устройство, осуществляющее фильтрацию, распространение и создание потока кадров по результатам анализа адреса получателя. Функционирует на канальном уровне модели OSI.

Методика и порядок выполнения работы:

Методика

Регистрация на Switch

При регистрации на устройстве Switch 2960 вы видите приглашение интерпретатора Eexec в пользовательском режиме, появляется информация об устройстве и управляющее меню:

Catalyst 2960 Management Console
Copyright (c) Cisco Systems, Inc. 1993-1999
All rights reserved.
Enterprise Edition Software
Ethernet Address: 00-30-80-C7-BE-C0

PCA Number: 73-3122-04
PCA Serial Number: FAB033723WJ
Model Number: WS-C1912-A
System Serial Number: FAB0338S10A
Power Supply S/N: APQ032404SA
PCB Serial Number: FAB033723WJ,73-3122-04

1 user(s) now active on Management Console.

User Interface Menu

[M] Menus

[K] Command Line

Enter Selection:

Menus(Меню) – этот пункт позволяет производить выбор пошаговой конфигурации. При выборе клавиши [M] появляется дополнительное меню:

Catalyst 2960 - Main Menu

[C] Console Settings

[S] System
 [N] Network Management
 [P] Port Configuration
 [A] Port Addressing
 [D] Port Statistics Detail
 [M] Monitoring
 [V] Virtual LAN
 [R] Multicast Registration
 [F] Firmware
 [I] RS-232 Interface
 [U] Usage Summaries
 [H] Help
 [K] Command Line
 [X] Exit Management Console
 Enter Selection: (Произведите выбор)

Рассмотрим второй пункт управляющего меню «Command Line (Командная строка)». Он позволяет производить конфигурацию устройства с помощью управляющих команд, список которых несложно получить. Для начала необходимо выбрать клавишу [K]. После выполненной операции появится приглашение интерпретатора Eхес о том, что вы находитесь в пользовательском режиме (user mode).

Enter Selection:
 CLI session with the switch is open.
 To end the CLI session, enter [Exit].

>

Для того чтобы узнать, какие команды вы можете выполнять необходимо, произвести следующую операцию:

>?

Eхес commands: Расшифровка каждой из команд представлена в таблице 2.1.

Таблица 2.1 – Команды пользовательского режима

Название команды	Расшифровка команды
Enable	Turn on privileged commands (Разрешить применение команд привилегированного режима)
Exit	Exit from the EXEC (Завершить работу интерпретатора Eхес)
Help	Description of the interactive help system (Описание диалоговой справочной системы)
Ping	Send echo messages (Послать эхо-сообщение)
Show	Show running system information (Вывести загрузочную информацию о системе)
Session	Tunnel to module (Модуль тунеля)
Terminal	Set terminal line parameters (Установка параметров терминала)

На данном этапе можно выполнять операции не глобального масштаба, что характеризует таблица 2.1.

Для того чтобы произвести глобальную конфигурацию необходимо выполнить команду «enable» и перейти в привилегированный режим (privileged mode).

>enable

#

Также как и в пользовательском режиме, список допустимых команд и получения любой справочной информации, необходимо произвести ввод «?».

#?

Exec commands:
clear Reset functions
configure Enter configuration mode
copy Copy configuration or firmware
delete Reset configuration
disable Turn off privileged commands
enable Turn on privileged commands
exit Exit from the EXEC
help Description of the interactive help system
menu Enter menu interface
ping Send echo messages
reload Halt and perform warm start
session Tunnel to module
show Show running system information
terminal Set terminal line parameters
vlan-membership VLAN membership configuration

Установка даты и времени на коммутатор

Для этого в привилегированном режиме используем следующую команду:

```
Switch#clock set 12:29:30 20 October 2011
```

Для просмотра установленной даты используем команду

```
Switch #SH CLOCK
```

```
11:35:49.985 word ?? ??? 20 2011
```

Установка паролей

Для установки паролей на управляющую консоль и входа в привилегированный режим необходимо производить следующим образом:

```
#config t
```

Enter configuration commands, one per line. End with CNTL/Z.

```
(config)#enable password level 1 kmlp
```

```
(config)#enable password level 15 mklp
```

```
(config)#^Z
```

```
%SYS-5-CONFIG_I: Configured from console by console
```

```
#
```

Для установки пароля разрешенный секрет необходимо выполнить следующие команды:

```
(config)#enable secret mklp9
```

```
(config)#^Z
```

```
%SYS-5-CONFIG_I: Configured from console by console
```

```
#disable
```

```
>en
```

```
Enter password:*****
```

```
#
```

Изменение имени в устройстве Switch 2960

Для изменения имени устройства необходимо выполнить в режиме конфигурации «configure terminal» команду «hostname» и ввести другое имя:

```
#configure terminal
```

Enter configuration commands, one per line. End with CNTL/Z.

```
(config)#hostname Switch_My
```

```
Switch_My(config)#^Z
```

```
%SYS-5-CONFIG_I: Configured from console by console
```

```
Switch_My#
```

Замечание: при выполнении какой-либо операции все изменения в устройстве сохраняются автоматически.

Установка IP-адреса

Для установки IP-адреса и шлюза в режиме конфигурации выполняем следующую команду:

```
#conf t
```

Enter configuration commands, one per line. End with CNTL/Z.

```
(config)#hostname MySwitch_1
```

```
MySwitch_1(config)#ip host 172.16.10.16 255.255.255.0
```

```
MySwitch_1(config)#ip default-gateway 172.16.10.1
```

```
MySwitch_1(config)#^Z
```

Чтобы посмотреть информацию о настройках свитча используем команду `show running-config`:

```
Switch_My #sh running-config
```

```
Building configuration...
```

```
Current configuration : 1108 bytes
```

```
!
```

```
version 12.1
```

```
no service timestamps log datetime msec
```

```
no service timestamps debug datetime msec
```

```
no service password-encryption
```

```
!
```

```
hostname Switch_My
```

```
!
```

```
enable secret 5 $1$mERr$H7PDxl7VYMqaD3id4jJVK/
```

```
!
```

```
clock timezone word 23 23
```

```
!
```

```
ip host 172.16.10.16 255.255.255.0
```

```
!
```

```
!
```

```
interface FastEthernet0/1
```

```
!
```

```
interface FastEthernet0/2
```

```
!
```

```
interface FastEthernet0/3
```

```
!
```

```
interface FastEthernet0/4
```

```
!
```

```
interface FastEthernet0/5
```

```
!
```

```
interface Vlan1
```

```
no ip address
```

```
shutdown
```

```
!
```

```
ip default-gateway 172.16.10.1
```

```
!
```

```
!
```

```
line con 0
```

```
!
```

```
line vty 0 4
```

```
password 123
login
line vty 5 15
login
!
!
End
```

Конфигурирование портов

Для выбора конфигурирования портов необходимо выполнить следующие команды. В режиме конфигурации производится ввод команды «interface» имя и номер интерфейса.

```
#config t
Enter configuration commands, one per line. End with CNTL/Z.
(config)#hostname My_switch
My_switch(config)#interface ethernet ?
<0-0> IEEE 802.3
My_switch(config)#interface ethernet 0?
/
My_switch(config)#interface ethernet 0/?
<1-25> IEEE 802.3
My_switch(config)#interface ethernet 0/1
My_switch(config-if)#?
Interface configuration commands:
cdp Cdp interface subcommands
description Interface specific description
duplex Configure duplex operation
exit Exit from interface configuration mode
help Description of the interactive help system
no Negate a command or set its defaults
port Perform switch port configuration
shutdown Shutdown the selected interface
spanntree Spanning tree subsystem
vlan-membership VLAN membership configuration
My_switch(config-if)#
My_switch(config-if)#?
Interface configuration commands:
cdp Cdp interface subcommands
description Interface specific description
duplex Configure duplex operation
exit Exit from interface configuration mode
help Description of the interactive help system
no Negate a command or set its defaults
port Perform switch port configuration
shutdown Shutdown the selected interface
spanntree Spanning tree subsystem
vlan-membership VLAN membership configuration
My_switch(config-if)#
```

Конфигурирование описания интерфейсов

При установлении описаний, необходимо произвести установку в режим конфигурации, выбрать порт и выполнить команду «description» и само описание.

```
My_switch#config t
Enter configuration commands, one per line. End with CNTL/Z.
My_switch(config)#interface e 0/1
My_switch(config-if)#description finance_vlan
My_switch(config-if)#int f0/26
My_switch(config-if)#description trunk_to_Building_4
My_switch(config-if)#
```

Информацию об интерфейсе можно получить с помощью следующей команды.

```
My_switch#show int e0/1
Ethernet 0/1 is Enabled
Hardware is Built-in 10Base-T
Address is 0030.80C7.0180
MTU 1500 bytes, BW 100000 Kbits
802.1d STP State: Forwarding Forward Transitions: 1
Port monitoring: Disabled
Unknown unicast flooding: Enabled
Unregistered multicast flooding: Enabled
Description: finance_vlan
Duplex/Flow Control setting: Half duplex
Enhanced Congestion Control: Disabled
Загрузочную информацию можно получить с помощью команды «show run».
My_switch#show run
Current configuration:
!
hostname My_switch
!
ip address 186.124.2.0 255.255.240.0!
!
interface Ethernet 0/1
description 'finance_vlan'
!
interface FastEthernet 0/26
description 'trunk_to_Building_4'
!
interface FastEthernet 0/27
line console
end
```

Применение команды «duplex»

Для конфигурации дуплекса применяется команда «duplex».

```
My_switch(config)#int f0/26
My_switch(config-if)#duplex full
My_switch(config-if)#^Z
```

Таблица 2.2 – Параметры команды duplex

Параметр	Описание параметра
Auto	Автоматическое определение режима
Full	Произвести установку в полнодуплексном режиме
Full-flow-control	Использование полнодуплексного режима на 100 Мбит/с
Half	Использование полудуплексного режима на 10 Мбит/с

Очистка памяти выполняется командой «delete»

```
My_Switch#delete nvram
```

This command resets the switch with factory defaults. All system parameters will revert to their default factory settings. All static and dynamic addresses will be removed.

Reset system with factory defaults, [Y]es or [N]o?YES

После проделанных операций выполняется очистка загрузочной информации.

Порядок выполнения работы:

- 1) изучение теоретического материала лабораторной работы;
- 2) разбор исходных примеров, приведенных в данной работе;
- 3) выполнение задания, предлагаемого в лабораторной работе.

Задания

Вначале необходимо очистить все информацию из Switch и выполнить указанные ниже операции.

Установить IP-адрес и шлюз для устройства, изменить имя Switch, установить разрешенный пароль на консоль и на привилегированный режим, разрешенный секрет. Порты ethernet 0/1, ethernet 0/2, ethernet 0/3 установить в полудуплексный режим, а FastEthernet 0/26 в полнодуплексный режим, FastEthernet 0/27 в полнодуплексный режим на 100 Мбит/с, а все остальные ethernet порты в автоматический режим. Проверить выполнение установленных параметров. Просмотреть информацию о Switch. Просмотрите все подпункты меню устройства.

После того как вы убедитесь, что все действия были выполнены правильно, необходимо позвать преподавателя и представить для контроля выполненное задание.

Содержание отчета и его форма:

Отчет должен иметь форму согласно оформлению простого реферата. Титульный лист должен включать: название дисциплины, название лабораторной работы, фамилию и инициалы сдающего студента, номер группы, фамилию и инициалы принимающего преподавателя. Основная часть лабораторной работы должна содержать:

- а) ответы на контрольные вопросы, предложенные в лабораторной работе;
- б) подробное рассмотрение предложенных примеров лабораторной работы
- в) описание каждого этапа выполнения задания;
- г) выводы по проделанной работе.

Задание должно быть выполнено в соответствии с примерами лабораторной работы.

Контрольные вопросы и защита работы:

Контрольные вопросы.

1. В чем заключается регистрация Switch?
2. Меню Switch.
3. Команды пользовательского режима.
4. Порты устройства Switch.
5. Установление паролей.
6. Что означает команда интерфейс?
7. Что означает команда duplex.
8. Параметры команды duplex.
9. Очистка и сохранение загрузочной информации.

Защита работы

Защита работы заключается:

- 1) в выполнении задания и оформлении отчета в соответствии с подпунктом «Содержание отчета и его форма»;
- 2) в ответах на контрольные вопросы по лабораторной работе;
в ответах на дополнительные вопросы

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],
Router(config-if)#no shut
```

Команда no shut (сокращение no shutdown) используется для того, чтобы бы интерфейс был активным (без этой команды возможно произвольное временное отключение интерфейса). Обратная команда – shut, выключит интерфейс.

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 команды

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. После этого он может автоматически определить соседние устройства, на которых также выполняется протокол CDP. Среди найденных устройств будут не только те, которые работают с протоколом IP.

CDP позволяет администраторам иметь доступ к данным о другом сетевом устройстве, к которому есть непосредственное соединение.

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

Команды ping и traceroute

Для диагностики возможности установления связи в сетях используются протоколы типа 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.

Соединение с сетевым устройством Cisco

Создайте в Packet Tracer топологию, изображённую на рисунке с использованием модели маршрутизатора по умолчанию - Generic. Назовите устройства так, как вы видите на рисунке 1: Router1, Router2 и Router4.

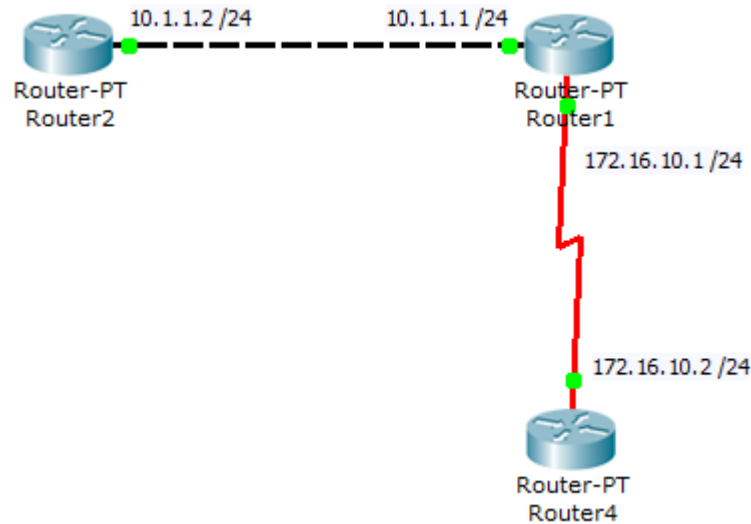


Рис.1

Черная линия означает Ethernet соединение. Красная означает последовательное соединение. Для создания последовательного соединения выбираем последовательное соединение точка-точка (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.

Ознакомление с сетевым устройством Cisco.

1. Для выбора сетевого устройства Router1 нажмите в рабочей области программы на изображение нужного. Откроется окно настроек сетевого устройства. Здесь выберем вкладку CLI для управления устройством.

2. В середине экрана Сетевого устройства Router1 вы увидите
Continue with configuration dialog? [yes/no]:

Введите: "no" и нажмите клавишу <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 RETURN to get started
```

Основные команды сетевого устройства

1. Войдите сетевое устройство Router1

```
Router>
```

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

```
Router>?
```

Клавишу Enter нажимать не надо.

3. Теперь войдите в 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 "parol".

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

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 FastEthernet0/0
```

```
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)#int fa0/0
```

Мы использовали сокращенное имя интерфейса.

4. Для каждой команды мы можем выполнить противоположную команду, поставивши 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 FastEthernet 0.

```
Router2(config)#interface fa0/0
```

Включите интерфейс.

```
Router2( config-if)#no shutdown
```

Теперь, когда интерфейсы на двух концах нашего Ethernet соединения включены на экране появится сообщение о смене состояния интерфейса на активное.

7. Перейдём к конфигурации последовательных интерфейсов. Зайдём на Router1. Проверим, каким устройством выступает наш маршрутизатор для последовательной линии связи: окончательным устройством DTE (data terminal equipment) либо устройством связи DCE (data circuit)

```
Router1#show controllers S2/0
```

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 s2/0
```

```
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 serial2/0.

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 s2/0
```

```
Router1#sh int e0/0
```

```
Router2#sh int e0/0
```

```
Router4#sh int s2/0
```

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.

```
FastEthernet0/0 is up, line protocol is up
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
Serial2/0 is up, line protocol is up
  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, P - Phone
Device ID        Local Intrfce  Holdtime    Capability    Platform    Port ID
Router4          Ser 2/0        138         R             PT1000       Ser 2/0
Router2          Fas 0/0        142         R             PT1000       Fas 0/0
```

Мы сделали всё правильно. Видим, что наш маршрутизатор router1 соединён с интерфейсом Fas 0/0 (**Port ID**) маршрутизатора (**Capability**) router2 (**Device ID**) серии 1000 (**Platform**) через интерфейс Fas 0/0 (**Local Intrfce**) и с интерфейсом Ser 2/0 маршрутизатора router4 серии 1000 через интерфейс Ser 2/0.

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.

10. 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.

11. 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
```

Для экономии полосы пропускания низкоскоростных устройств CDP можно отключить

```
router1 (config)#no cdp run
```

и снова включить для всего устройства

```
router1 (config)#cdp run
```

12. Иногда необходимо отключить CDP для определённого интерфейса, например при его узкой полосе пропускания или в целях безопасности. На устройстве router1, отключите CDP на интерфейсе FastEthernet 0/0.

```

router1 (config)#interface fa0/0
router1 ( config-if)#no cdp enable
router1 (config)#Ctrl-Z
router1#show cdp interface

```

В полученном выводе вы не увидите сведений об FastEthernet 0/0.

Настройка IP адресов интерфейсов

1. Подключимся к устройству **Router1** и установим IP адрес Ethernet интерфейса

```

Router1(config)#interface fa0/0
Router1( config-if)#ip address 10.1.1.1 255.255.255.0

```

2. Теперь назначим интерфейсу S0/0 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 FastEthernet 0/0 IP адрес 10.1.1.2 255.255.255.0

```

Router2(config)#interface fa0/0
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 2/0.

```

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
FastEthernet0/0	10.1.1.1	YES	manual	up	up
FastEthernet1/0	unassigned	YES	manual	up	down
Serial2/0	172.16.10.1	YES	manual	up	up

Router2#show ip in bri

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	10.1.1.2	YES	manual	up	up
FastEthernet1/0	unassigned	YES	manual	up	down
Serial2/0	unassigned	YES	manual	administratively down	down

Router4#show ip in bri

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	unassigned	YES	manual	administratively down	down
FastEthernet1/0	unassigned	YES	manual	administratively down	down
Serial2/0	172.16.10.2	YES	manual	up	up

8. Подключимся к устройству **Router1**. Вы должны успешно пропинговать непосредственно подсоединённый FastEthernet 0/0 интерфейс на устройстве Router2.

Router1#ping 10.1.1.2

```
Type escape sequence to abort.
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 = 1/4/7 ms
```

Попробуем пропинговать интерфейс Serial 2/0 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 = 6/6/7 ms
```

Успешно.

9. Вернёмся на Router2. Вы должны успешно пропинговать адрес 10.1.1.1 непосредственно подсоединённого FastEthernet 0/0 интерфейса на устройстве Router1. Вернёмся на Router4. Вы должны успешно пропинговать адрес 172.16.10.1 непосредственно подсоединённого интерфейса Serial 2/0 на устройстве Router1. Попробуем пропинговать интерфейс FastEthernet 0/0 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)
```

Неудача. Попробуем пропинговать адрес 10.1.1.2 FastEthernet 0/0 интерфейса на устройстве Router2. Неудача.

10. Вернёмся на Router2. Попробуем пропинговать адрес 172.16.10.1 интерфейса

Serial 2/0 на устройстве Router1. Неудача. Попробуем пропинговать адрес 172.16.10.2 интерфейса Serial 2/0 на устройстве Router4. Неудача.

Неудачи нас постигли потому, что мы не настроили на маршрутизаторах маршрутизацию.

11. Зайдите на устройство Router1. Определите пути прохождения пакетов на Router2

```
Router1#tracert 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 parol
```

3. Войдите на устройство Router2 и установим telnet-соединение с устройством Router1. Для этого мы используем IP адресу его интерфейсу FastEthernet 0/0

```
Router2#telnet 10.1.1.1
```

4. Мы увидим просьбу ввести пароль. Введите пароль parol и нажмите <enter>. Заметьте, что имя сетевого устройства поменялось на “Router1”, потому, что мы установили telnet-соединение с Router1. Команда

```
Router1>show user
```

```
* 67 vty 0 idle 00:00:00 10.1.1.2
```

покажет, что соединение осуществлено от адреса 10.1.1.2 устройства router2 .

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 в 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	1	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 [confirm]

Сохраните проект в целом и конфигурацию каждого роутера в отдельности (в текстовый файл).

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 PacketTracer практическую часть.
4. Получите вариант и выполните в PacketTracer задание для самостоятельной работы
5. Предъявите преподавателю результат выполнения задания для самостоятельной работы. Продемонстрируйте свои файлы с конфигурациями роутеров. 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. Получите 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

Таблица 3.

Выберите в дизайнера подходящие устройства и создайте топологию, изображённую на рисунке 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

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 4.

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 5.

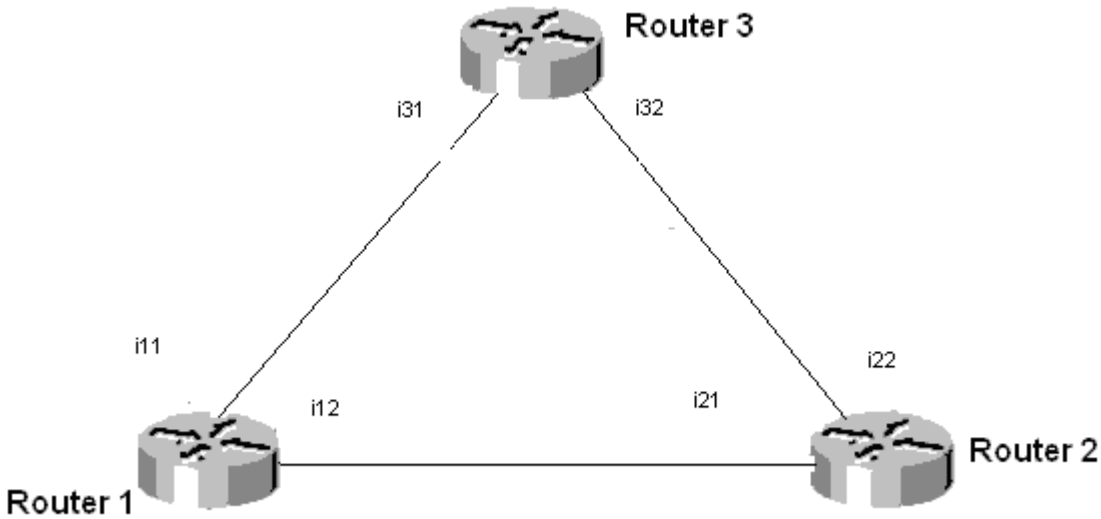


Рис 2.

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 6.

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 сессии. Сохраните скриншот консоли Router2.

8. Сохраните топологию.

Содержание отчёта.

Отчёт готовится в электронном виде и распечатывается. Отчёт содержит

1. Скриншот топологии из рисунка 1
2. Конфигурации трёх маршрутизаторов из .txt файлов, созданных при выполнении практической части.
- 3 Скриншот топологии из рисунка 2
4. Все скриншоты, указанные в **Задании для самостоятельной работы**
5. Конфигурации трёх маршрутизаторов из .txt файлов.

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:

```
...
C192.168.4.0/24 is directly connected, Ethernet0
  10.0.0.0/16 is subnetted, 3 subnets
C10.3.0.0 is directly connected, Serial0
C10.4.0.0 is directly connected, Serial1
```

C10.5.0.0 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):

S 10.6.0.0 via Serial1

S 10.7.0.0 [1/0] via 10.4.0.2

Когда интерфейс падает, все статические маршруты, отображаемые на этот интерфейс, удаляются из таблицы маршрутов. Если маршрутизатор не может больше найти адрес следующего хопа по пути к адресу, указанному в статическом маршруте, то маршрут исключается из таблицы.

Заметим, что для сетей типа Ethernet рекомендуется всегда использовать форму (2) команды ip route. Ethernet интерфейс на маршрутизаторе, как правило, соединён с несколькими Ethernet интерфейсами других устройств в сети. Указание в команде ip route IP адреса позволит маршрутизатору правильно сформировать физический адрес выходного пакета по своим ARP таблицам.

Маршрутизация по умолчанию.

Совсем не обязательно, чтобы каждый маршрутизатор обслуживал маршруты ко всем возможным сетям назначения. Вместо этого маршрутизатор хранит маршрут по умолчанию или шлюз последнего пристанища (last resort). Маршруты по умолчанию 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 создается 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          -          00D0.58B7.80A1  ARPA   FastEthernet0/0
```

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	-	00D0.BA88.005C	ARPA	FastEthernet0/0

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	10.1.1.1	0	00D0.58B7.80A1	ARPA	FastEthernet0/0
Internet	10.1.1.2	-	00D0.BA88.005C	ARPA	FastEthernet0/0

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	-	00D0.58B7.80A1	ARPA	FastEthernet0/0
Internet	10.1.1.2	2	00D0.BA88.005C	ARPA	FastEthernet0/0

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
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       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, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
```

Gateway of last resort is not set

```
10.0.0.0/24 is subnetted, 1 subnets
C      10.1.1.0 is directly connected, FastEthernet0/0
```

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

```
10.0.0.0/24 is subnetted, 1 subnets
C    10.1.1.0 is directly connected, FastEthernet0/0
172.16.0.0/24 is subnetted, 1 subnets
S    172.16.10.0 [1/0] via 10.1.1.1
```

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, Serial2/0
```

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, Serial2/0
```

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 выходу во внешний мир: через интерфейсы с адресами 10.1.1.1 и 172.16.10.1, соответственно. Поэтому, можно не определять на какие подсети мы маршрутизируем пакеты и использовать маршрутизацию по умолчанию.

1. Вначале удалим старые маршруты.

```
Router2(config)#no ip route 172.16.10.0 255.255.255.0 10.1.1.1
```

```
Router4(config)#no ip route 10.1.1.0 255.255.255.0 172.16.10.1
```

2. И назначим маршруты по умолчанию.

```
Router2(config)#ip route 0.0.0.0 0.0.0.0 10.1.1.1
```

```
Router4(config)#ip route 0.0.0.0 0.0.0.0 172.16.10.1
```

3. Посмотрите таблицу маршрутов на всех устройствах.

```
Router2#sh ip route
```

```
Gateway of last resort is 10.1.1.1 to network 0.0.0.0
```

```
    10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, FastEthernet0/0
S*     0.0.0.0/0 [1/0] via 10.1.1.1
```

Router4#**sh ip route**

```
Gateway of last resort is 172.16.10.1 to network 0.0.0.0
```

```
    172.16.0.0/24 is subnetted, 1 subnets
C       172.16.10.0 is directly connected, Serial2/0
S*     0.0.0.0/0 [1/0] via 172.16.10.1
```

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 маршрутизатора в отдельный файл.

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 Packet Tracer практическую часть.
4. Получите вариант и выполните в Packet Tracer 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 Packet Tracer топологию, представленную на рисунке. Использовать необходимые маршрутизаторы.

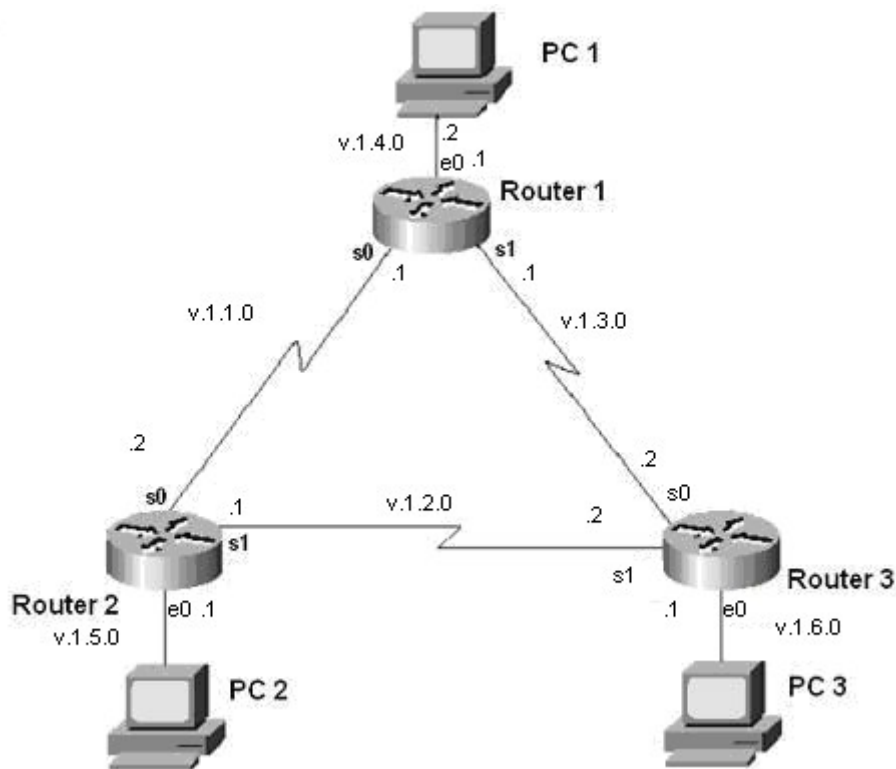


Рисунок 1

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 делает 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 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 .txt файлов, созданных при выполнении практической части.
3. Скриншот топологии из рисунка 1 с адресами своего варианта
4. Таблицу 2 с адресами своего варианта
5. Конфигурации трёх маршрутизаторов из .txt 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).

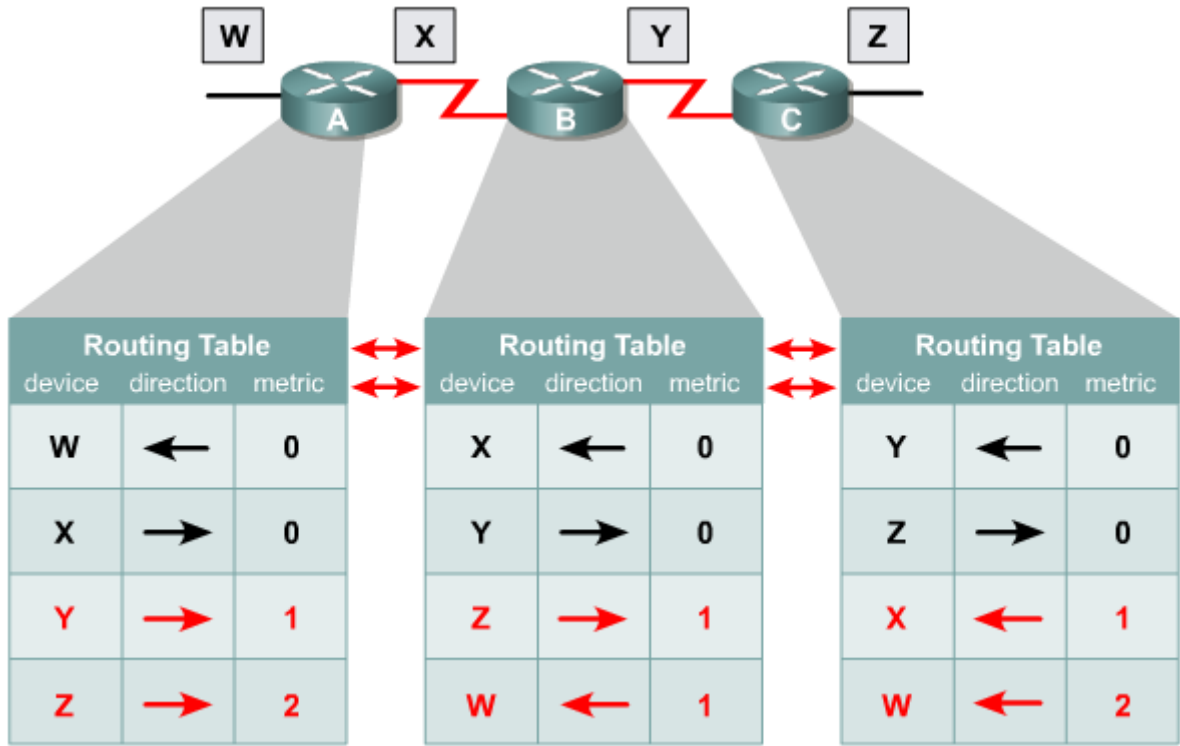


Рис.1. Дистанционно-векторная маршрутизация.

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` и `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.

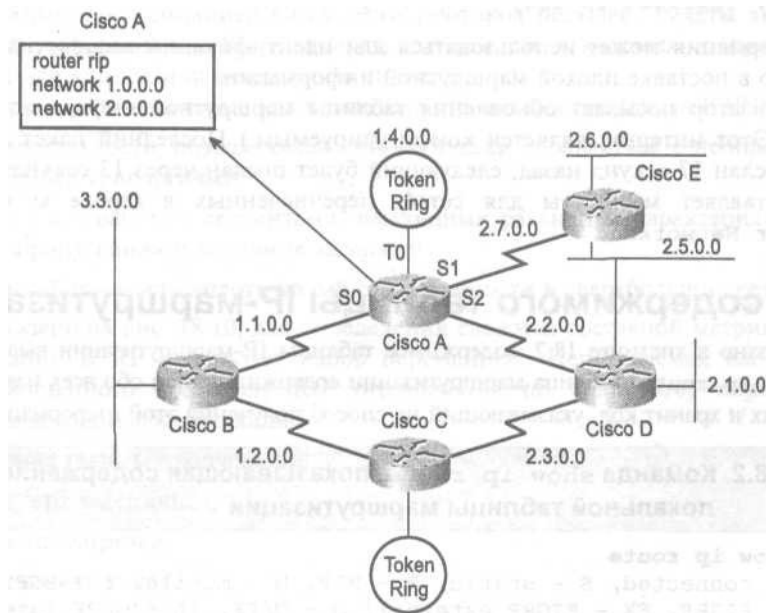


Рис.2.

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 протокол характеризует универсальность, позволяющую 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 примере на рис.3 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.

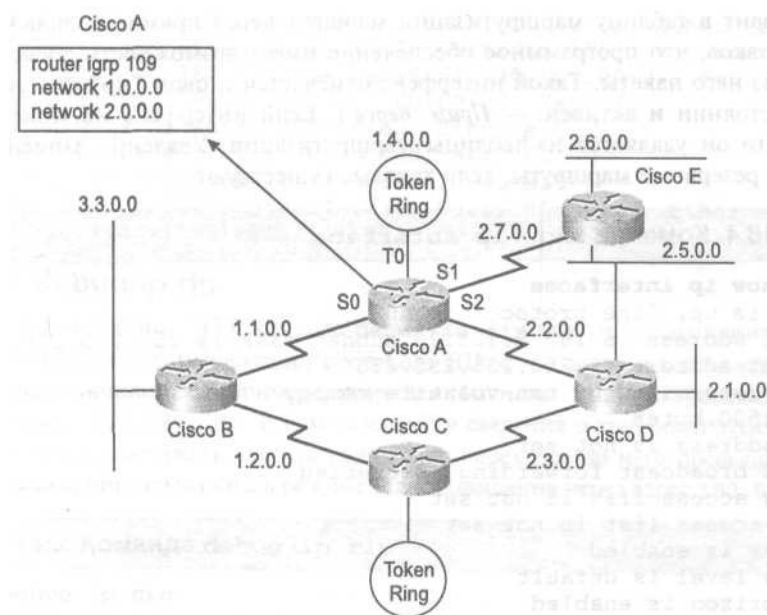


Рис.3.

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 отсылает 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.

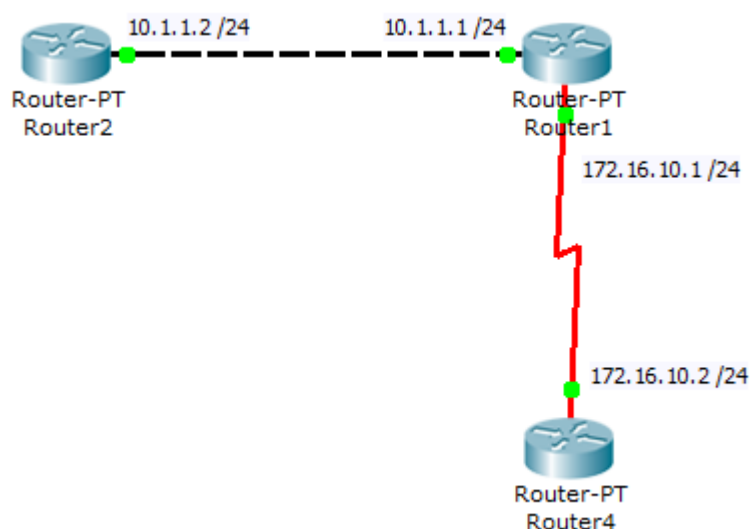


Рисунок 4.

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 172.16.0.0/16. 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 6 seconds
Invalid after 180 seconds, hold down 180, flushed after 240
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Redistributing: rip
Default version control: send version 1, receive any version
  Interface          Send Recv Triggered RIP Key-chain
  Serial2/0           1     2  1
  FastEthernet0/0     1     2  1
Automatic network summarization is in effect
Maximum path: 4
Routing for Networks:
  10.0.0.0
  172.16.0.0
Passive Interface(s):
Routing Information Sources:
  Gateway            Distance      Last Update
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, FastEthernet0/0
R    172.16.0.0/16 [120/1] via 10.1.1.1, 00:00:15, FastEthernet0/0

```

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**

```

R    10.0.0.0/8 [120/1] via 172.16.10.1, 00:00:22, Serial2/0
172.16.0.0/24 is subnetted, 1 subnets
C    172.16.10.0 is directly connected, Serial2/0

```

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.

5. Командой `debug ip rip` посмотрим как маршрутизаторы обмениваются маршрутной информацией. Например, для Router1 имеем повторяющиеся каждые 30 секунд сообщения

6.

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 трассировку

```
Router1# no debug ip rip
```

Сохраните конфигурацию.

EIGRP

Остановим на всех маршрутизаторах RIP командой

```
Router(config)#no router rip.
```

1. Включим EIGRP на всех маршрутизаторах, образуя автономную систему с номером 100

```
Router1(config)# router eigrp 100
```

```
Router1(config-router)# network 172.16.10.0
```

```
Router1(config-router)# network 10.1.1.0
```

```
Router2(config)# router eigrp 100
```

```
Router2 (config-router)# network 10.1.1.0
```

```
Router4(config)# router eigrp 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 172.16.0.0/16. Это связано с классами IP адресов.

3. Командой `show ip protocols` посмотрим с какими параметрами работает протокол EIGRP. Например, для Router1 имеем

```

Routing Protocol is "eigrp 100 "
  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
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
Redistributing: eigrp 100
  Automatic network summarization is in effect
  Automatic address summarization:
    172.16.0.0/16 for FastEthernet0/0
      Summarizing with metric 20512000
    10.0.0.0/8 for Serial2/0
      Summarizing with metric 28160
  Maximum path: 4
  Routing for Networks:
    172.16.0.0
    10.0.0.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    172.16.10.2      90            8321604
    10.1.1.2         90            8350639
  Distance: internal 90 external 170

```

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, FastEthernet0/0
D    172.16.0.0/16 [90/20514560] via 10.1.1.1, 00:08:37, FastEthernet0/0

```

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**

```

D    10.0.0.0/8 [90/20514560] via 172.16.10.1, 00:12:30, Serial2/0
172.16.0.0/24 is subnetted, 1 subnets
C    172.16.10.0 is directly connected, Serial2/0

```

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.

7. Командами `debug ip eigrp transactions` и `debug ip eigrp events` посмотрите как маршрутизаторы обмениваются маршрутной информацией.

Сохраните конфигурацию.

OSPF

Остановим на всех маршрутизаторах EIGRP командой

```
Router(config)#no router eigrp 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 0.0.0.255 area 0
```

```
Router1(config-router)# network 10.1.1.0 0.0.0.255 area 0
```

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

```
Router2(config-router)# network 10.1.1.0 0.0.0.255 area 0
```

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

```
Router4(config-router)# network 172.16.10.0 0.0.0.255 area 0
```

2. Команда `show running-config` показывает, что маршрутизаторы поняли наши команды в том же виде, как мы их и задавали.

3. Командой `show ip protocols` посмотрим с какими параметрами работает протокол OSPF. Например, для Router1 имеем

```
Routing Protocol is "ospf 100"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 172.16.10.1
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    172.16.10.0 0.0.0.255 area 0
    10.1.1.0 0.0.0.255 area 0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.1.1.2         110          00:01:40
    172.16.10.2      110          00:01:45
  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, FastEthernet0/0
172.16.0.0/24 is subnetted, 1 subnets
O    172.16.10.0 [110/782] via 10.1.1.1, 00:08:26, FastEthernet0/0
```

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**

```

10.0.0.0/24 is subnetted, 1 subnets
O    10.1.1.0 [110/782] via 172.16.10.1, 00:03:26, Serial2/0
172.16.0.0/24 is subnetted, 1 subnets
C    172.16.10.0 is directly connected, Serial2/0

```

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.

8. 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 EIGRP?
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 EIGRP?
15. Уточните, что в EIGRP понимается под автономной системой?
16. В чём различие в формате команды `router` для EIGRP и OSPF?
17. В чём различие в формате команд `network` для RIP, EIGRP 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 Packet Tracer 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.

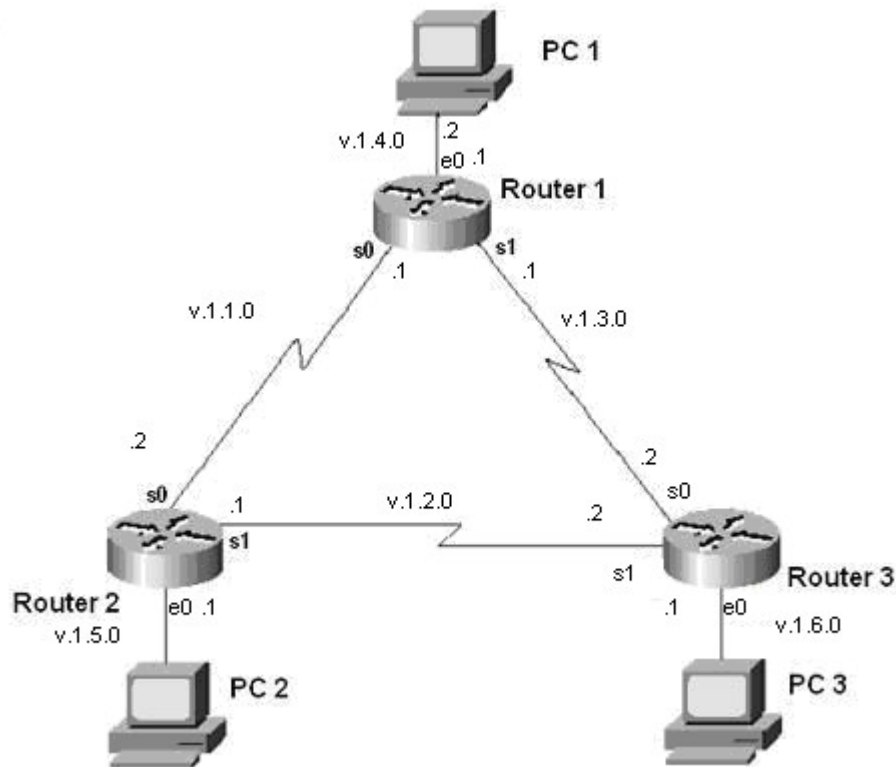


Рисунок 5.

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 `tr` 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 посмотрите таблицу маршрутизации командой `show ip route`. Сделать скриншоты. Например, для маршрутизатора `router1` для варианта 1 (`v=1`) имеем

```

C      1.1.4.0/24 is directly connected, Ethernet0
C      1.1.1.0/24 is directly connected, Serial0
C      1.1.3.0/24 is directly connected, Serial1
R      1.1.6.0/24 [120/1] via 1.1.3.2, 00:01:43, Serial1
R      1.1.5.0/24 [120/1] via 1.1.1.2, 00:06:30, Serial0
R      1.1.2.0/24 [120/1] via 1.1.1.2, 00:03:35, Serial0

```

7. На каждом компьютере выполните команды трассировки `tracert` других компьютеров. Сделайте скриншоты. Всего шесть скриншотов. Например, трассировка из PC1 на PC2 для варианта 1 (v=1) имеет вид

```
PC1:#tracert 1.1.5.2
```

```

"Type escape sequence to abort."
Tracing the route to 1.1.5.2

```

```

 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`. Сделайте скриншоты. Например, для маршрутизатора router1 для варианта 1 (v=1) имеем

```

C      1.1.4.0/24 is directly connected, Ethernet0
C      1.1.3.0/24 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).

11. Сохраните конфигурацию.

12. Отключите RIP и настройте на каждом маршрутизаторе динамическую маршрутизацию по протоколу IGRP.

13. На каждом маршрутизаторе посмотреть таблицу маршрутизации командой `show ip route`. Сделать скриншоты. Например, для маршрутизатора `router1` для варианта 1 ($v=1$) имеем

```
C    1.1.4.0/24 is directly connected, Ethernet0
C    1.1.1.0/24 is directly connected, Serial0
C    1.1.3.0/24 is directly connected, Serial1
I    1.1.6.0/24 [100/651] via 1.1.3.2, 00:02:38, Serial1
I    1.1.5.0/24 [100/651] via 1.1.1.2, 00:04:26, Serial0
I    1.1.2.0/24 [100/651] via 1.1.1.2, 00:08:28, Serial0
```

14. Проверьте, что вы всё сделали правильно. На каждом компьютере выполните команд трассировки `tracert` других компьютеров.

15. Сохраните конфигурацию.

16. Отключите IGRP и настройте на каждом маршрутизаторе динамическую маршрутизацию по протоколу OSPF.

17. На каждом маршрутизаторе посмотреть таблицу маршрутизации командой `show ip route`. Сделать скриншоты. Например, для маршрутизатора `router1` для варианта 1 ($v=1$) имеем

```
C    1.1.4.0/24 is directly connected, Ethernet0
C    1.1.1.0/24 is directly connected, Serial0
C    1.1.3.0/24 is directly connected, Serial1
O    1.1.5.0/24 [110/65] via 1.1.1.2, 00:00:39, Serial0
O    1.1.2.0/24 [110/65] via 1.1.3.2, 00:00:22, Serial1
O    1.1.6.0/24 [110/65] via 1.1.3.2, 00:00:26, Serial1
```

18. Проверьте, что вы всё сделали правильно. На каждом компьютере выполните команд трассировки `tracert` других компьютеров.

19. 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 `.txt` файлов, созданных при выполнении практической части для каждого протокола маршрутизации. Итого содержимое 9 (девяти) 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 `.txt` файлов, созданных при выполнении задание для самостоятельной работы для каждого протокола маршрутизации. Итого содержимое 9 (девяти) 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

10101100000111101000000000010001 <-> 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

11111111111111110000000000000000 <-> 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 255.255.255.0 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 требование 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 бесклассовой 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 новый адрес 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 А 44.0.0.0/8, в которой рассматривается 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

Таблица 1.

Первые два октета (16 бит) представляют адрес подсети. Так как первые 16 бит адреса каждой из этих восьми подсетей уникальны, то классовый 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 безклассовой 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

Таблица 2.

Из таблицы видно, что адреса имеют общий 23-битовый префикс 11001111 00010101 0011011. Дополняя префикс справа нулями 11001111 00010101 00110110 00000000, получим надсеть 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

Таблица 3.

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

Таблица 4.

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.

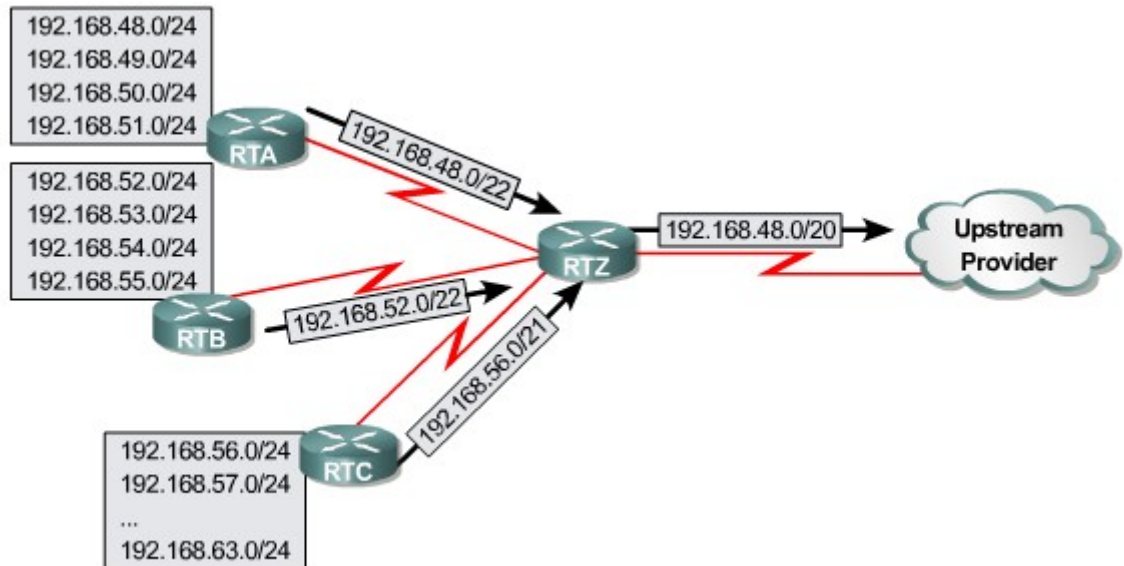


Рис. 1

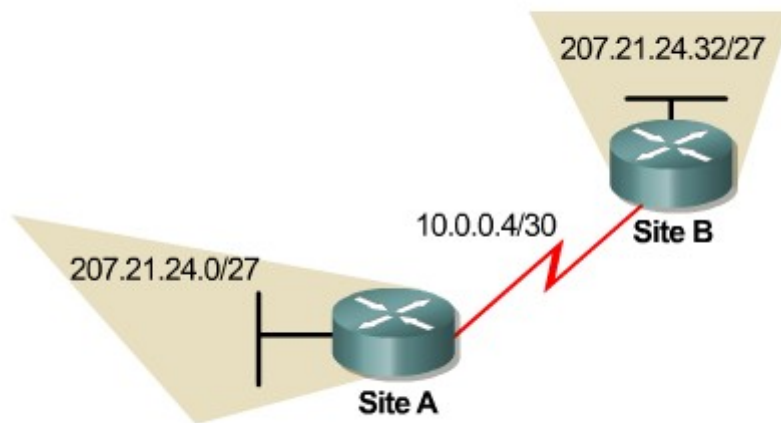


Рис. 2

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

Таблица 5.

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

Таблица 6.

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. Поднимите интерфейсы и проверьте сеть командой **show cdp neighbors**.
2. Назначьте адреса согласно рисунку. Проверьте назначение командой **show ip interface brief**.

3. Для компьютера 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**

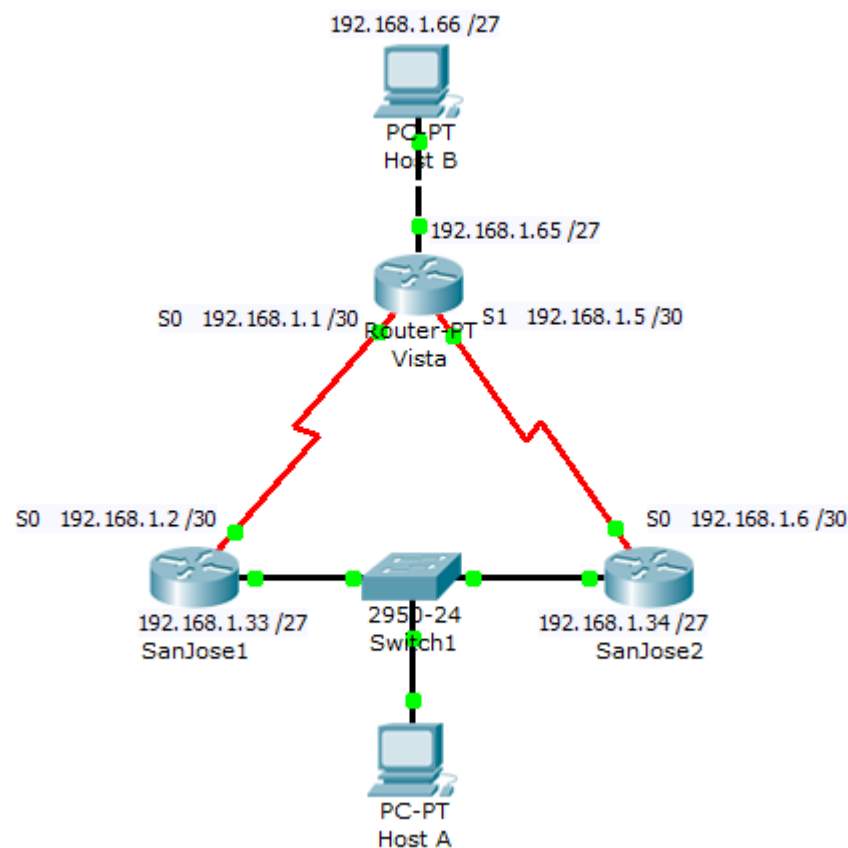


Рис.3

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      192.168.1.0/30 is directly connected, Serial2/0
C      192.168.1.4/30 is directly connected, Serial3/0
R      192.168.1.32/27 [120/1] via 192.168.1.6, 00:00:15, Serial3/0
                        [120/1] via 192.168.1.2, 00:00:24, Serial2/0
C      192.168.1.64/27 is directly connected, FastEthernet0/0
```

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.

6. 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, Serial2/0
R    192.168.1.4/30 [120/1] via 192.168.1.34, 00:00:18, FastEthernet0/0
C    192.168.1.32/27 is directly connected, FastEthernet0/0
R    192.168.1.64/27 [120/1] via 192.168.1.1, 00:00:18, Serial2/0
```

Сделайте скриншот.

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

```
R    192.168.1.0/30 [120/1] via 192.168.1.33, 00:00:09, FastEthernet0/0
C    192.168.1.4/30 is directly connected, Serial3/0
C    192.168.1.32/27 is directly connected, FastEthernet0/0
R    192.168.1.64/27 [120/1] via 192.168.1.5, 00:00:26, Serial3/0
```

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 Packet Tracer практическую часть.
4. Выполните в Packet Tracer 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.

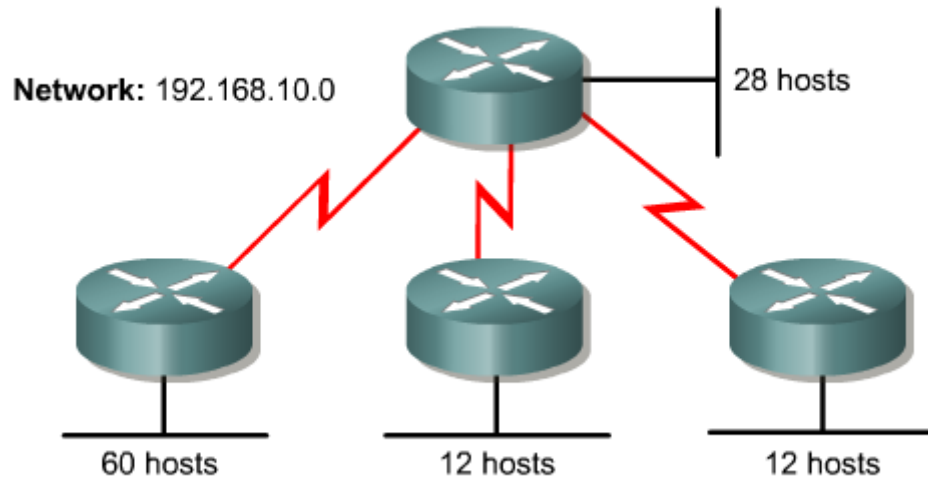


Рис 4. Вариант 1.

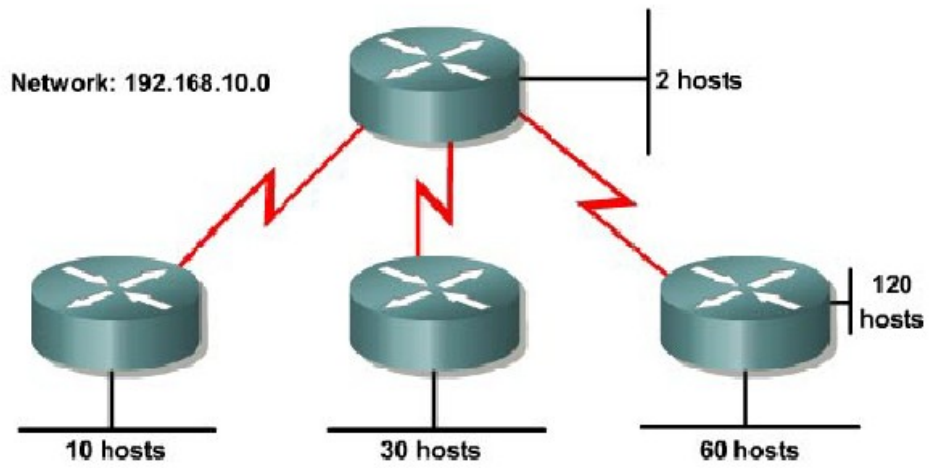


Рис 5. Вариант 2

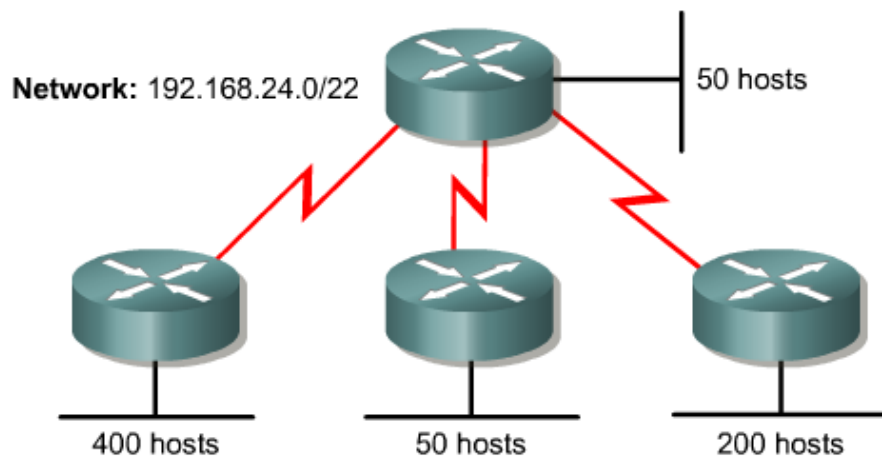


Рис 6. Вариант 3.

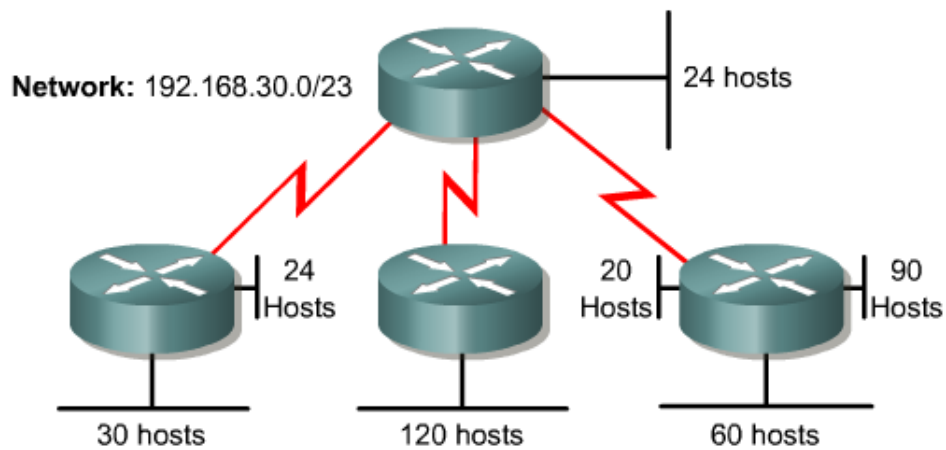
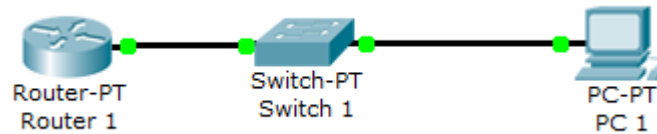


Рис 6. Вариант 4.

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 из .txt файлов, созданных при выполнении практической части.
2. Скриншот топологии вашего варианта с адресами и масками.
4. Конфигурации всех маршрутизаторов из .txt 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, помещается 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

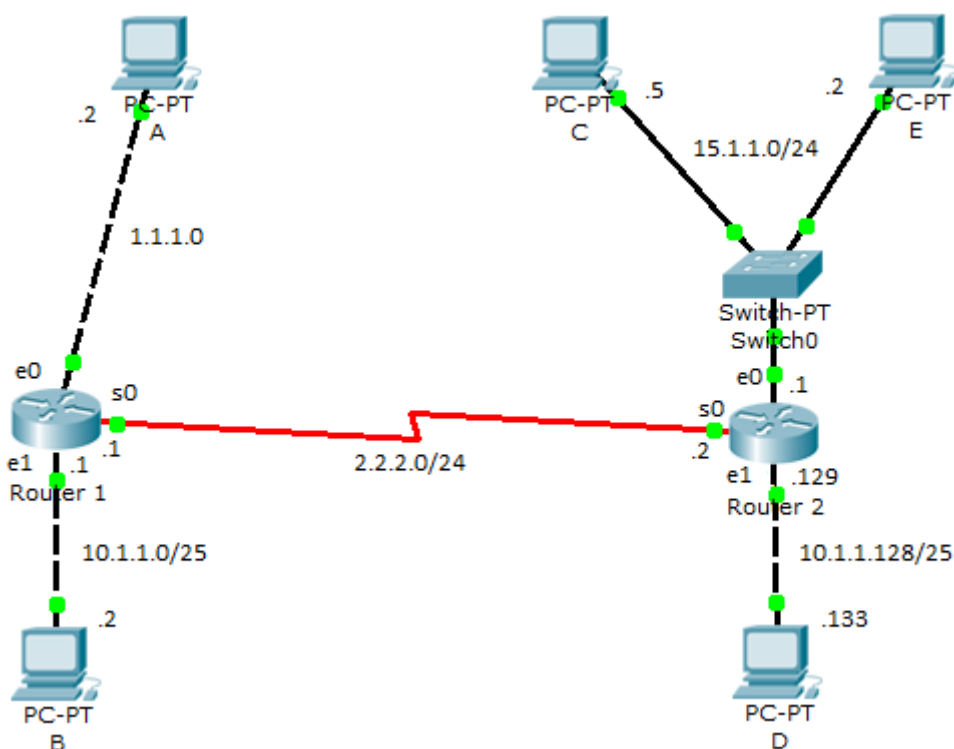


Рисунок 1.

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 неименованного 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.

Именованный 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 po в начале строки.

Для редактирования числовых ACLs следует его уничтожить и создать заново или изменить список офлайн и загрузить в устройство с помощью.

Практическая часть.

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

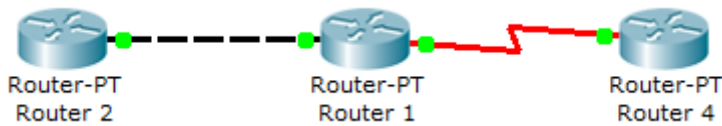


Рисунок 2.

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)#version 2
```

```
Router1(config- router)#network 24.0.0.0
```

На Router2

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

```
Router1(config- router)#version 2
```

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

и на Router4

```
Router4(config)#router rip
```

```
Router1(config-router)#version 2
```

```
Router4(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 FastEthernet0/0
```

```
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 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
deny host 24.17.2.18
permit any (4 match(es))
```

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,

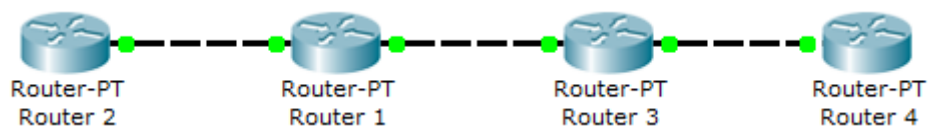


Рисунок 3.

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

```
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
    permit 175.10.1.0 0.0.0.255
```

Присоедините список как входной к интерфейсу Ethernet 1

```
router1(config)#interface FastEthernet1/0
```

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

Проверьте присоединение командой

```
router1# show running-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 (router4) и блокируем трафик от других устройств.

```
router1(config)# no access-list 2
router1(config)# access-list 2 permit 180.10.1.0 0.0.0.255
```

Проверьте, что он изменился

```
router1#show access-list
Standard IP access list 1
    permit 180.10.1.0 0.0.0.255
```

Присоедините список как входной к интерфейсу Ethernet 1

```
router1(config)#interface FastEthernet1/0
router1(config-if)#ip access-group 1 in
```

Проверьте присоединение командой

```
router1# show running-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 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 маршрутизаторе 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 FastEthernet0/0
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	A	B	C	E	D
A	+	+	+	-	-
B	+	+	+	+	+
C	+	+	+	+	+
E	-	+	+	+	+
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 fa0/0

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

router1(config-if)#int s2/0

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

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

Из\В	A	B	C	E	D
A	+	+	+	-	-
B	+	+	+	-	-
C	+	+	+	+	+
E	-	-	+	+	+
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 1 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 fa0/0

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 Serial2/0
```

```
router1(conf-if)# ip access-group 101 in
```

```
router1(conf-if)# interface fa0/0
```

```
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 `Serial2/0` 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 сокоашения 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 пропиnguем 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

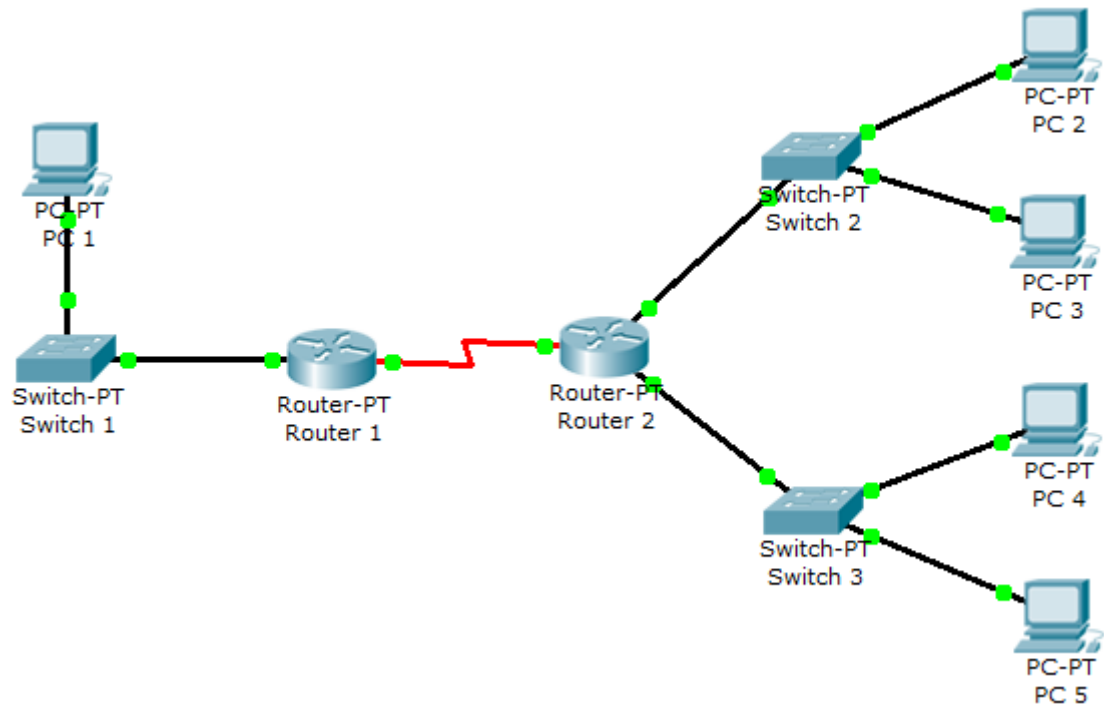


Рисунок 4.

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
Fa1/0		1.1.1.1/25
Serial2/0	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 serial2/0 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 Serial2/0
```

```
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
  permit ip any any
```

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 пинговать 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 пинговать 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 Serial2/0**

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

и

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

Router2(conf-if)#**no ip access-group 101 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 0.0.0.127 host 1.1.3.2
  permit ip any any
```

Применим список к исходящему трафику на интерфейсе Serial2/0 Router2

Router2(conf)#**interface Serial2/0**

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 вы можете увидеть так

```
Router2#sh ac
Extended IP access list 102
  deny ip 1.1.1.128 0.0.0.127 host 1.1.3.2 (8 match(es))
  permit ip any any

Router2#sh ac
Extended IP access list 102
  deny ip 1.1.1.128 0.0.0.127 host 1.1.3.2 (16 match(es))
  permit ip any any
```

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 Packet Tracer 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 из .txt файлов конфигураций, созданных при выполнении практической части.
4. Конфигурацию маршрутизаторов из .txt 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 .txt 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 позволяет 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 команда определяется 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 М с внутренним глобальным адресом Х. Пусть имеется пакет П от внутреннего локального адреса Y_i и порта P_i к внешнему адресу G и порту p. Пакет проходит через маршрутизаторМ. 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 М. 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 М 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.

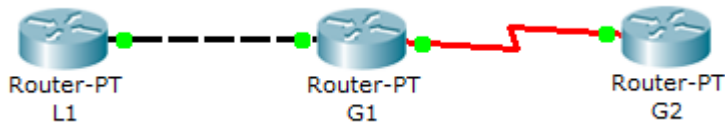


Рисунок 1

Создайте конфигурацию

	L1	G1	G2
Ethern	10.10.1.2 255.255.255.0	10.10.1.1 255.255.255.0	
Serial		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 телнет 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	00:01:18	
* 67 vty 0		idle	00:00:00	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 fastEthernet0/0 интерфейса как внутреннюю

```
G1(config)#interface fa0/0
```

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

Определим сеть serial2/0 интерфейса как внешнюю

```
G1(config-if)#interface s2/0
```

```
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 Serial2/0 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 10.10.1.2 175.10.1.1
```

```
L1# telnet 175.10.1.2
```

```
Password:G2
```

```
G2> show users
```

Line	User	Host(s)	Idle	Location
0 con 0		idle	00:03:49	
* 67 vty 0		idle	00:00:00	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
!
!
access-list 1 permit 10.10.1.0 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
tcp	169.10.1.50:1025	10.10.1.2:1025	175.10.1.2:23	175.10.1.2:23
tcp	169.10.1.50:1026	10.10.1.2:1026	175.10.1.2:23	175.10.1.2:23
tcp	169.10.1.50:1027	10.10.1.2:1027	175.10.1.2:23	175.10.1.2:23
tcp	169.10.1.50:1028	10.10.1.2:1028	175.10.1.2:23	175.10.1.2:23
tcp	169.10.1.50:1029	10.10.1.2:1029	175.10.1.2:23	175.10.1.2:23

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 serial2/0 (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 s2/0 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 Serial2/0 overload
ip classless
!
!
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	00:00:34	
* 67 vty 0		idle	00:00:00	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
tcp	175.10.1.1:1025	10.10.1.2:1025	175.10.1.2:23	175.10.1.2:23
tcp	175.10.1.1:1026	10.10.1.2:1026	175.10.1.2:23	175.10.1.2:23
tcp	175.10.1.1:1027	10.10.1.2:1027	175.10.1.2:23	175.10.1.2:23
tcp	175.10.1.1:1028	10.10.1.2:1028	175.10.1.2:23	175.10.1.2:23
tcp	175.10.1.1:1029	10.10.1.2:1029	175.10.1.2:23	175.10.1.2:23

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

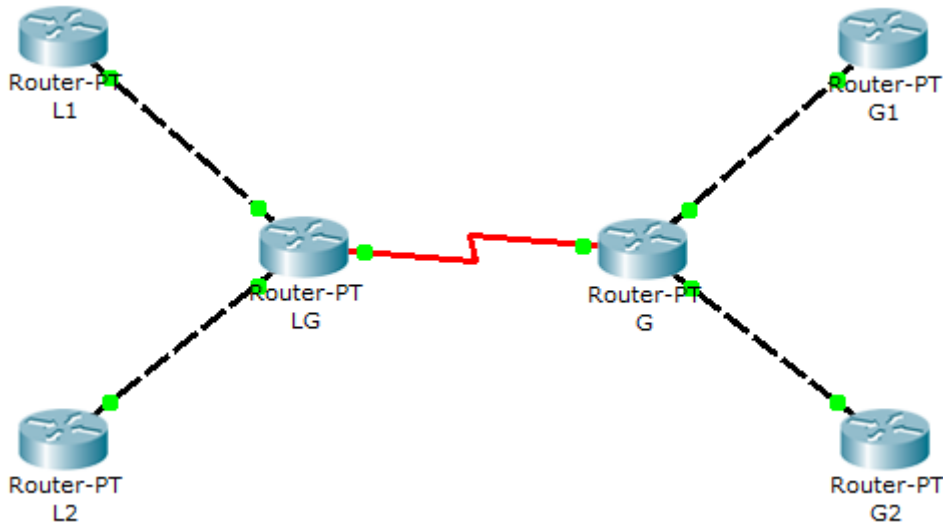


Рисунок 2

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

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**

G(config-router)#**network 1.1.1.0 0.0.0.255 area 10**

G(config-router)#**network 1.1.2.0 0.0.0.255 area 10**

G(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 s2/0
LG(config-if)#ip nat outside
LG(config-if)#interface fa0/0
LG(config-if)#ip nat inside
LG(config-if)#interface fa1/0
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 Serial2/0

```
LG(config)#ip nat inside source list 2 interface s2/0 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	00:08:20	
* 67 vty 0		idle	00:00:00	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	00:09:55	
* 67 vty 0		idle	00:00:00	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
67 vty 0		idle	00:03:56	1.1.1.1
* 68 vty 1		idle	00:00:00	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
67 vty 0		idle	00:04:36	1.1.1.1
* 68 vty 1		idle	00:00:00	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 Serial2/0.

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 Serial2/0.

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 Packet Tracer 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	
Serial0		175. v.1.1 255.255.255.0	175. v.1.2 255.255.255.0

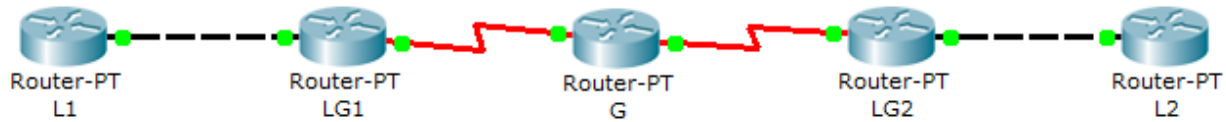
Сделайте те же скриншоты, которые сделаны в пункте 1 практической части.

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) входил 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 .txt файлов, созданных при выполнении практической части.
6. Конфигурацию маршрутизаторов из .txt файлов конфигураций, 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).

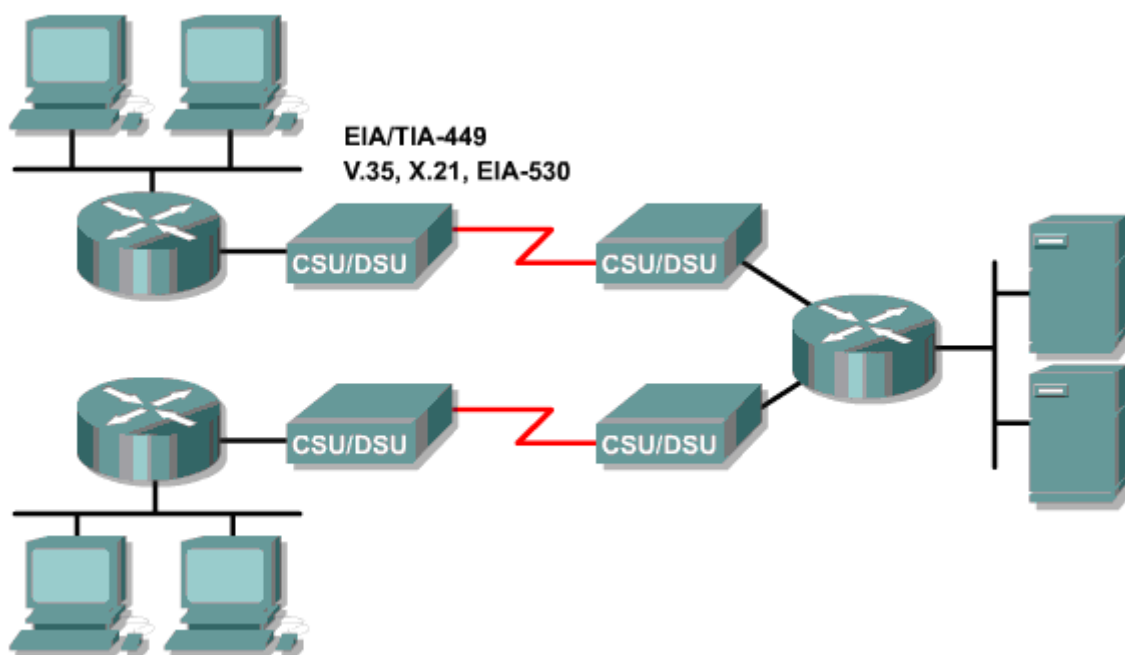


Рисунок 1.

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.

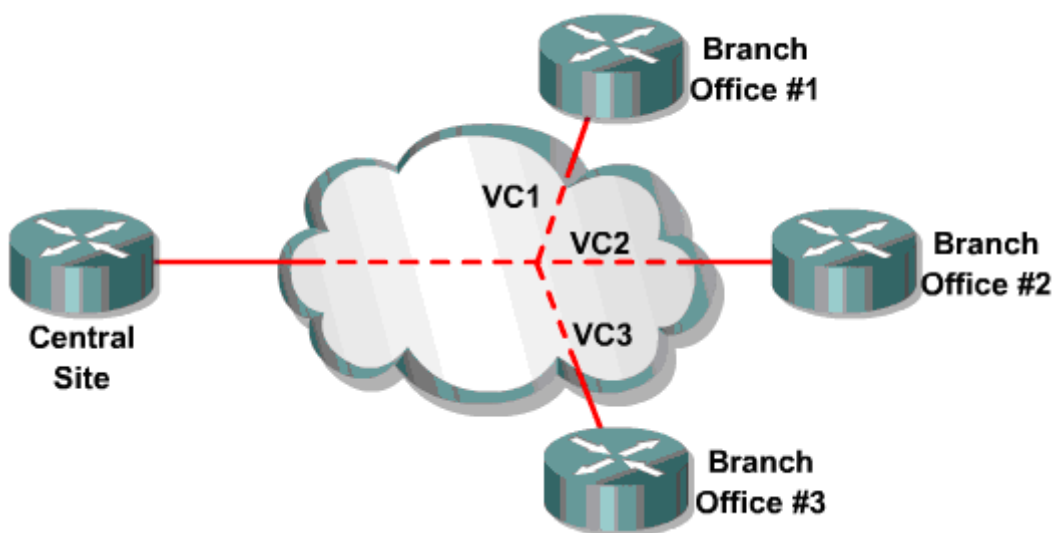


Рисунок 2.

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, Т3 и 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.

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

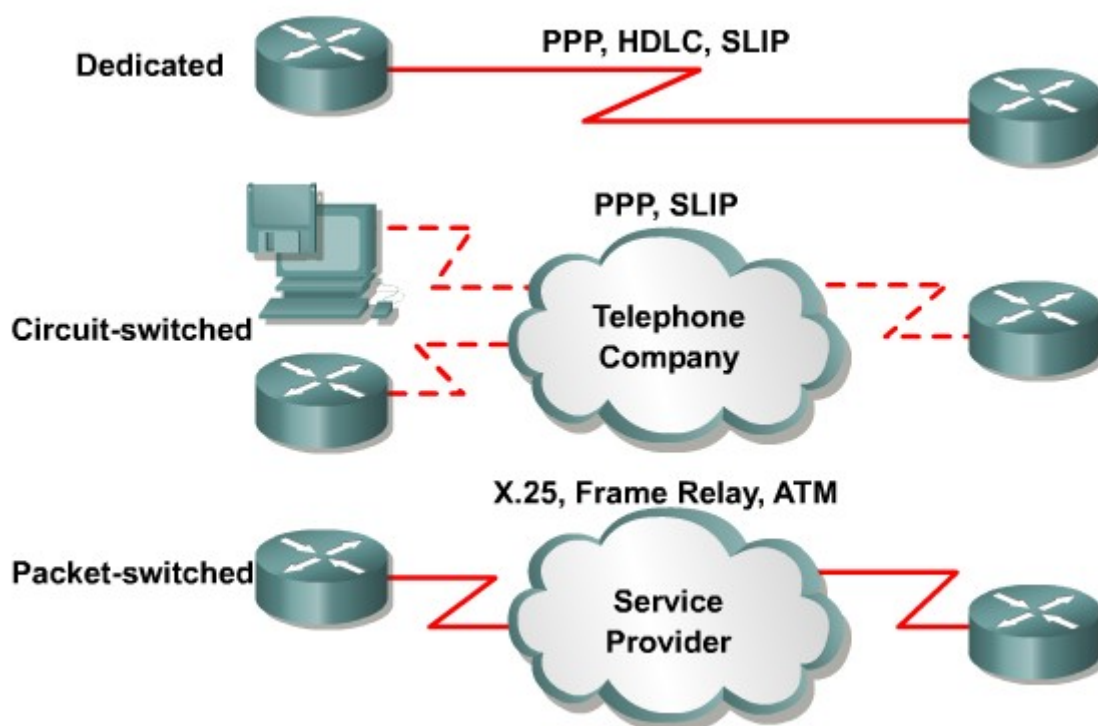


Рисунок 3.

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 надежности 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 высоконадежной 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 оконечное 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.

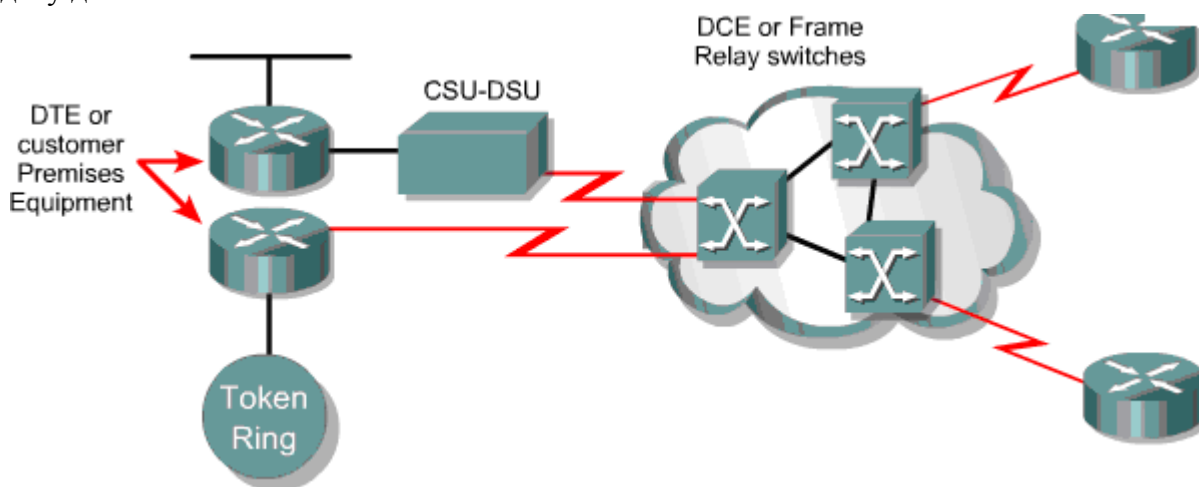


Рисунок 4.

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.

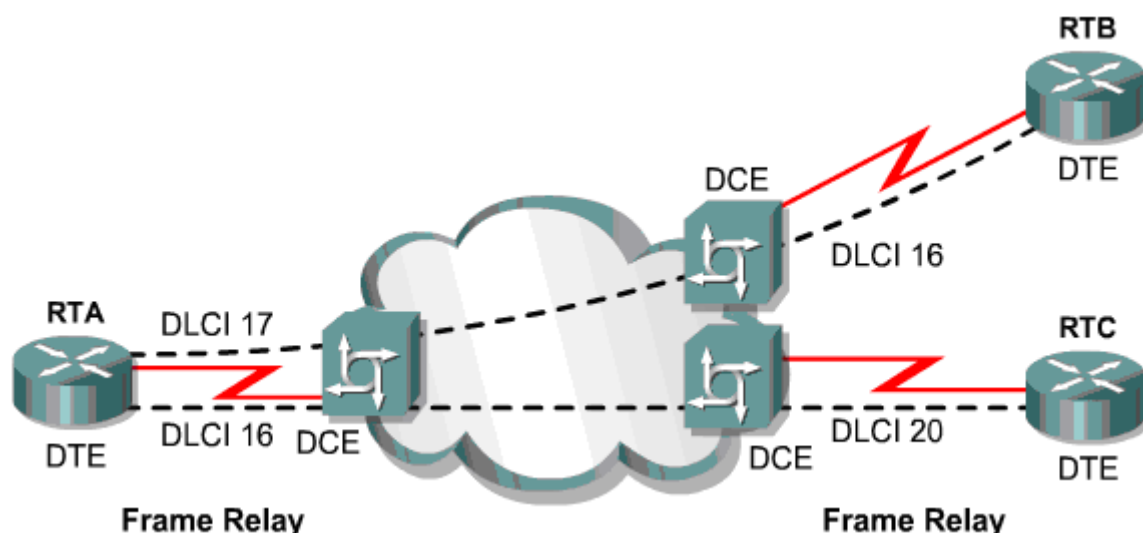


Рисунок 5. Frame Relay DLCI

В сетях Frame Relay такой идентификатор называется DLCI (data-link connection identifier). Они идентифицируют 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.

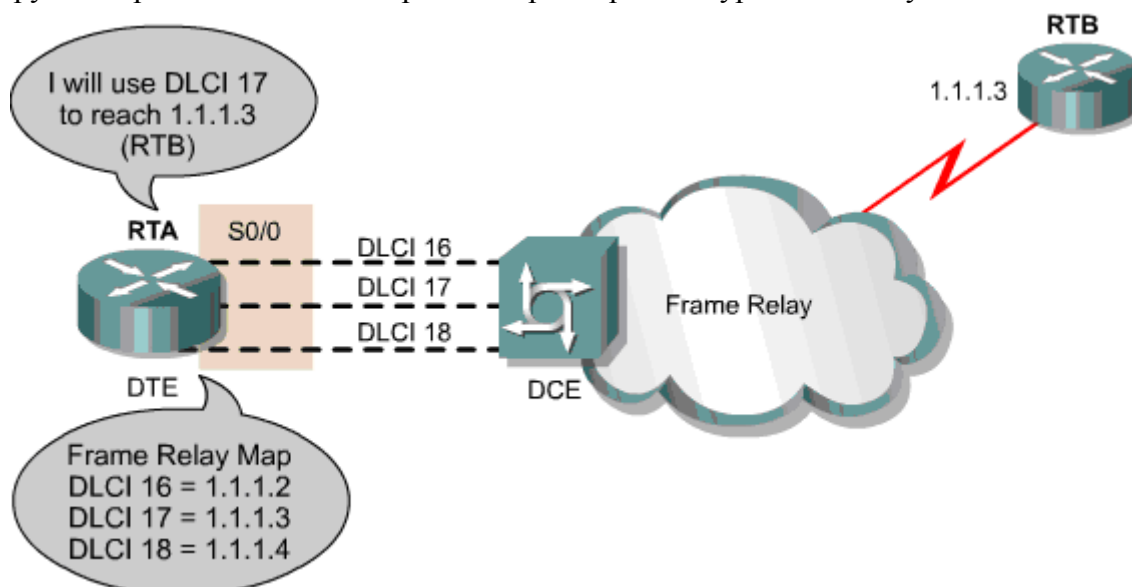


Рисунок 6.

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 широкополосного 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 Serial2/0
```

```
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 не Cisco, 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 сконфигурированного соединения и статистику трафика. Команда показывает статус всех PVC, сконфигурированных 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)

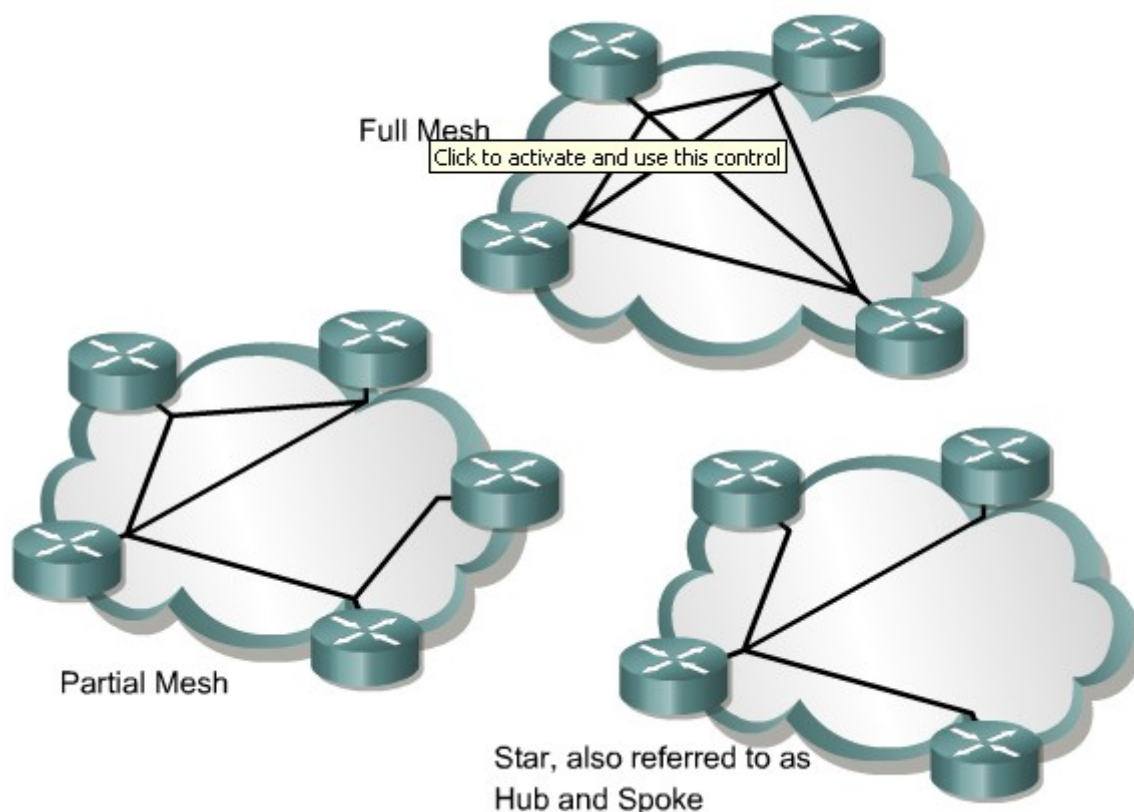


Рисунок 7.

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 физический 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)#interface s2/0
```

```
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 2/0:

```
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 подинтерфейса точка-точка, обычной практикой является номеровать подинтерфейс согласно значению DLCI данного PVC.

После создания подинтерфейса 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 подинтерфейса с DLCI. Эта команда требуется для всех подинтерфейсов точка-точка. Она также требуется для подинтерфейсов multipoint с разрешенным режимом инверсного ARP. Она не требуется для подинтерфейсов 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.

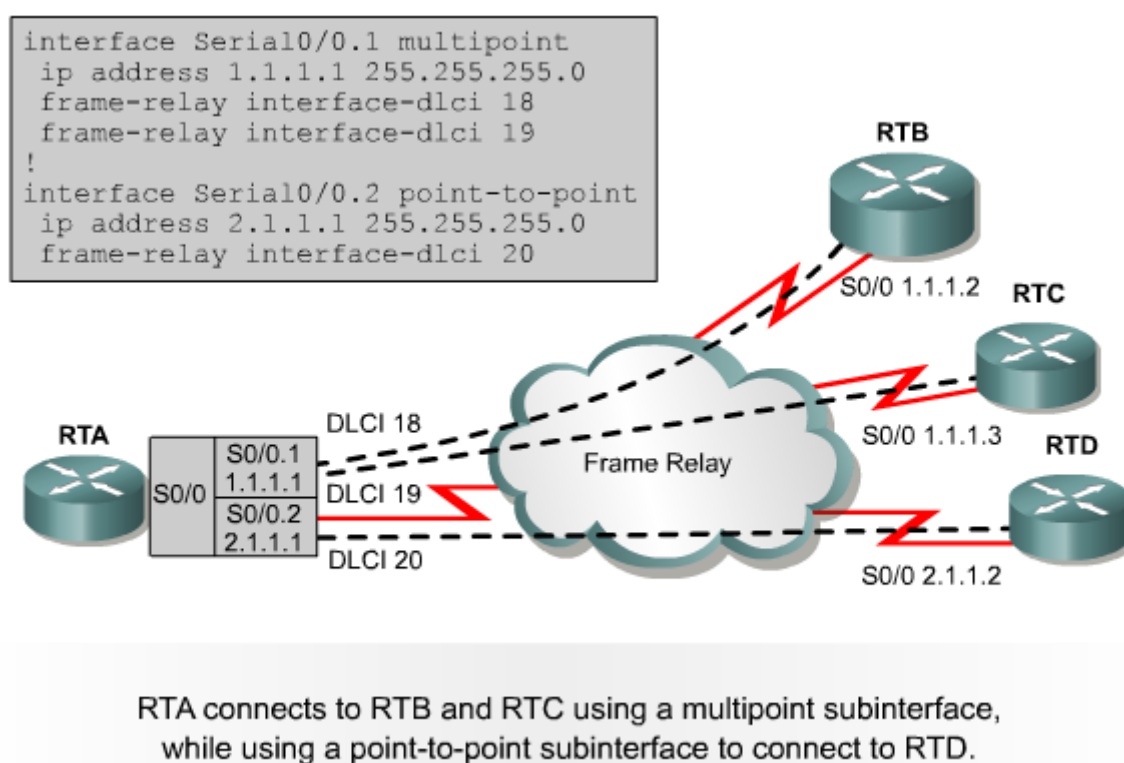


Рисунок 8.

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 маршрутизаторов. 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

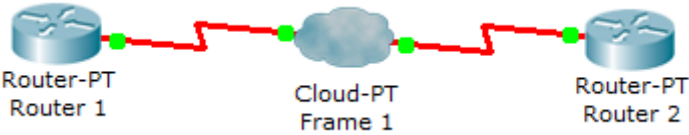


Рисунок 9.

Конфигурируем Frame Relay

TO ROUTER					
FROM ROUTER		Router 1	Router 2		
	Router 1		102		
	Router 2	201			

Рисунок 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 подинтерфейсов

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.

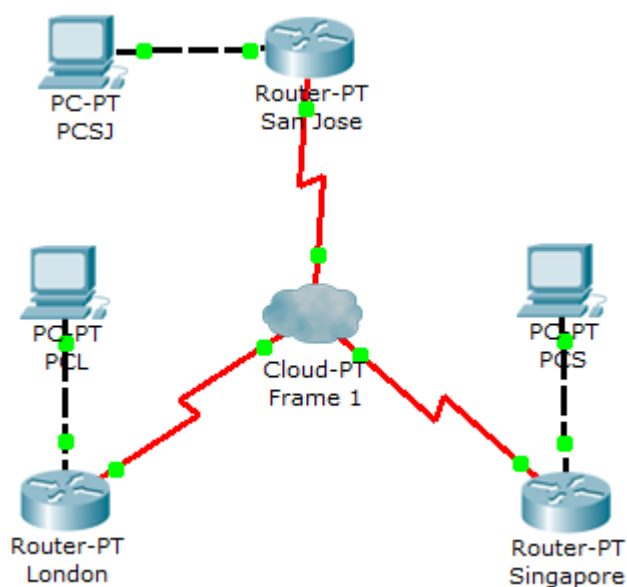


Рисунок 11.

Назначим DLCI согласно рисунку

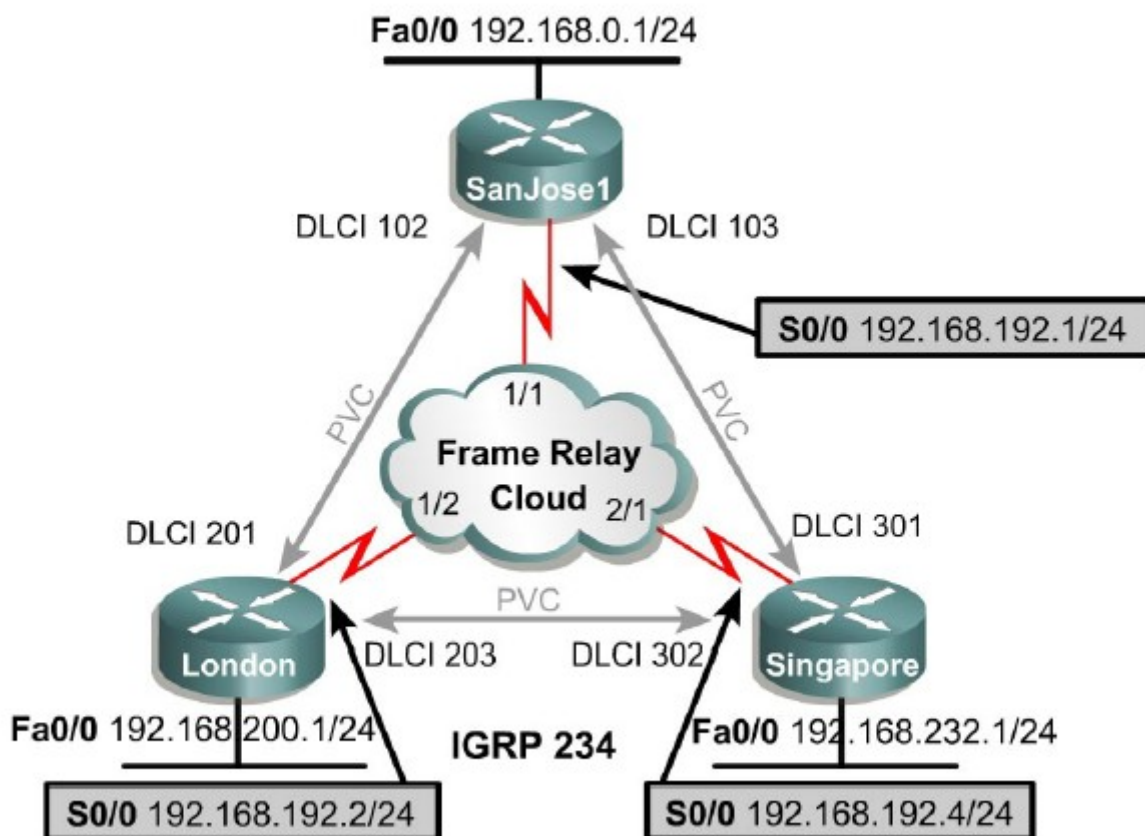


Рис 12.

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 команду “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 Bc

```

е

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 dci 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 dci 102(0x66,0x1860), static,  
              broadcast,CISCO, status defined, active  
Serial0 (up): ip 192.168.192.4 dci 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.

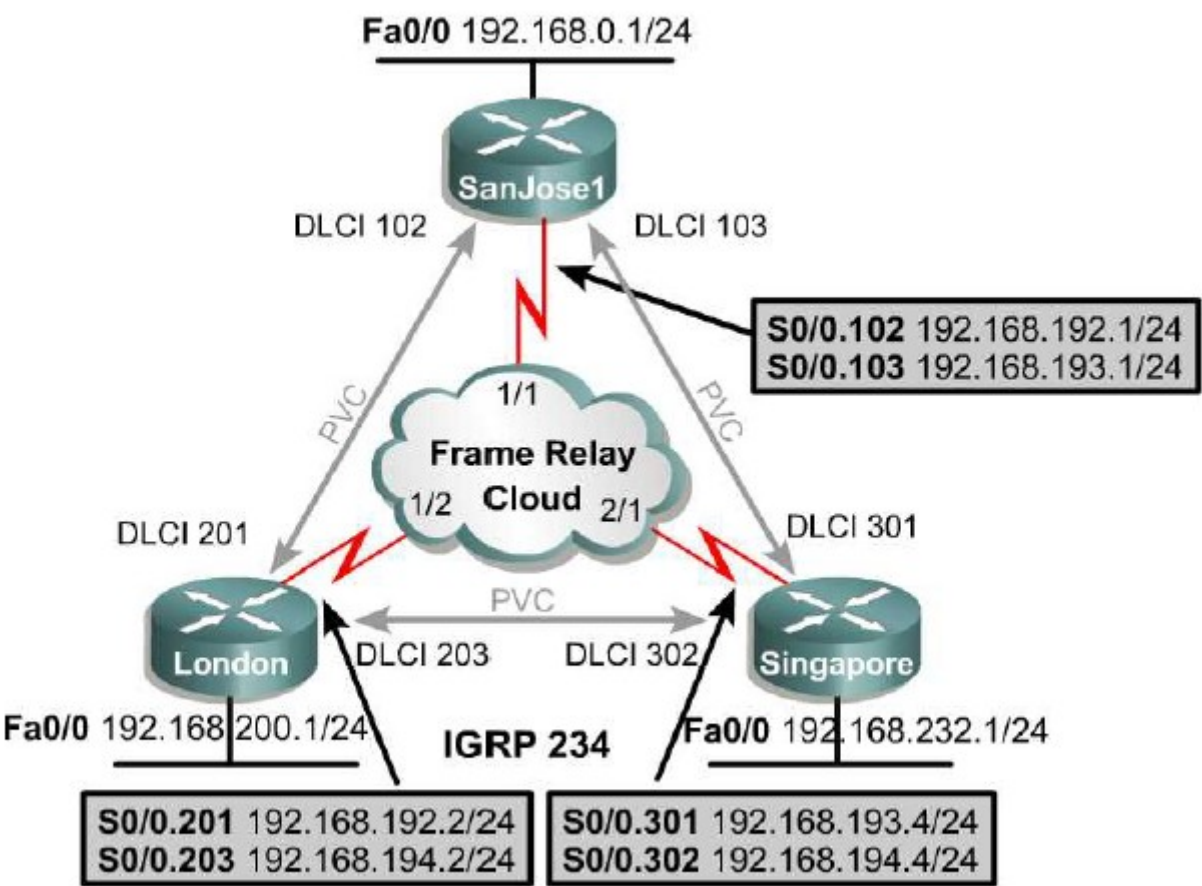


Рисунок 13.

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 подинтерфейс. Например, подинтерфейс 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 подинтерфейс. Подинтерфейс точка-точка может использовать только один DLCI. Если подинтерфейс для DLCI не определён, то DLCI асоциируется с главным интерфейсом `s0`.

Конфигурируем подинтерфейс 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 подынтерфейсы 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 PC.

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	

Рисунок 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. Ометим 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 PC.

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 ATM?
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 конфигурирование 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 Packet Tracer 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 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

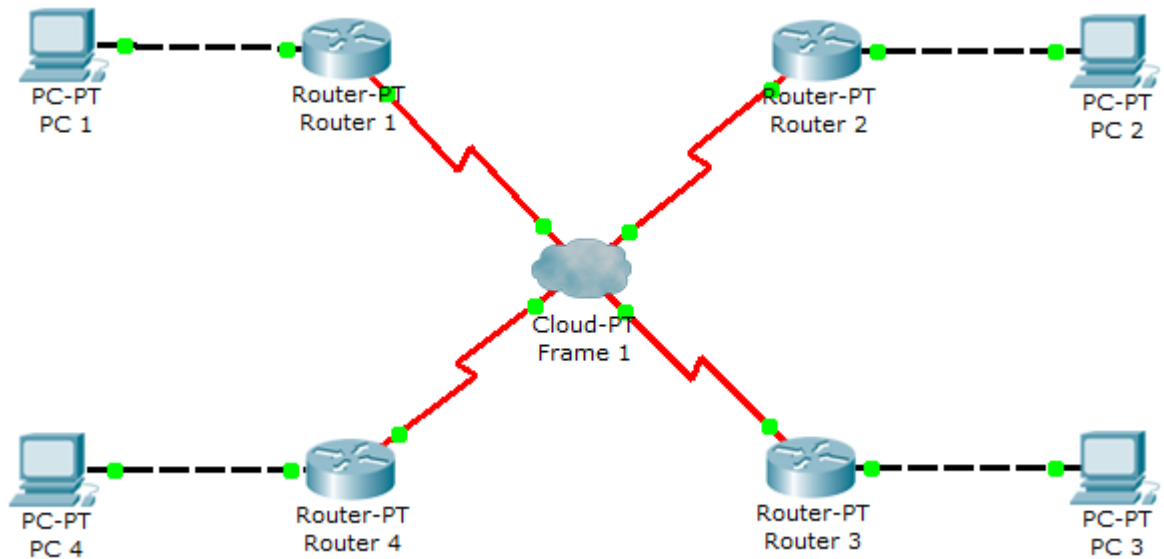


Рисунок 15.

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					
F R O M R O U T E R		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					
F R O M R O U T E R		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:

- 4 таблиц маршрутизаций для каждого маршрутизатора.
- Команды расширенного пинга из любого маршрутизатора для двух произвольных пар компьютеров.
- 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 конфигурационных .txt файлов, созданных при выполнении практической части.
6. Конфигурацию маршрутизаторов из конфигурационных .txt файлов, созданных при выполнении заданий для самостоятельной работы.
7. 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.

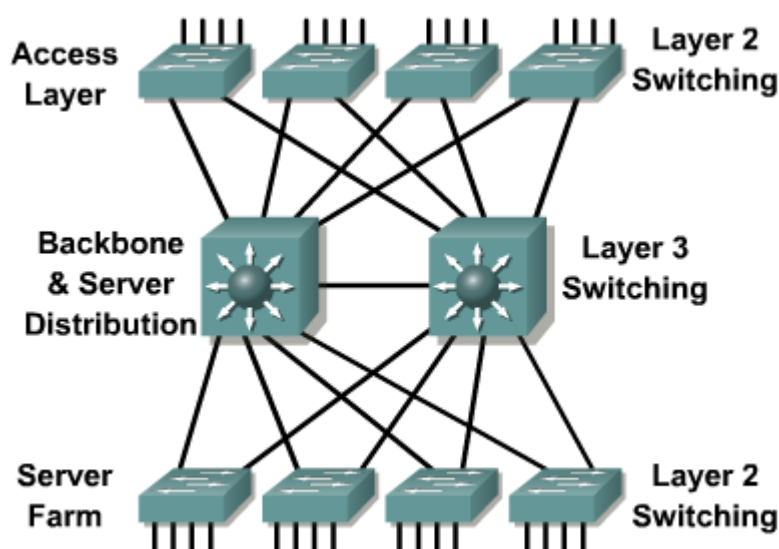


Рисунок 1.

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).

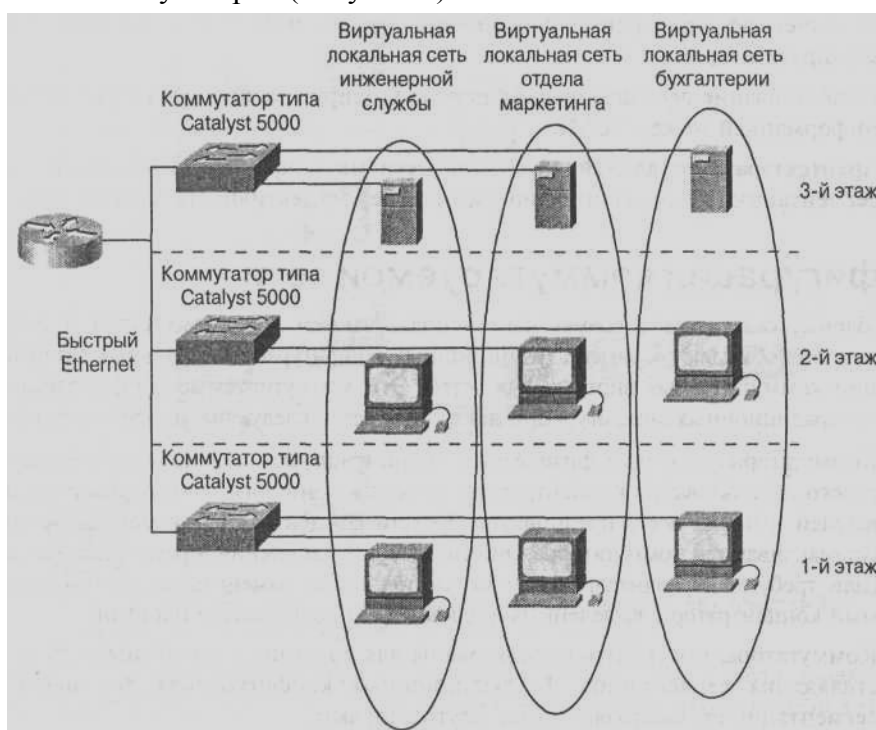


Рисунок 2.

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.

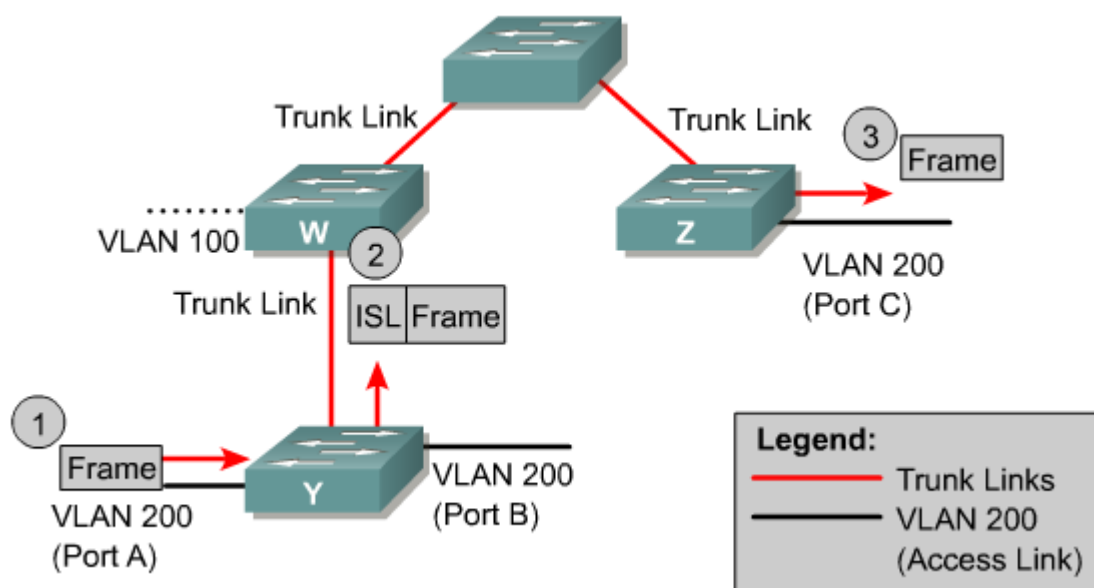


Рисунок 3.

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

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.

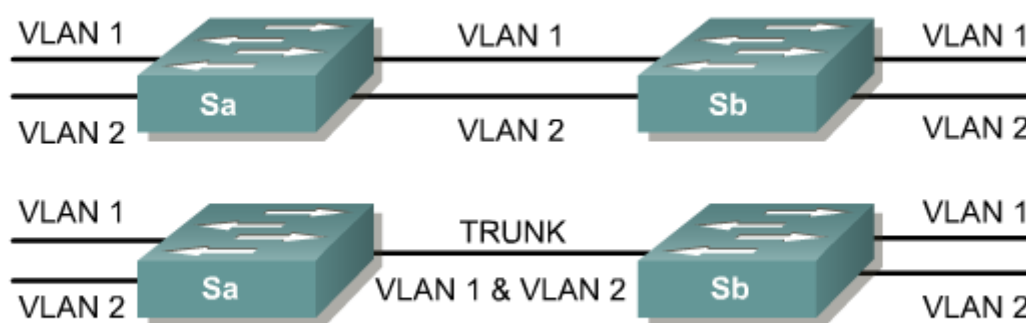


Рисунок 4.

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 связь является 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 направляются. Для такого оформления пакетов используются либо стандарт IEEE 802.1Q либо Cisco протокол ISL (Inter-Switch Link).

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 магистральных 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.

Для создания пустой VLAN с номером №VLAN на коммутаторах Cisco серии 2950 используются команды

```
Switch#vlan database
Switch(vlan)#vlan №VLAN
Switch(vlan)# exit
```

Например, команды

```
Switch#vlan database
Switch(vlan)#vlan 33
Switch(vlan)# 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 магистрали 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 магистрали нет портов нужной VLAN, то полоса пропускания и процессорное время устройств тратится попусту.

Если VLAN не используется на другом конце магистрали, нет нужды разрешать эту VLAN на этой магистрали. 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 магистрали VLAN 3 и затем VLAN с 6 по 10. О том, какие VLAN разрешены на магистрали можно посмотреть командой **show running-config**.

Для удаления большого числа VLANs из магистрали 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 изображенную на рисунке 5. Коммутаторы соедините двумя GigabitEthernet (gi1/1 и gi1/2) 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.

```
Switch1(conf)#interface fa0/2
```

```
Switch1(conf-if)#switchport access vlan 20
Switch1 (conf-if)#interface fa0/3
Switch1(conf-if)#switchport access vlan 30
Switch2 (conf)#interface fa0/2
Switch2(conf-if)#switchport access vlan 20
Switch2 (conf-if)#interface fa0/3
Switch2(conf-if)#switchport access vlan 30
```

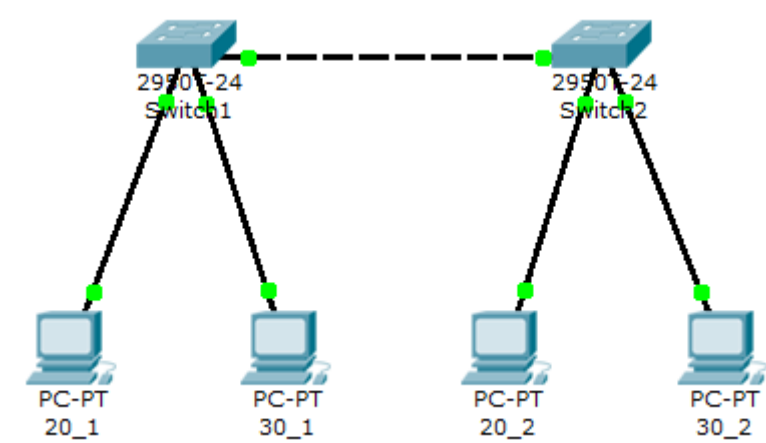


Рисунок 5.

Компьютер	Switch1	Switch2	Адрес
20_1	Fa0/2		172.16.20.1/16
30_1	Fa0/3		172.16.20.2/16
20_2		Fa0/2	172.16.30.1/16
30_2		Fa0/3	172.16.30.2/16
	Gi1/1	Gi1/1	

Таблица 1.

На обоих коммутаторах проверьте результаты создания VLAN, например
Switch1# sh vl name 20

VLAN Name		Status	Ports
20	20	active	Fa0/2

VLAN	Type	SAID	MTU	Parent	RingNo	BridgeNo	Stp	BrdgMode	Trans1	Trans2
20	enet	100020	1500	-	-	-	-	-	0	0

И

Switch1#sh vl br

VLAN Name		Status	Ports
1	default	active	Fa0/1, Fa0/4, Fa0/5, Fa0/6 Fa0/7, Fa0/8, Fa0/9, Fa0/10 Fa0/11, Fa0/12, Fa0/13, Fa0/14 Fa0/15, Fa0/16, Fa0/17, Fa0/18 Fa0/19, Fa0/20, Fa0/21, Fa0/22 Fa0/23, Fa0/24
20	20	active	Fa0/2
30	30	active	Fa0/3
1002	fddi-default	active	
1003	token-ring-default	active	
1004	fddinet-default	active	
1005	trnet-default	active	

Так как нет обмена информации о VLAN между коммутаторами, то компьютеры будут пинговать только самих себя.

Организуем магистрали на коммутаторах. Для этого используем GigabitEthernet порт

Switch1(conf)#**interface gi1/1**

Switch1 (conf-if)#**switchport trunk allowed vlan add 20**

Switch1 (conf-if)#**switchport trunk allowed vlan add 30**

Switch1 (conf-if)#**no shutdown**

Switch2 (conf)#**interface gi1/1**

Switch2 (conf-if)#**switchport trunk allowed vlan add 20**

Switch2 (conf-if)#**switchport trunk allowed vlan add 30**

Switch2 (conf-if)#**no shut**

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 конфигурации коммутаторов.

2. Сохраним топологию предыдущего задания в новом файле. В этой топологии маршрутизатор соединён двумя связями с интерфейсами fa0/1 коммутаторов (рисунок 6). Снова проверьте, что компьютеры в пределах одной VLAN pingуются, а компьютеры, находящиеся в различных VLAN не pingуются.

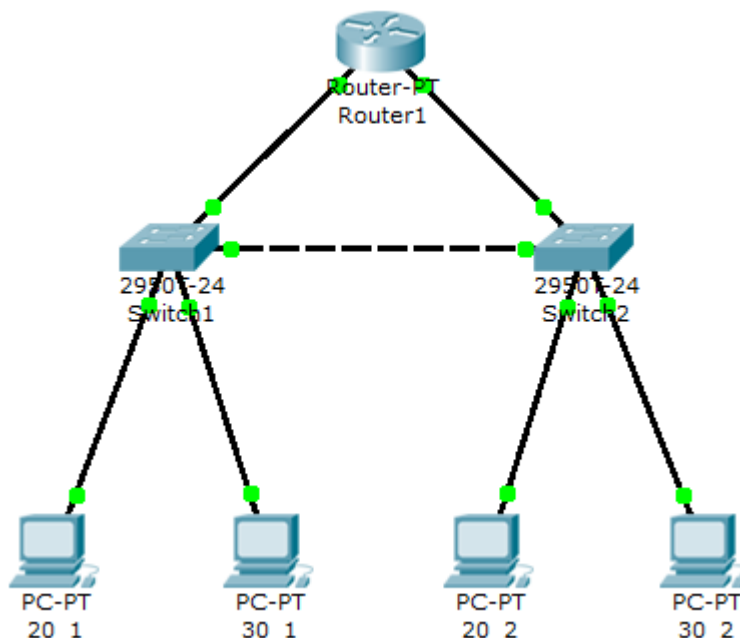


Рисунок 6.

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 pingуются 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

```

Switch1(conf)#interface fa0/1
Switch1(conf-if)#switchport access vlan 20
Switch2(conf)#interface fa0/1
Switch2(conf-if)#switchport access vlan 30
Настроим IP адреса на маршрутизаторе
Router(conf)#interface fa0/0
Router(conf-if)#ip address 172.16.20.254 255.255.255.0
Router(conf-if)#no shutdown
Router(conf-if)#interface fa1/0
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.

Host	Gataway
20 1	172.16.20.254
20 2	172.16.20.254
30 1	172.16.30.254
30 2	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 *gil/2* 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 конфигурациии коммутатора Switch2 установка магистрали на *gil/2* не нужна, хотя она и не мешает.

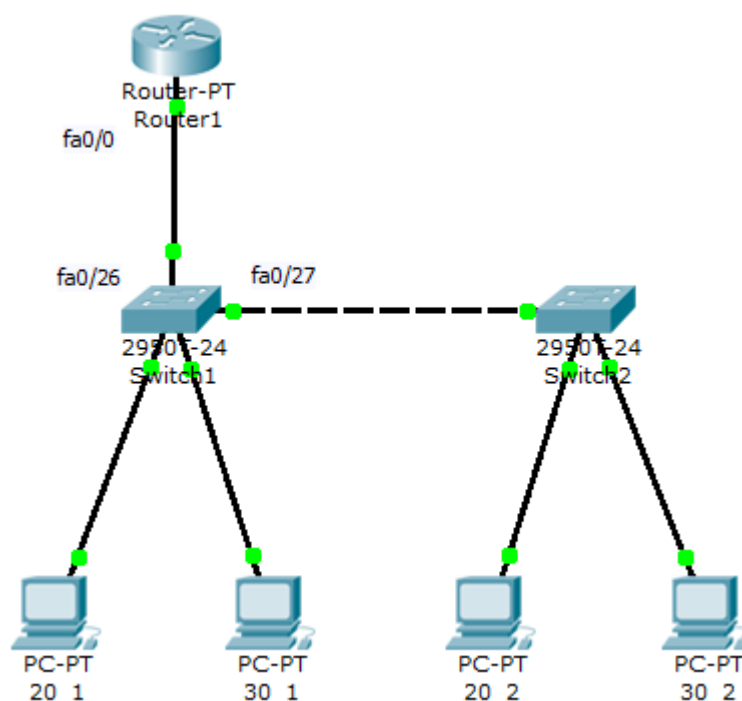


Рисунок 7.

На маршрутизаторе разобьём интерфейс *fa0/0* на два подинтерфейса *fa0/0.20* и *fa0/0.30*. Определим на них инкапсуляцию *dot1q* и поместим их в виртуальные сети 20 и 30, соответственно.

```
Router(conf)#interface FastEthernet0/0.20
Router(conf-subif)#encapsulation dot1q 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 dot1q 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 is directly connected, FastEthernet0/0.20
C      172.16.30.0 is directly connected, FastEthernet0/0.30
```

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 Switch1 к адресу 172.16.30.2 компьютера 30_2 из VLAN 30, подключённого к коммутатору Switch2

```
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 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 Packet Tracer практическую часть.
4. Выполните в Packet Tracer 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 из конфигурационных .txt файлов, созданных при выполнении практической части.

3. Конфигурации всех маршрутизаторов из .txt файлов, созданных при выполнении задания для самостоятельной работы.
5. Все скриншоты, указанные в задании для самостоятельной работы