

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет радиотехники и электроники

Кафедра информационных радиотехнологий

Н. И. Листопад, Е. Н. Каленкович

***СИСТЕМЫ И СЕТИ ПЕРЕДАЧИ ДАННЫХ. ЗАЩИТА
ИНФОРМАЦИИ В КОМПЬЮТЕРНЫХ СЕТЯХ.
ЛАБОРАТОРНЫЙ ПРАКТИКУМ***

*Рекомендовано УМО по образованию в области информатики
и радиоэлектроники в качестве учебно-методического пособия
для специальностей 1-39 01 01 «Радиотехника (по направлениям)»,
1-39 01 03 «Радиоинформатика»*

Минск БГУИР 2020

УДК [004.738.5+004.056.5](076.5)

ББК 32.971.35я73+32.972.5я73

Л63

Р е ц е н з е н т ы:

кафедра телекоммуникационных систем учреждения образования
«Белорусская государственная академия связи»
(протокол №8 от 05.03.2019);

начальник центра информационных технологий
Белорусского государственного университета
кандидат технических наук, доцент В. П. Кочин

Листопад, Н. И.

Л63 Системы и сети передачи данных. Защита информации в компьютерных сетях. Лабораторный практикум : учеб.-метод. пособие / Н. И. Листопад, Е. Н. Каленкович. – Минск : БГУИР, 2020. – 114 с. : ил.
ISBN 978-985-543-539-7.

Включает методические материалы для практического выполнения лабораторных работ по курсу «Системы и сети передачи данных. Защита информации в компьютерных сетях». Издание направлено на получение студентами начальных знаний и приобретение умений проектирования и настройки систем и сетей передачи данных.

УДК [004.738.5+004.056.5](076.5)

ББК 32.971.35я73+32.972.5я73

ISBN 978-985-543-539-7

© Листопад Н. И., Каленкович Е. Н., 2020

© УО «Белорусский государственный университет информатики и радиоэлектроники», 2020

СОДЕРЖАНИЕ

Лабораторная работа №1. Основы компьютерных сетей.....	4
Лабораторная работа №2. IP-адресация	26
Лабораторная работа №3. Протокол передачи файлов FTP.....	38
Лабораторная работа №4. Виртуальные локальные сети.....	53
Лабораторная работа №5. Маршрутизация трафика с использованием коммутаторов третьего уровня.....	70
Лабораторная работа №6. Статическая маршрутизация	80
Лабораторная работа №7. Протокол динамической настройки узла DHCP и система доменных имен DNS.....	94
Список использованных источников.....	113

Лабораторная работа №1

ОСНОВЫ КОМПЬЮТЕРНЫХ СЕТЕЙ

Цель работы:

- ознакомиться с типами компьютерных сетей и основными сетевыми топологиями;
- изучить варианты доступа к среде передачи и основные типы используемых линий связи;
- ознакомиться с сетевым оборудованием и изучить взаимодействие компьютеров в сети.

1.1. Краткие теоретические сведения

Сеть (Network) – группа компьютеров и(или) других устройств, соединенных каким-либо способом для обмена информацией и совместного использования ресурсов.

Ресурсы – программы, файлы данных, а также принтеры и другие совместно используемые периферийные устройства в сети.

Компьютерные сети можно классифицировать по ряду признаков.

В зависимости от расстояния между связываемыми узлами сети выделяют следующие типы компьютерных сетей:

- локальные сети (LAN – Local Area Network);
- региональные сети (MAN – Metropolitan Area Network);
- глобальные сети (WAN – Wide Area Network).

Локальная вычислительная сеть (ЛВС) – небольшая группа компьютеров, связанных друг с другом и расположенных обычно в пределах одного здания или организации.

Региональная сеть – сеть, соединяющая множество локальных сетей в рамках одного города, района, региона.

Глобальная сеть – сеть, объединяющая компьютеры разных городов, регионов и государств.

Объединение глобальных, региональных и локальных вычислительных сетей позволяет создавать многоуровневые иерархии, представляющие собой мощные средства для обработки больших массивов данных и доступа к практически неограниченным информационным ресурсам.

По типу среды передачи выделяют:

- проводные сети – с использованием коаксиальных кабелей, витой пары или же оптического волокна;
- беспроводные сети – когда передача данных производится по радиоканалам или в инфракрасном диапазоне волн.

По скорости передачи информации компьютерные сети бывают:

- низкоскоростные – скорость передачи данных до 10 Мбит/с;
- среднескоростные – скорость передачи данных от 10 до 100 Мбит/с;

– высокоскоростные – скорость передачи данных составляет свыше 100 Мбит/с.

С точки зрения архитектуры все компьютерные сети подразделяют на одноранговые и иерархические.

В *одноранговых сетях* (или Peer-to-Peer Network) все компьютеры равноправны между собой, т. е. в компьютерной сети нет единого центра управления и каждый из компьютеров сети может выступать как в роли сервера (предоставлять другим компьютерам в сети доступ к своим программным и аппаратным ресурсам), так и в роли клиента, использующего ресурсы других компьютеров.

Одноранговые сети имеют свои преимущества и недостатки. К преимуществам можно отнести следующие:

- простота установки и настройки, а также низкая стоимость развертывания и поддержки;
- независимость отдельных компьютеров и их ресурсов друг от друга;
- возможность контроля ресурсов компьютера со стороны пользователя;
- отсутствие необходимости в дополнительном программном обеспечении (за исключением операционной системы);
- отсутствие необходимости в постоянном присутствии системного администратора.

Кроме достоинств, у одноранговых сетей можно выделить ряд недостатков:

- отсутствие возможности централизованного управления сетью;
- эффективность работы сети зависит от числа компьютеров;
- общая низкая защищенность сети, которая зависит от настроек каждого компьютера, входящего в сеть;
- необходимость выполнения процедуры резервного копирования данных на каждом из компьютеров в отдельности.

Некоторые отмеченные недостатки одноранговых сетей можно устранить, если в компьютерной сети выделить один или несколько компьютеров, которые будут хранить общие данные, используемые всеми компьютерами сети, и обрабатывать поток запросов от других компьютеров, т. е. перейти к *иерархической* архитектуре сети или же сети с выделенным сервером, или клиент-серверной сети (Dedicated Server Network).

В иерархической сети компьютеры, выделенные для хранения информации и обработки запросов пользователей и называемые серверами, решают задачи по обработке большого числа запросов других компьютеров, называемых клиентами. При этом запросы пользователей могут иметь различную сложность: от самых простых – проверки имени и пароля пользователя при входе в систему – до сложных запросов поиска по имеющимся базам данных.

В роли серверов обычно выступают отдельные более производительные рабочие станции в сравнении с компьютерами клиентов. Они могут оснащаться дополнительным оборудованием в виде высокоскоростных сетевых адаптеров или дисковых накопителей, объединенных в RAID-массивы, способных хранить большой объем информации и имеющих высокую скорость чтения и записи.

Серверы в своем большинстве работают в круглосуточном режиме, предоставляя доступ к своим ресурсам, и обеспечивают доступ пользователям к своим службам.

Основными преимуществами использования иерархических сетей являются:

- использование мощного серверного оборудования позволяет обеспечить быстрый доступ к ресурсам и эффективную обработку запросов клиентов;
- четкое управление информацией и пользовательскими данными за счет централизации ресурсов;
- упрощение процедуры резервного копирования за счет размещения данных на сервере, а не на многих пользовательских компьютерах;
- высокая общая защищенность сети и сохранность данных.

К недостаткам иерархических сетей можно отнести следующие:

- возможные неисправности сервера могут сделать всю сеть неработоспособной, а данные – недоступными;
- сложность развертывания и технической поддержки;
- увеличенная стоимость сопровождения сети за счет необходимости выделенного оборудования и специализированного программного обеспечения;
- требуется наличие как минимум одного постоянно присутствующего системного администратора для управления сетью.

Для взаимодействия компьютеров друг с другом в сети необходимо каким-либо образом соединить между собой все оборудование, входящее в сеть – серверы, рабочие станции пользователей, ноутбуки, карманные компьютеры, принтеры, хранилища данных и т. д. Для этого используются различные решения в виде сетевых кабелей различного типа, оптоволокна, радиоканалов.

Помимо установления физического соединения между всеми устройствами сети требуется также обеспечить взаимодействие устройств между собой. Это делается при помощи набора протоколов и интерфейсов передачи данных.

Взаимодействие различных устройств внутри компьютерной сети принято соотносить с определенной эталонной моделью взаимодействия открытых систем. В целях формализации процесса взаимодействия открытых систем в 1984 г. Международная организация по стандартизации ISO (International Standards Organization) выпустила ряд спецификаций, названных эталонной моделью взаимодействия открытых систем OSI (Open Systems Interconnection). Модель OSI стала международным стандартом для построения сетей различных типов.

Модель OSI стандартизирует:

- понятия и основные термины, используемые при построении открытых систем;
- уровни взаимодействия систем в сетях с коммутацией пакетов;
- стандартные названия уровней;
- функции, которые должен выполнять каждый уровень.

Протокол – это набор формализованных правил и процедур, регулирующих порядок обмена информацией на одном уровне сетевой модели, но между различными устройствами в сети.

Интерфейс – это набор формализованных правил, по которым производится обмен информацией между соседними уровнями одного устройства.

В сущности, интерфейс и протокол выражают одно и то же понятие, но протоколы определяют правила взаимодействия модулей одного уровня в разных узлах, а интерфейсы – модулей соседних уровней в одном узле.

Иерархически организованный набор протоколов, достаточный для организации взаимодействия узлов в сети, называют *стеком коммуникационных протоколов*.

Коммуникационные протоколы могут быть реализованы как программно, так и аппаратно. Протоколы нижних уровней часто реализуются комбинацией программных и аппаратных средств, а протоколы верхних уровней – как правило, чисто программными средствами. Программный модуль, реализующий некоторый протокол, часто тоже называют протоколом.

Модель OSI имеет вертикальную структуру, в которой все сетевые функции распределены между сетевыми уровнями (рис. 1.1). Каждому уровню соответствуют строго определенные операции, оборудование и протоколы.

Реальное взаимодействие уровней (передача информации внутри одного компьютера) возможно только по вертикали и только с соседними уровнями.

Логическое взаимодействие (в соответствии с правилами того или иного протокола) осуществляется по горизонтали – с аналогичным уровнем другого компьютера на противоположном конце линии связи. Каждый более высокий уровень пользуется услугами нижележащего уровня, зная, в каком виде и каким способом (через какой интерфейс) можно передать ему данные.

Задача более низкого уровня – принять данные, добавить свою информацию (форматирующую, адресную, которая необходима для правильного взаимодействия с аналогичным уровнем на другом компьютере) и передать данные дальше. Дойдя до физического уровня сетевой модели, информация попадает в среду передачи и достигает компьютера-получателя. В нем она проходит через все уровни в обратном порядке, пока не достигнет такого же уровня, с которого она была послана компьютером-отправителем.

Модель OSI включает семь уровней: прикладной, представительный, сеансовый, транспортный, сетевой, канальный, физический.

Самым верхним уровнем модели является прикладной, самый нижним – физический.

Обмен данными происходит путем их перемещения с верхнего уровня на нижний, а также транспортировки по сети и обратного воспроизведения на компьютере-получателе с нижнего уровня на верхний.

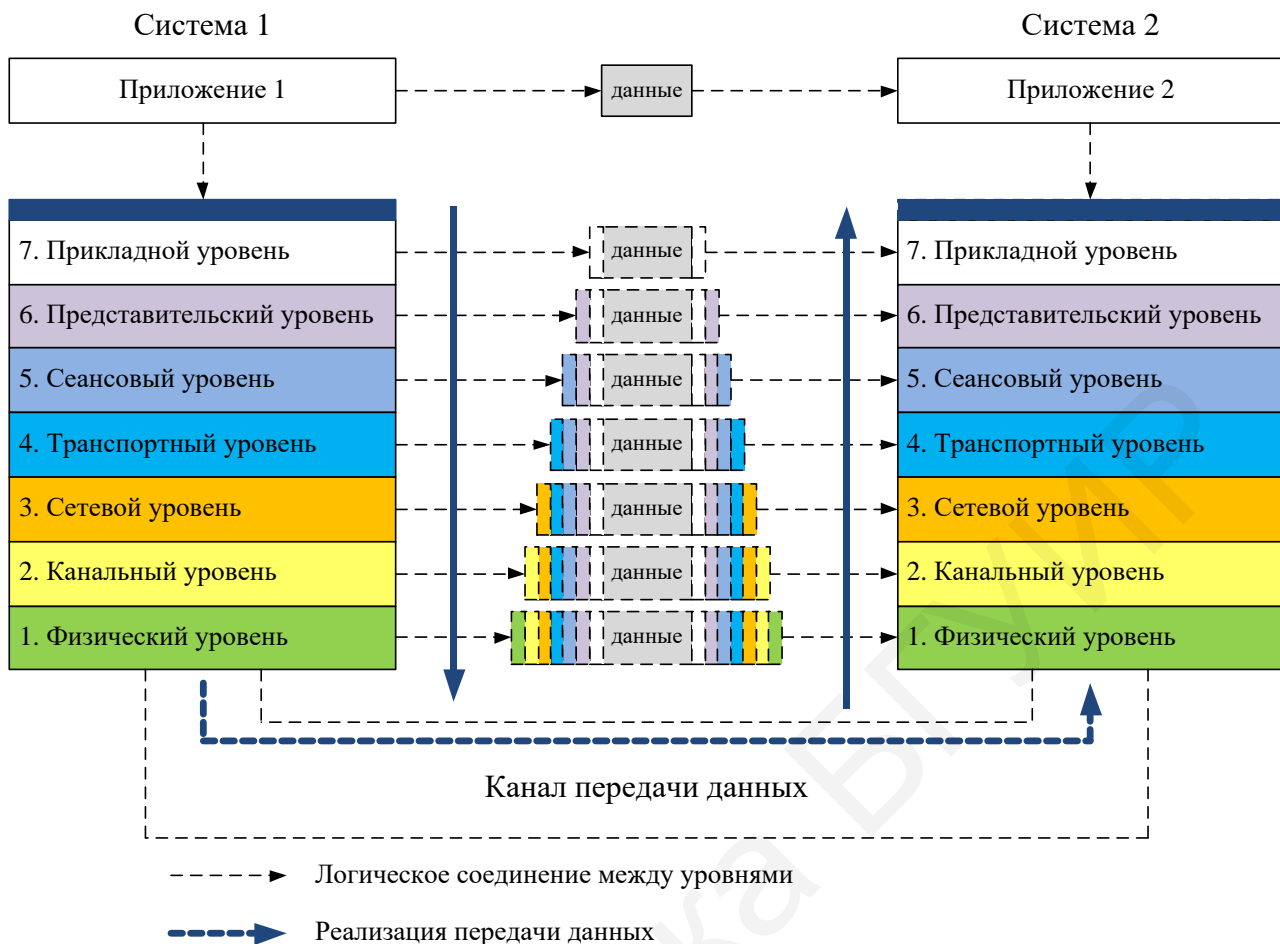


Рис. 1.1. Модель OSI

При этом на каждом уровне к исходному сообщению, которое надо передать по сети, добавляется заголовок данного уровня, содержащий служебную информацию, необходимую для передачи. На компьютере-получателе каждый уровень, в свою очередь, анализирует соответствующий ему заголовок, выполняет нужные функции, а затем удаляет этот заголовок и передает сообщение вышележащему уровню. Таким образом, с физической точки зрения вся информация проходит два раза через все уровни модели.

Рассмотрим уровни модели OSI.

Первым уровнем модели определен *физический уровень* (Physical Layer). На этом уровне осуществляется передача неструктурированного потока битов, полученных от вышестоящего канального уровня, через физическую среду. Физический уровень отвечает за поддержание связи и детально описывает электрические, оптические, механические и функциональные интерфейсы взаимодействия с физической средой передачи данных: напряжения, частоты, длины волн, типы разъемов, число и функциональное назначение контактов, схемы кодирования сигналов и др. Единицей передаваемой информации на этом уровне является 1 бит.

Второй уровень модели – *канальный*, или *уровень передачи данных* (Data Link Layer). Он обеспечивает безошибочную передачу данных, полученных от вышестоящего сетевого уровня, через физический уровень, который сам по себе отсутствия ошибок не гарантирует и может искажать данные. Информация на этом уровне помещается в кадры.

При получении данных на канальном уровне определяются начало и конец кадра в потоке битов, сам кадр извлекается из потока и проверяется на наличие ошибок. Поврежденные при передаче кадры, а также кадры, для которых не получено подтверждение о приеме, ретранслируются (пересылаются заново). На канальном уровне обеспечивается управление доступом к среде передачи.

Канальный уровень часто разбивают на два подуровня:

- подуровень управления доступом к среде (Media Access Control, MAC);
- подуровень управления логической связью (Logical Link Control, LLC).

Уровень MAC обеспечивает совместный доступ сетевых адаптеров к физическому уровню, определение границ кадров, распознавание адресов назначения кадров (физические, или MAC-адреса).

Уровень LLC (действует над уровнем MAC) отвечает за установление канала связи, безошибочную посылку и прием сообщений с данными и обеспечение заданного качества обслуживания QoS (Quality of service).

Третий уровень – *сетевой* (Network Layer). Отвечает за обеспечение связи между любыми точками в компьютерной сети. Этот уровень осуществляет передачу информационных сообщений по сети, которая может состоять из множества отдельных сетей, соединенных множеством линий связи. Такая доставка требует маршрутизации, т. е. определения пути доставки сообщения, а также решения задач управления потоками данных и обработки ошибок передачи. Таким образом, главная функция сетевого уровня – маршрутизация информационных потоков.

Четвертый уровень модели – *транспортный* (Transport Layer). Гарантирует доставку информации от одного компьютера другому. На этом уровне компьютера-отправителя большие блоки данных разбиваются на более мелкие пакеты, которые доставляются компьютеру-получателю в нужной последовательности без потерь и дублирования. На транспортном уровне компьютера-получателя пакеты вновь собираются в исходные блоки данных. Таким образом, транспортный уровень завершает процесс передачи данных, скрывая от более высоких уровней все детали и проблемы, связанные с доставкой информации любого объема между любыми точками во всей сети.

Пятый уровень – *сеансовый* (Session Layer). Позволяет двум сетевым приложениям на разных компьютерах устанавливать, поддерживать и завершать соединение, называемое сетевым сеансом. Этот уровень также отвечает за восстановление аварийно прерванных сеансов связи. Кроме того, на пятом уровне выполняется преобразование удобных для пользователей имен компьютеров в сетевые адреса (распознавание имен), а также реализация функции защиты сеанса.

Шестой уровень – *представительский*, или *уровень представления данных* (Presentation Layer). Определяет форматы передаваемой между компьютерами

информации. Здесь решаются такие задачи, как перекодировка (перевод информации в вид, понятный для всех участвующих в обмене компьютеров), сжатие и распаковка данных, шифрование и дешифровка, поддержка сетевых файловых систем и др.

Седьмой уровень – *прикладной*, или уровень приложений (Application Layer). Обеспечивает интерфейс взаимодействия программ, работающих на компьютерах в сети. Именно с помощью этих программ пользователь получает доступ к таким сетевым услугам, как электронная почта, удаленный терминальный доступ и т. д.

Уровни модели OSI можно разделить на две группы:

- сетезависимые уровни, зависящие от конкретной технической реализации сети;
- сетенезависимые уровни, ориентированные на работу с приложениями.

К сетезависимым относятся три нижних уровня: сетевой, канальный, физический. К сетенезависимым относятся: прикладной, представительный, сеансовый. Транспортный уровень занимает промежуточное положение между нижними и верхними уровнями и отвечает главным образом за установку и разрыв соединения.

Исходя из приведенного деления уровней, можно сделать вывод, что от организации физических связей между компьютерами будет зависеть набор используемых протоколов и интерфейсов для обеспечения функционирования сети. Компьютеры, объединяемые в сети, подключаются к ним не хаотично, а в определенном порядке, определяемом топологией сети.

Сетевая топология – это схема расположения и соединения сетевых устройств. Сетевая топология может быть представлена различными видами: физической, логической, информационной.

Физическая топология сети описывает расположение компьютеров и сетевого оборудования в реальном мире.

Логическая топология сети показывает прохождение сигналов в рамках физической топологии.

Информационная топология – описывает прохождение перемещения, направления и перенаправления потоков информации, передаваемых по компьютерной сети.

При проектировании сети передачи данных необходимо выбирать такую топологию, которая будет обеспечивать надежную и эффективную работу сети, а также удобное управление потоками данных. Также желательно, чтобы сеть по стоимости создания и сопровождения получилась недорогой, но в то же время оставались возможности для ее дальнейшего расширения.

Рассмотрим основные типы топологий организации компьютерных сетей по физической структуре.

Базовыми сетевыми топологиями организации физических связей между элементами компьютерной сети являются:

- полносвязная топология;

- топология «общая шина»;
- топология «звезда»;
- топология «кольцо»;
- ячеистая топология;
- смешанная топология.

Полносвязная топология представляет собой сеть передачи данных, в которой каждый компьютер в сети непосредственно соединен со всеми остальными. Данный тип сетей представляет очень громоздкий и неэффективный вариант в силу наличия большого количества физических соединений между элементами сети. Неоспоримым достоинством этой топологии является то, что каждый компьютер или же другое сетевое устройство может напрямую обратиться к любому из устройств.

Топология типа «общая шина» представляет собой общий кабель (называемый шиной или магистралью), к которому подключаются все устройства в сети (рис. 1.2).



Рис. 1.2. Топология сети типа «общая шина»

Данные от компьютера или другого сетевого устройства могут передаваться по шине в обе стороны. Ввиду того что шина является длинной линией, то на концах возможно появление отраженных волн, что приводит к нарушению передачи данных. Для устранения отражения сигналов на концах используемого кабеля устанавливают согласующие элементы в виде резисторов с сопротивлением, равным волновому сопротивлению используемого кабеля. Такие элементы также называют терминаторами.

Таким образом, информация поступает на все устройства, но принимается лишь тем, которому она адресована.

Данная топология применялась в локальных сетях с архитектурой Ethernet 10Base-5 и 10Base-2, основанных на применении «толстого» и «тонкого» коаксиальных кабелей соответственно.

Достоинствами данной сетевой топологии являются:

- уменьшение длины используемого кабеля для основной магистрали;
- выход из строя одного или нескольких устройств не влияет на работоспособность сети в целом;
- простота в настройке и конфигурации сети.

Однако топология «общая шина» имеет ряд существенных недостатков, которые ограничивают сферу ее применения:

- разрыв магистрального кабеля вызывает нарушение работы всей сети;
- ограниченность в количестве подключаемых устройств и длине используемых кабелей;
- низкая производительность сети, вызванная тем, что все устройства должны делить между собой один канал связи.

Топология типа «звезда» представляет собой топологию сети, где каждый компьютер соединен с центральным узлом. В качестве центрального узла может выступать концентратор (Hub) или же центральный компьютер (сервер). В первом случае мы имеем топологию типа «пассивная звезда» (Star-Bus) (рис. 1.3), во втором – топологию «активная звезда» (Active Star) (рис. 1.4).

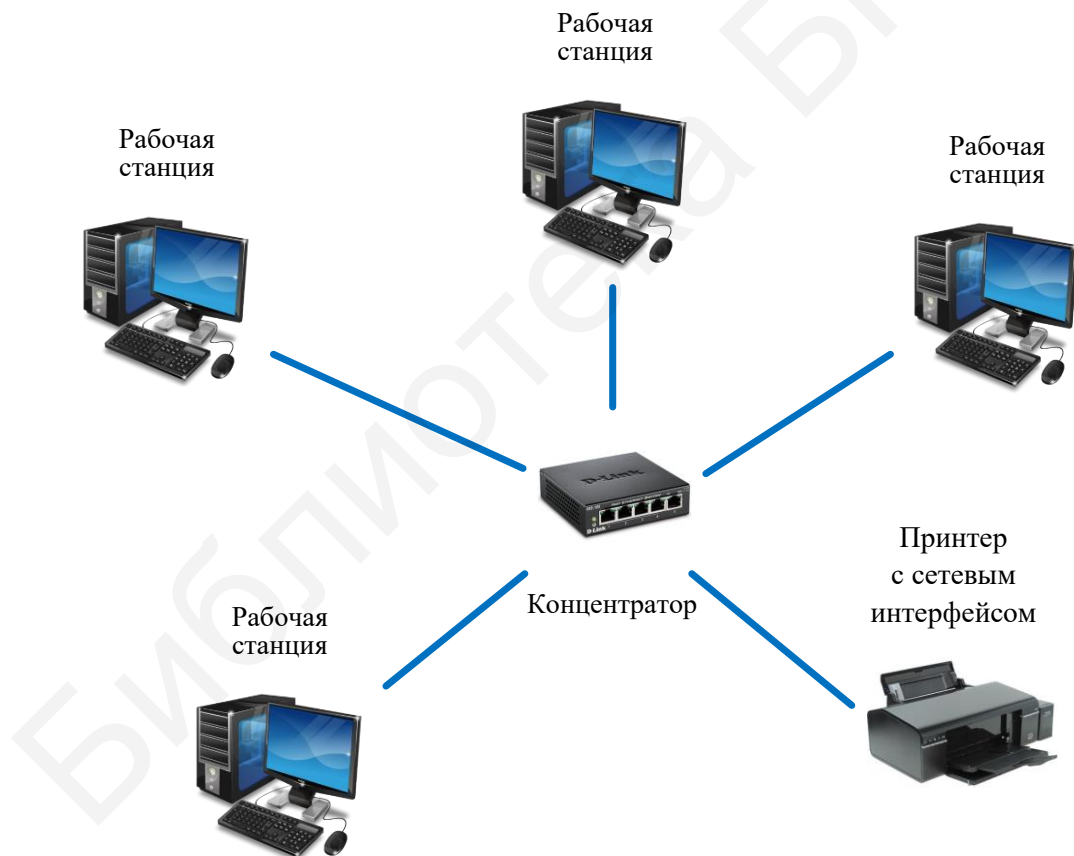


Рис. 1.3. Пример топологии «пассивная звезда»

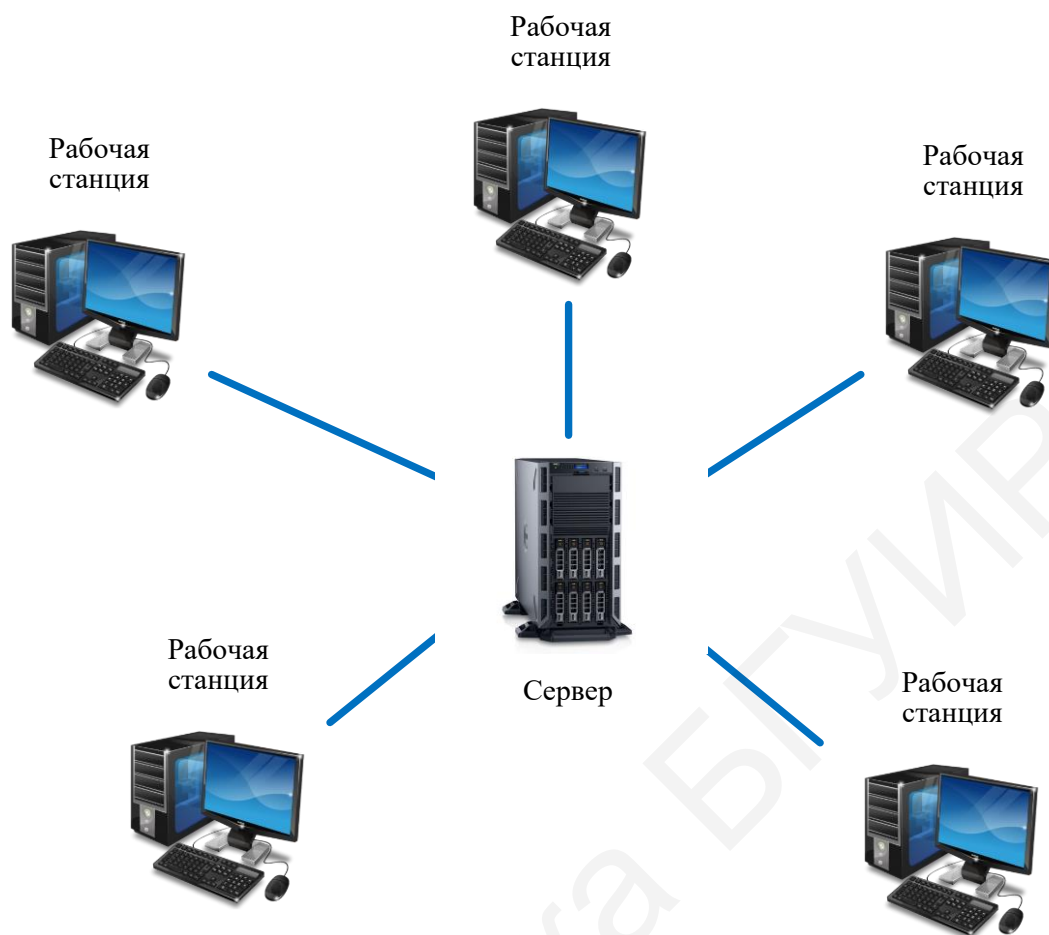


Рис. 1.4. Пример топологии «активная звезда»

При использовании топологии «пассивная звезда» все компьютеры подключаются к концентратору, который никак не отвечает за управление обменом данными, а лишь выполняет функцию повторителя, т. е. восстанавливает приходящие сигналы и пересылает их всем остальным подключенным к нему компьютерам и устройствам.

В топологии «активная звезда» все устройства сети подключаются к центральному компьютеру. В такой конфигурации все потоки данных идут исключительно через центральный компьютер; он же полностью отвечает за управление информационным обменом между всеми участниками сети. Конфликты при такой организации сети невозможны, однако нагрузка на центральный компьютер столь велика, что ничем другим, кроме обслуживания сети, этот компьютер не занимается. Выход его из строя приводит к отказу всей сети.

При использовании вместо концентраторов коммутаторов и маршрутизаторов получается промежуточный тип топологии между активной и пассивной звездой. В этом случае устройства связи не только ретранслируют поступающие сигналы, но и производят управление их обменом.

Достоинствами топологии типа «звезда» являются:

- надежность – подключение к центральному концентратору и отключение концентраторов от него никак не отражается на работе остальной сети; обрывы кабеля влияют только на единичные компьютеры;

- легкость при обслуживании и устранении проблем – все компьютеры и сетевые устройства подключаются к центральному соединительному устройству, что существенно упрощает обслуживание и ремонт сети;

- защищенность – концентрация точек подключения в одном месте позволяет легко ограничить доступ к жизненно важным объектам сети.

К недостаткам топологии можно отнести большой расход кабеля при организации сети и выход из строя всей сети при отказе центрального коммутирующего устройства.

Сетевая топология типа «кольцо». При такой организации сети каждый из компьютеров или сетевых устройств подключается последовательно от одного к другому, образуя неразрывное кольцо, по которому передаются данные (рис. 1.5). Данные в кольце передаются, как правило, в одном и том же направлении.

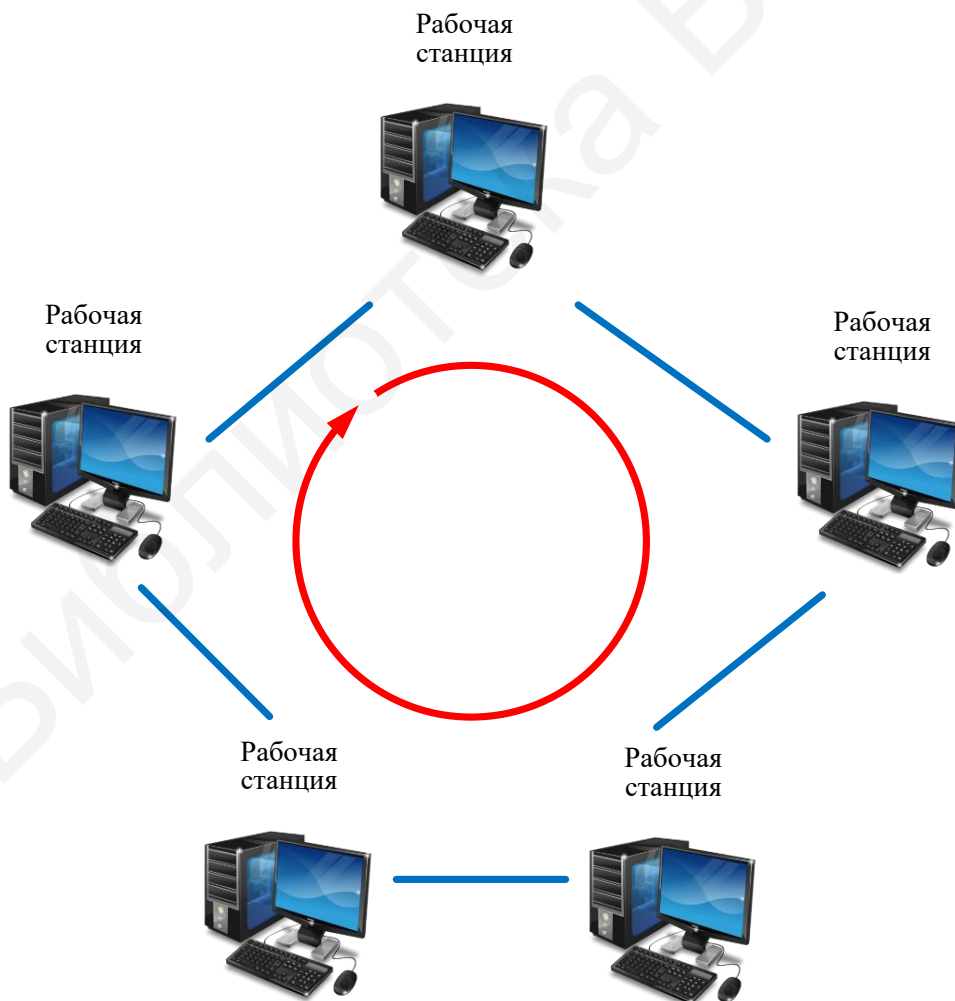


Рис. 1.5. Топология типа «кольцо»

Преимущества данной топологии:

- каждый из компьютеров выступает в роли повторителя, усиливая сигнал, что позволяет строить сети большой протяженности;
- высокая устойчивость к перегрузкам.

Недостатки:

- сигнал должен пройти последовательно (и только в одном направлении) через все компьютеры, каждый из которых проверяет, не ему ли адресована информация, поэтому время передачи может быть достаточно большим;
- подключение к сети нового компьютера часто требует ее остановки, что нарушает работу всех других компьютеров;
- выход из строя хотя бы одного из компьютеров может нарушить работу всей сети;
- обрыв или короткое замыкание в любом из кабелей кольца делает работу всей сети практически невозможной.

Ввиду указанных выше недостатков в чистом виде топологию типа «кольцо» не применяют. Основное применение она нашла в оптоволоконных сетях стандарта Token Ring.

Ячеистая топология сети основана на принципе полносвязной топологии, в которой удалены некоторые возможные связи между устройствами. При данном типе соединения каждый компьютер имеет множество возможных путей соединения с другими компьютерами. Такая топология характеризуется высокой отказоустойчивостью, однако имеет недостатки в виде сложной настройки сети и при использовании кабельных линий связи требует большого расхода кабеля.

В настоящее время развитие получили беспроводные сети передачи данных, которые не требуют больших затрат на оборудование и прокладку кабельных линий связи. Подобные сети обладают рядом достоинств, таких как самовосстановление и самоадаптация. При подключении к такой сети каждое устройство автоматически получает информацию обо всех других устройствах в сети. Каждое такое устройство помимо передачи и приема информации способно производить маршрутизацию данных в сети, основываясь на ранее полученных сведениях о структуре сети.

Крупные сети передачи информации с произвольными связями между оборудованием строятся в преобладающем большинстве по *смешанной топологии*. В таких сетях можно выделить отдельные части сети (подсети), которые имеют типовую топологию в виде общей шины, звезды, кольца и др.

Далее рассмотрим основные типы линий связи и сетевого оборудования, используемых при построении компьютерных сетей передачи данных.

В качестве среды передачи в системах передачи данных используют проводные линии связи, оптические каналы и радиоканалы передачи данных.

Наиболее часто в компьютерных сетях применяются проводные линии связи в виде кабельных соединений, выступающие в качестве среды передачи электрических или оптических сигналов между компьютерами и другими сетевыми устройствами. В настоящее время используются следующие типы кабелей:

- коаксиальные кабели;

- «витая пара» проводников;
- волоконно-оптические, или же оптоволоконные, кабели.

Коаксиальный кабель исторически был первым типом кабеля, нашедшим применение в компьютерных сетях передачи данных. Кабель данного типа состоит из центрального проводника, покрытого пластиковым изолирующим материалом, который, в свою очередь, окружен экранирующей оплеткой. Этот внешний проводник обеспечивает заземление и защиту центрального проводника от внешних электромагнитных воздействий. Сети на основе коаксиального кабеля обеспечивают передачу со скоростью до 10 Мбит/с. Максимальная длина сегмента сети лежит в пределах от 185 до 500 м в зависимости от типа используемого кабеля. Примерами сетей, построенных на основе соединений при помощи коаксиального кабеля служат сети Ethernet стандартов 10Base-5 и 10Base-2.

Кабель типа «витая пара» (Twisted pair) является одним из наиболее распространенных типов кабеля в настоящее время. Он состоит из нескольких пар проводов, покрытых пластиковой оболочкой. Провода, составляющие каждую пару, закручены вокруг друг друга, что обеспечивает защиту от взаимных наводок. Кабели данного типа делятся на два класса – «экранированная витая пара» («Shielded twisted pair») и «неэкранированная витая пара» («Unshielded twisted pair»). Отличие этих классов состоит в том, что экранированная витая пара является более защищенной от внешнего электромагнитного воздействия благодаря наличию дополнительного экрана из медной сетки и/или алюминиевой фольги, окружающего провода кабеля. Сети на основе «витой пары» в зависимости от категории кабеля обеспечивают передачу со скоростью от 10 Мбит/с до 1 Гбит/с при длине сегмента сети не более 100 м (до 100 Мбит/с) или 30 м (1 Гбит/с).

Кабели в виде «витой пары» проводников, применяемые для подключения компьютеров к концентраторам и коммутаторам, обжимаются с двух сторон одинаково, при этом получается прямой кабель. Для непосредственного соединения сетевых адаптеров компьютеров либо для связи между концентраторами и коммутаторами используется перекрестный кабель (кросс-кабель). При использовании современного сетевого оборудования при всех соединениях можно применять прямой кабель, перестройка контактов происходит на программном уровне без вмешательства пользователя.

Оптоволоконные кабельные линии связи в настоящее время находят широкое применение в качестве среды передачи данных в сетях различных уровней, начиная от магистральных и заканчивая локальными сетями передачи данных. Волоконно-оптические линии связи получили широкое применение за счет неоспоримых преимуществ перед кабельными проводными линиями и беспроводными сетями передачи данных. В первую очередь это связано с высокой скоростью передачи информации (более 10 Гбит/с на расстоянии 1 км); малым затуханием сигнала в оптоволоконном кабеле; высокой помехоустойчивостью и защищенностью от несанкционированного доступа; небольшими массогабаритными показателями, а также относительно большими сегментами сети.

Оптоволоконный кабель состоит из центрального стеклянного или пластикового проводника, окруженного слоем стеклянного или пластикового покрытия

и внешней защитной оболочкой. Передача данных осуществляется с помощью лазерного или светодиодного передатчика, посылающего однонаправленные световые импульсы через центральный проводник. Сигнал на другом конце принимается фотодиодным приемником, осуществляющим преобразование световых импульсов в электрические сигналы, которые могут обрабатываться компьютером.

К недостаткам использования оптоволоконных линий можно отнести относительно высокую стоимость кабеля, сложность заделки соединительных разъемов и необходимость применения оптоэлектронных и электрооптических преобразователей.

Помимо проводных и оптических линий связи в сетях передачи данных широкое распространение получили беспроводные технологии передачи данных, базирующиеся на использовании радиоволн. Отличительной особенностью сетей с беспроводной передачей данных является обеспечение мобильности подключаемых к сети устройств.

Примерами сетей передачи данных, построенных с использованием стандартов беспроводной передачи данных, могут служить технологии организации беспроводных сетей, такие как Bluetooth, Zigbee, Wi-Fi, WiMAX, GPRS, UMTS, LTE и др.

Помимо этого, любая сеть передачи данных, кроме физической среды передачи информации, содержит различного рода оборудование, позволяющее передавать и принимать данные между различными узлами сети согласно существующим стандартам, правилам, соглашениям. Тип оборудования, его технические характеристики, количество оборудования зависит от разных факторов, таких как:

- топология сети;
- тип среды передачи данных;
- тип используемого сетевого стандарта;
- количество узлов в сети;
- потребности пользователей;
- уровень безопасности работы с данными.

К оборудованию, входящему в состав сетей передачи данных и обеспечивающему их нормальное функционирование, относят:

– сетевые адаптеры (проводные и беспроводные) – ключевые элементы сети передачи данных, обеспечивающие доступ компьютерам и другим сетевым устройствам к используемой среде передачи данных;

– повторители (или репитеры) – устройства, обеспечивающие усиление (повторение) сигнала с целью увеличения размера сегмента сети;

– многопортовые репитеры (или концентраторы) – устройства, предназначенные для объединения пользователей в сеть с типовой топологией «пассивная звезда»;

– мосты и коммутаторы (многопортовые мосты) – сетевые устройства, предназначенные для объединения нескольких сегментов сети и позволяющие осуществлять фильтрацию передаваемых данных;

– маршрутизаторы – сетевые устройства, предназначенные для объединения сегментов сети и обеспечивающие фильтрацию и перенаправление трафика на основе сетевых адресов;

– точки доступа и ретрансляторы – устройства активного типа, необходимые для объединения компьютеров в беспроводную сеть и усиления сигнала;

– медиаконвертеры – устройства, предназначенные для преобразования используемой среды передачи данных, например «витая пара» – оптоволокно.

1.2. Работа с программным пакетом Cisco Packet Tracer

Задачей данной лабораторной работы является ознакомление с основными типами сетевых топологий, видами сетевого оборудования, а также приобретение навыков проектирования простейших сетей в среде Cisco Packet Tracer.

Программный пакет Cisco Packet Tracer представляет собой симулятор сети передачи данных, выпускаемой компанией Cisco Systems. Он позволяет моделировать работу сети передачи данных, настраивать маршрутизаторы, коммутаторы, межсетевые экраны и другое оборудование, выпускаемое компанией.

Основное окно программы представлено на рис. 1.6.

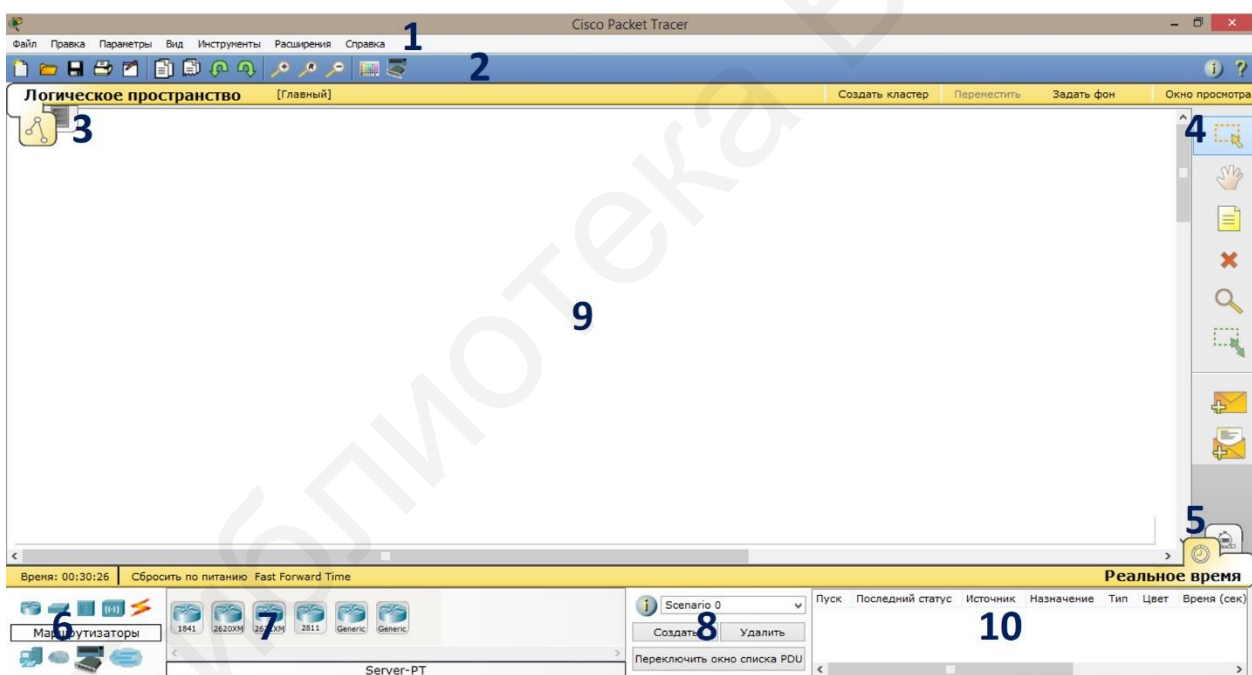


Рис. 1.6. Внешний вид основного окна Cisco Packet Tracer

Основными элементами пользовательского интерфейса являются:

1. Главное меню программы.
2. Панель инструментов.
3. Переключатель «логическая/физическая организация».
4. Дополнительная панель инструментов.
5. Переключатель «реальное время/режим симуляции».
6. Панель с группами устройств и линий связи.

7. Устройства.

8. Панель создания пользовательских сценариев.

9. Рабочее пространство.

Главное меню программы, как и любой другой программы, содержит базовые операции работы с файлами и средой: открытие, закрытие, сохранение файлов; стандартные операции: копировать, вырезать, вставить; настройки интерфейса программы; справку об основных возможностях программы.

Панель инструментов и дополнительная панель инструментов содержат набор инструментов для создания и редактирования схем, масштабирования объектов, формирования произвольных пакетов.

Переключатель «логическая/физическая организация» позволяет выполнить переход от физической топологии сети к логической и наоборот.

Переключатель «реальное время/режим симуляции» позволяет посмотреть работу сети передачи данных в режиме реального времени или же перейти в режим симуляции.

В панели с группами сетевых устройств и линий связи находятся модели различных сетевых устройств: коммутаторы, концентраторы, маршрутизаторы, различные варианты линий связи между устройствами сети и другое оборудование.

Панель «устройства» отражает набор выбранных текущих сетевых устройств.

В панели создания пользовательских сценариев можно производить наблюдение за передачей пакетов при визуальном моделировании работы сети.

В поле рабочего пространства создается требуемая сеть передачи данных из оборудования, представленного в панелях с группами сетевых устройств. Схема сети формируется по принципу перетаскивания из указанных панелей требуемых устройств и создания между ними необходимых связей.

Для создания сети необходимо выбрать тип оборудования в окне 6, далее выбрать оборудование в окне 7 и перетащить его в рабочее пространство.

Соединение оборудования согласно карте сети производится следующим образом. Для этого в окне 6 следует выбрать «Соединения», после чего в окне 7 выбрать необходимый тип соединения. Щелчок левой кнопкой мыши на устройстве вызовет выпадающее меню, где необходимо выбрать порт для подключения кабеля (рис. 1.7).

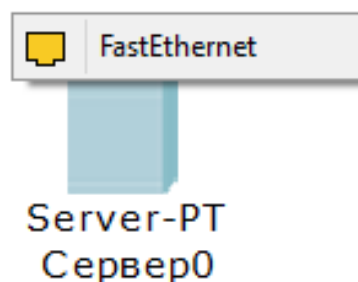


Рис. 1.7. Выбор порта для подключения кабеля

При правильном выборе линии связи между соединяемыми устройствами сети у начала и конца линии связи должен загореться зеленый индикатор, как показано на рис. 1.8.

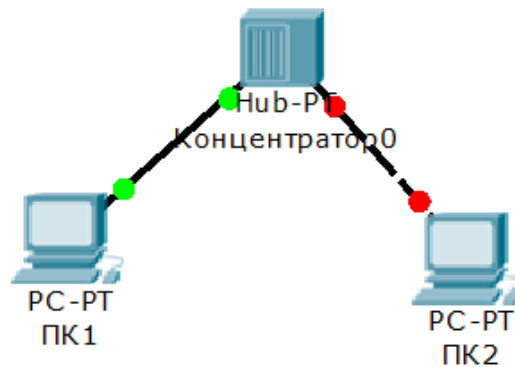


Рис. 1.8. ПК1 подключен к концентратору правильно (индикатор зеленый), ПК2 подключен неправильно (индикатор красный)

При подключении устройств к коммутаторам и маршрутизаторам индикатор на коммутаторе (маршрутизаторе) какое-то время будет иметь оранжевый цвет, что свидетельствует о выполнении инициализации соответствующего порта. Когда индикатор станет зеленым – оборудование готово к использованию (рис. 1.9).

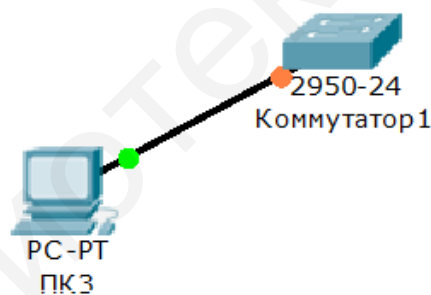


Рис. 1.9. Оранжевый цвет индикатора указывает на проведение процедуры инициализации порта

Дальнейшая настройка компьютерной сети передачи данных заключается в настройке параметров оборудования, расположенного на схеме. Для настройки оборудования следует выполнить двойной щелчок левой кнопкой мыши на интересующем устройстве на рабочем поле. После этого появится окно с настройками оборудования (рис. 1.10).

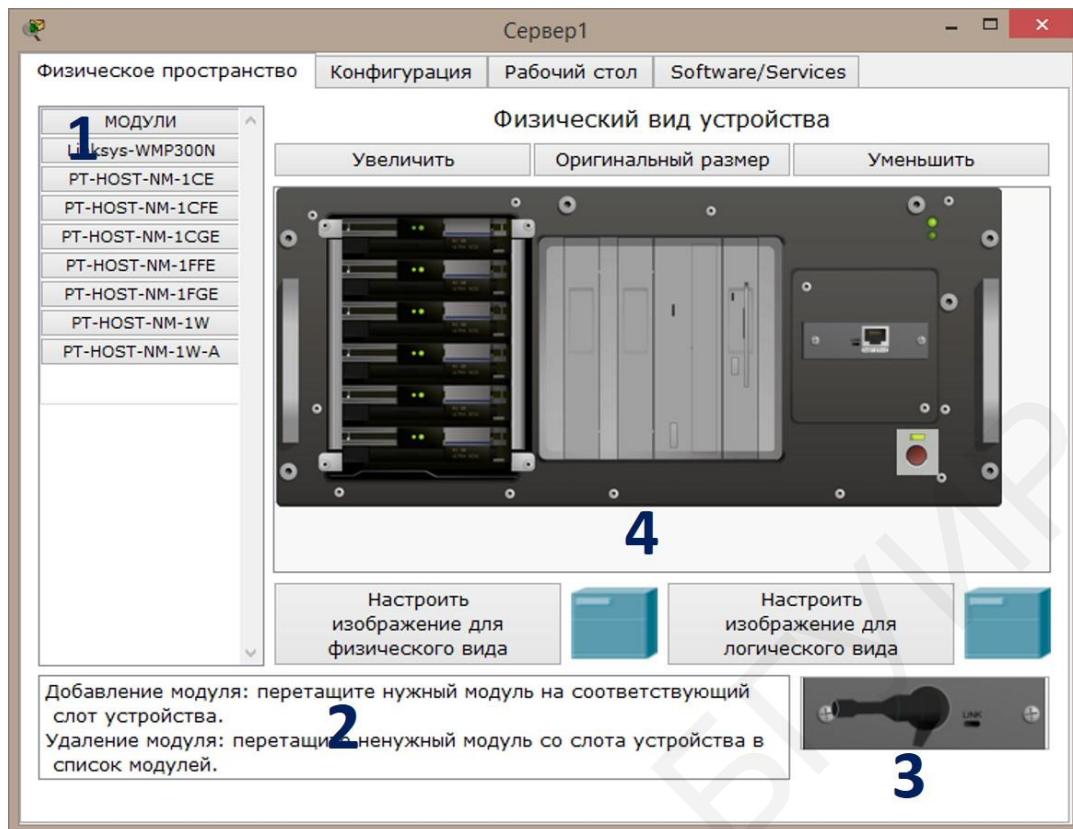


Рис. 1.10. Меню настроек физического пространства устройства «Сервер1»

Вкладка «Физическое пространство» позволяет выбрать необходимые модули для создания сети и подключить их к устройству. В окне 1 представлены возможные для подключения к выбранному устройству модули. В окне 2 приводится описание модуля, в окне 3 – внешний вид выбранного модуля, в окне 4 – внешний вид устройства.

Подключение и отключение модулей производится путем их перетаскивания из окна 1 в свободные слоты на устройстве в окне 4. Если подключаемый модуль не поддерживает технологию Plug-and-Play, то перед его установкой необходимо отключить питание.

Вкладка «Конфигурация» (рис. 1.11) позволяет выполнить настройку устройства. В окне 1 нужно выбрать необходимую область настройки, после чего в окне 2 выполнить настройки оборудования.

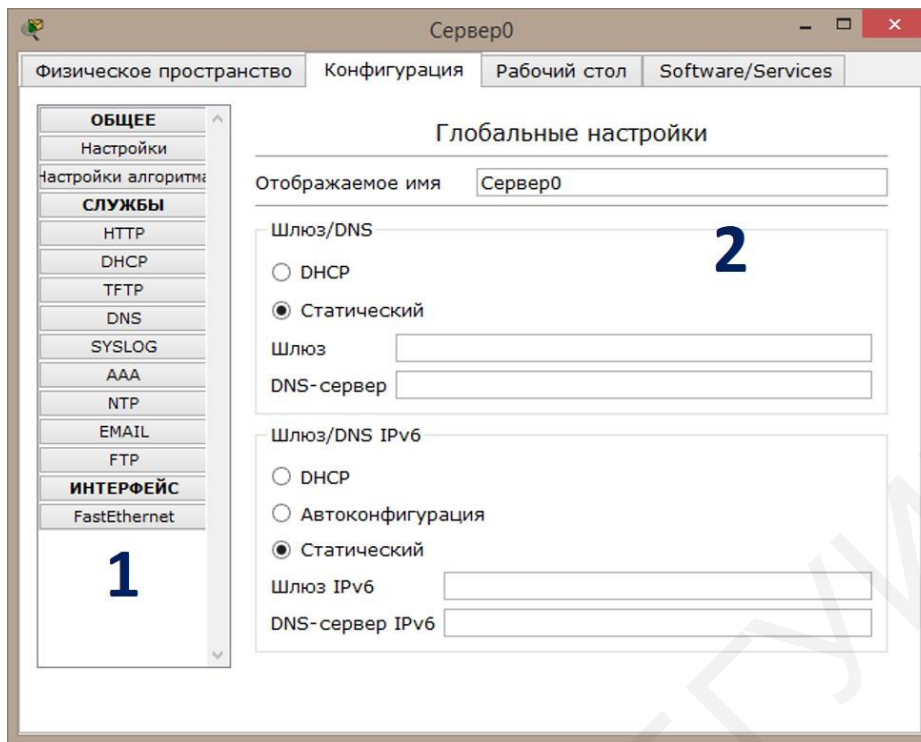


Рис. 1.11. Вкладка «Конфигурация» окна настроек оборудования

На вкладке «Рабочий стол» (не для всех устройств) (рис. 1.12) можно запускать различные приложения, необходимые для проверки работы сети, а также выполнения некоторых настроек.



Рис. 1.12. Вкладка «Рабочий стол»

Для некоторых устройств можно выполнять конфигурирование оборудования путем прописывания команд на вкладке «CLI» (рис. 1.13).

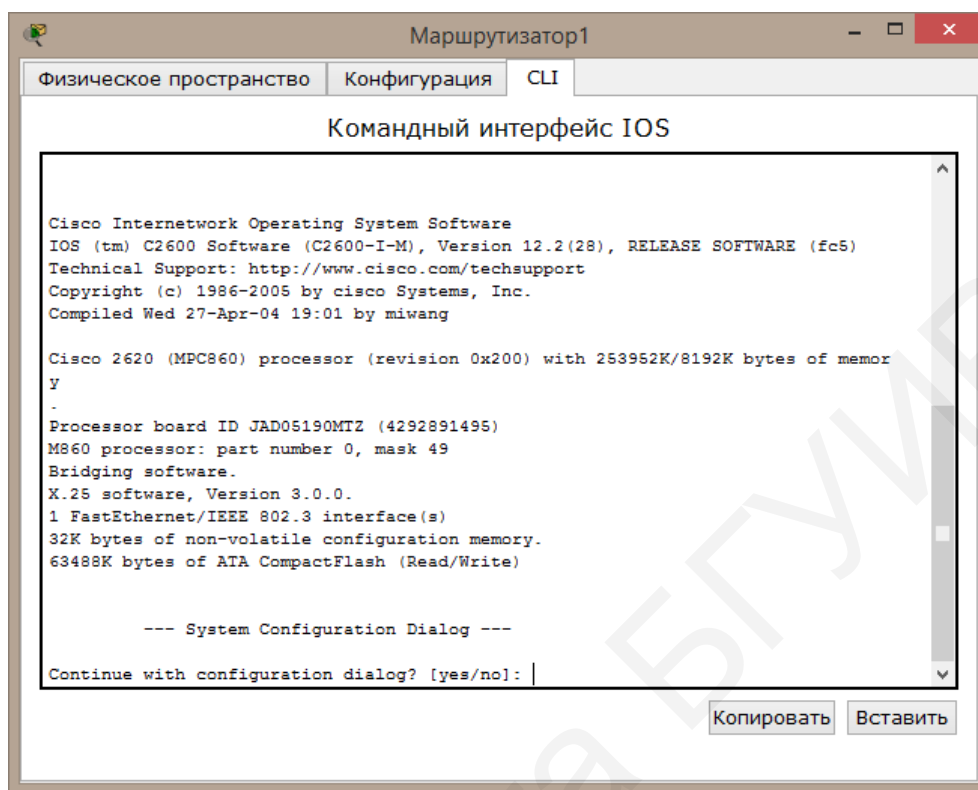


Рис. 1.13. Содержимое вкладки «CLI»

1.3. Задание для выполнения лабораторной работы

В ходе выполнения лабораторной работы требуется создать локальную компьютерную сеть со статической IP-адресацией. Компьютерная сеть должна иметь 1 сервер, от 6 до 12 персональных компьютеров (задается преподавателем). Соединение устройств в сеть должно производиться при помощи концентратора с использованием технологии FastEthernet в программном пакете Cisco Packet Tracer. В ходе работы необходимо настроить оборудование и проверить правильность работы полученной сети путем передачи пакетов данных между различными устройствами.

1. Используя панели с группами устройств и перечнем устройств, расположить на рабочем поле необходимое оборудование и выполнить его соединение в сеть при помощи линий связи.

2. Произвести настройку сервера. Для этого в меню настройки устройства установить следующие параметры:

- IP-адрес сервера 192.168.1.1;
- маска сети 255.255.255.0;
- отключить DHCP;

– настроить сервер в качестве HTTP-сервера.

3. Назначить IP-адреса всем оставшимся персональным компьютерам из диапазона адресов 192.168.1.2...192.168.1.254 и маску сети 255.255.255.0.

4. Проверить командой *ping* связь между любыми компьютерами и между компьютером и сервером.

Узнать адрес устройства, связь с которым необходимо проверить, можно путем наведения указателя мыши на устройство. В появившемся окне будет отображен IP-адрес данного устройства.

Далее на устройстве, с которого проверяется связь, следует открыть окно настроек, перейти во вкладку «Рабочий стол», открыть командную строку. В командной строке прописать команду, имеющую следующий синтаксис:

ping <IP-адрес устройства, связь с которым проверяется>

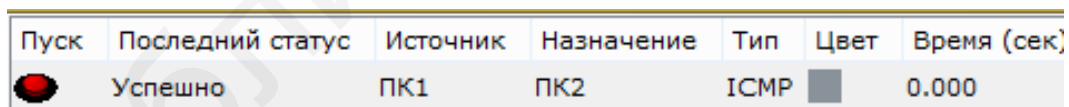
Пояснить выводимые в консоль командной строки параметры ответа опрашиваемого устройства.

5. Проверить работу HTTP-сервера. Для этого в панели настроек открыть вкладку «Рабочий стол» и запустить «Веб-браузер».

В адресной строке браузера ввести IP-адрес HTTP-сервера. При правильной настройке откроется веб-страница.

6. Проследить процесс передачи простого PDU (P) пакета между сервером и одним из компьютеров, между двумя различными компьютерами. Пояснить путь прохождения пакета.

Для выполнения этого пункта лабораторного задания следует в дополнительной панели инструментов выбрать «Добавить простой PDU». Щелчком левой кнопки мыши выбрать вначале отправителя, а затем – получателя. При правильной настройке в окне «Переданные пакеты» появится новая запись со статусом «успешно», как показано на рис. 1.14.





Пуск	Последний статус	Источник	Назначение	Тип	Цвет	Время (сек)
	Успешно	ПК1	ПК2	ICMP		0.000

Рис. 1.14. Успешная пересылка пакета

Далее следует перейти в режим симуляции. В окне 1 (рис. 1.15) можно проследить пути прохождения пакета. Ползунком 2 можно регулировать скорость анимации прохождения пакета. Нажатием кнопки «Захват/вперед» проследить путь прохождения пакета от отправителя к получателю и обратно.

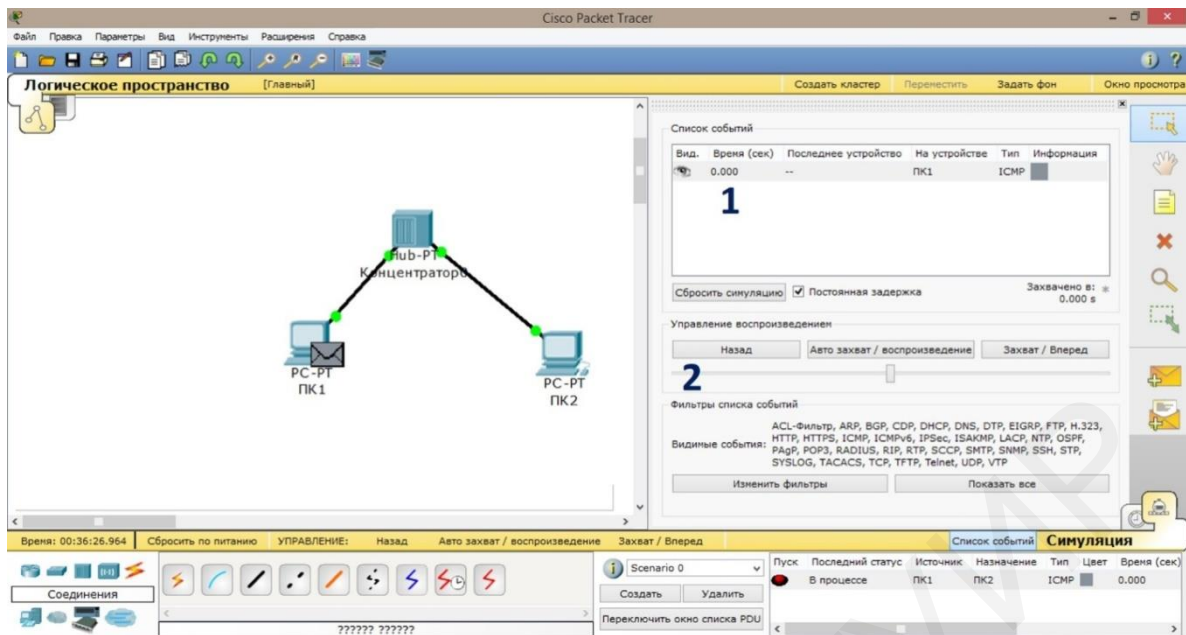


Рис. 1.15. Режим «Симуляция»

7. На схеме сети заменить концентратор коммутатором и повторить п. 4.

1.4. Содержание отчета

1. Титульный лист.
2. Цель работы.
3. Схема топологии сети.
4. Примеры конфигурации используемого оборудования.
5. Результаты проверки связи между устройствами сети.
6. Результаты проверки работоспособности HTTP-сервера.
7. Результаты прохождения пакета данных между устройствами сети.
8. Выводы.

1.5. Контрольные вопросы

1. Приведите классификацию компьютерных сетей.
2. В чем заключается разница между одноранговыми и иерархическими сетями?
3. Что такое модель взаимодействия открытых систем?
4. Эталонная модель взаимодействия открытых систем OSI.
5. Перечислите основные топологии компьютерных сетей и дайте их краткую характеристику.
6. Назовите основные виды линий связи между устройствами в системах передачи данных.
7. Назовите основные виды и назначение сетевого оборудования.

Лабораторная работа №2 IP-АДРЕСАЦИЯ

Цель работы:

- изучить основы IP-адресации в компьютерных сетях;
- изучить классы и типы IP-адресов, принципы разбиения сетей на подсети;
- ознакомиться с символьным представлением имени компьютера в сети.

2.1. Краткие теоретические сведения

2.1.1. Методы адресации устройств в сети

Для того чтобы устройства в сети могли взаимодействовать друг с другом, они должны иметь возможность однозначно себя идентифицировать. Таким образом, каждое устройство в сети передачи данных должно иметь адрес, по которому к нему могут обратиться другие устройства.

К адресу устройства можно предъявить ряд требований:

- адрес должен быть уникальным для идентификации в сети любого масштаба;
- схема назначения адресов должна сводить к минимуму ручной труд при его назначении и вероятность дублирования адресов;
- адрес должен иметь иерархическую структуру, удобную при построении больших сетей;
- он должен быть удобен пользователям сети, т. е. должен иметь символьное представление;
- адрес должен иметь компактное представление для уменьшения загрузки памяти сетевых устройств.

Приведенные требования невозможно совместить в рамках одной схемы адресации, поэтому на практике используется сразу несколько схем адресации. В этом случае компьютер или иное устройство имеет сразу несколько сетевых адресов, и каждый из этих адресов используется в той ситуации, когда он наиболее удобен.

Множество всех адресов, допустимых в рамках определенной адресации, называется *адресным пространством*. Оно может иметь линейную или иерархическую организацию.

В настоящее время наибольшее распространение получили следующие типы адресации устройств в системах передачи данных:

- локальные адреса;
- числовые составные адреса;
- символьные адреса (имена).

Для преобразования типов адресов используют специальные протоколы, которые называются протоколами разрешения адресов.

По количеству адресуемых сетевых интерфейсов различают:

– одноадресный, или уникальный, – используется для идентификации отдельных интерфейсов (например, физического интерфейса между компьютером и сетью) устройства и позволяет пересылать сообщения в одну точку сети;

– групповой – идентифицирует сразу несколько интерфейсов, в этом случае данные доставляются сразу нескольким устройствам, входящим в группу произвольно расположенных узлов сети;

– широковещательный – используется для доставки сообщений всем узлам сети;

– адрес произвольной рассылки – задает группу интерфейсов, но данные доставляются только одному члену группы (стандарт IPv6); этот адрес назначается только интерфейсам маршрутизатора.

Локальный адрес (также называют физический или аппаратный) – такой тип адреса, который используется средствами базовой технологии для доставки данных в пределах подсети, являющейся элементом составной сети. В разных подсетях допустимы различные сетевые технологии, следовательно, различные протоколы. Поэтому существуют разные типы локальных адресов. Для ЛВС локальный адрес – это MAC-адрес сетевого адаптера. Физические адреса не имеют иерархической структуры. Компьютер может иметь несколько сетевых интерфейсов и соответственно несколько физических адресов.

Во многих случаях для работы в больших сетях в качестве адресов узлов используют числовые составные адреса. Представителями адресов этого типа являются адреса стандартов IPv4 и IPv6.

2.1.2. Адреса IPv4

IPv4-адрес – представляет собой 32-битный адрес (4 байта). Он может быть представлен в двоичном или шестнадцатеричном формате.

Для удобства чтения в технической литературе и прикладных программах IPv4-адреса представляются в виде четырех десятичных чисел, разделенных точками. Каждое из чисел соответствует одному октету (8 битам) и может иметь значения от 0 до 255. Этот формат называется точечно-десятичным. Например:

10000000.00001010.00000010.00011110
128.10.2.30

В IPv4-адресах используется двухуровневая иерархия. Адрес делится на две логические части. Старшую часть – номер сети (ID сети) и младшую – номер узла (ID узла). Такое деление позволяет передавать сообщения между сетями только на основании номера сети, а номер узла используется после доставки сообщения в нужную сеть.

Устройствами, которым назначаются IP-адреса, могут быть сетевые интерфейсы конечных узлов, коммуникационные серверы, порты маршрутизаторов. Конечный узел может входить в несколько IP-сетей, следовательно, может иметь

несколько IP-адресов. Таким образом, IP-адрес характеризует не отдельный узел, а одно сетевое соединение данного узла.

Чтобы понять, какие из 32 битов используются для ID сети и для ID узла, требуется дополнительная информация. Она представлена маской подсети.

Маска – это 32-разрядное двоичное число, которое используется в паре с IP-адресом и содержит единицы в тех разрядах, которые должны в IP-адресе интерпретироваться как номер сети. Таким образом, маска используется для определения части IPv4-адреса, которая представляет ID сети.

Маска может быть указана в точечно-десятичной нотации либо как десятичное число после косой черты вслед за IP-адресом.

Например, маска подсети /16 в представлении с разделительными точками выглядит как 255.255.0.0, а маска подсети /24 – как 255.255.255.0.

Представление с косой чертой называется представлением бесклассовой междоменной маршрутизации CIDR (Classless Inter Domain Routing).

Диапазоны IPv4-адресов. Индивидуальные IPv4-адреса можно разбить на *публичный диапазон, частный диапазон, групповые адреса и APIPA*. Адреса APIPA используются лишь в качестве временных адресов или для изолированных компьютеров, а публичные и частные диапазоны делятся на блоки, которые можно назначать целым сетям.

Каждый IPv4-адрес в сети Интернет должен быть уникален. Распределение адресного пространства курирует Администрация адресного пространства Интернет (Internet Assigned Numbers Authority, IANA).

IANA делегирует ответственность за распределение адресов региональным регистраторам: Asia-Pacific Network Information Center (APNIC), American Registry for Internet Numbers (ARIN) и Reseaux IP Europeans Network Coordination Centre (RIPE NCC).

Затем региональные регистраторы выделяют блоки адресов крупным поставщикам услуг сети Интернет (Internet Service Provider, ISP), которые предоставляют блоки своего адресного пространства потребителям и небольшим интернет-провайдерам.

Администрация IANA зарезервировала также определенные диапазоны IPv4-адресов в качестве *частных адресов*. Они никогда не используются в глобальной сети Интернет. Эти частные IPv4-адреса применяются для узлов сетей, которым нужны коммуникации IPv4 без отображения в сеть Интернет.

Диапазоны частных адресов:

Начальный адрес	Конечный адрес
10.0.0.0	10.255.255.254
172.16.0.0	172.31.255.254
192.168.0.0	192.168.255.254

Узлы с частными адресами могут подключаться к сети Интернет через сервер или маршрутизатор, выполняющий преобразование сетевых адресов NAT.

NAT (Network Address Translation) – технология трансляции адресов – преобразование адресов с помощью специальных таблиц соответствия.

Часто публичные адреса назначаются общедоступным серверам, а частные – клиентским компьютерам. Каждая организация, которой требуются коммуникации в сети Интернет, должна располагать хотя бы одним публичным адресом. Этот адрес может использоваться множеством клиентов посредством NAT и диапазонов частных адресов.

Автоматические частные IP-адреса (Automatic Private IP Addressing, APIPA). Если компьютеру автоматически назначается IP-адрес, то по умолчанию в случае недоступности DHCP-сервера всем сетевым подключениям назначаются адреса APIPA. Частные адреса APIPA расположены в диапазоне от 169.254.0.1 до 169.254.255.254. Маска подсети 255.255.0.0.

Групповые IPv4-адреса начинаются с префикса 1110. Значение первого октета у них от 224 и выше. Они не делятся на номер сети и номер узла – это особый групповой адрес. Пакет с групповым адресом должны получить все узлы, которым присвоен данный адрес. Групповая адресация широко используется в сети Интернет.

Специальные IP-адреса. Некоторые IP-адреса зарезервированы для специальных целей. Такие адреса не назначаются конечным узлам и не передаются маршрутизаторами. Специальными IP-адресами являются:

1. Адрес 0.0.0.0 обозначает все сетевые интерфейсы данного узла либо шлюз по умолчанию (default gateway). В IPv6 это адрес « :: ».

2. Если номер сети равен нулю, то по умолчанию считается, что узел назначения принадлежит той же самой сети, что и узел-отправитель.

3. Если все двоичные разряды IP-адреса равны единице (255.255.255.255), то пакет с таким адресом назначения должен рассылаться всем узлам, находящимся в той же сети, что и источник пакета. Такая рассылка называется ограниченным широковещательным сообщением (limited broadcast), т. е. только в пределах той сети, где находится отправитель. Пакет с ограниченным широковещательным адресом никогда не будет пропущен через маршрутизатор. В IPv6 он имеет префикс 1111 1111 и относится к групповым адресам.

4. Если в поле номер узла назначения – все двоичные единицы, то пакет рассылается всем узлам сети с заданным номером сети. Такая рассылка называется направленным широковещательным сообщением (broadcast или multicasting). Например, 192.190.21.255/24.

5. IP-адрес, первый байт которого равен 127, используется для тестирования программ и взаимодействия процессов в пределах локальной машины. Данные при этом не передаются по сети, а опускаются до физического уровня и сразу возвращаются модулям верхнего уровня. Образуется «петля». Поэтому адрес 127.0.0.1 называется loopback или «петля обратной связи». Достижим только с локальной машины, поэтому называется еще localhost. В IPv6 это адрес ::1.

6. Адрес, у которого в поле номер узла – все нули, обозначает пул адресов. Например, 129.35.0.0/16.

7. Адреса, значение первого байта которых превышает 223, не могут использоваться в качестве номера узла, так как они используются для групповой адресации.

2.1.3. Адреса IPv6

Версия IPv4 обеспечивает 4,3 млрд возможных уникальных адресов. Для решения проблемы исчерпания адресного пространства IPv4 была разработана версия IPv6. Вместо 32-битовых адресов версии IPv4 в версии IPv6 используются 128-битовые. Адресное пространство IPv6 обеспечивает 2^{128} , или 340 282 366 920 938 463 463 374 607 431 768 211 456 ($3,4 \cdot 10^{38}$), уникальных адресов.

IPv6-адреса состоят из восьми блоков по четыре шестнадцатеричных цифры в каждом. Каждый блок отделяется двоеточиями. Пример полного IPv6-адреса:

2001:00B8:3FA9:0000:0000:0000:0003:9C5A

IPv6-адрес можно сократить, исключив все незначащие нули в блоках. Таким образом, предыдущий адрес можно сократить до такого:

2001:DB8:3FA9:0:0:0:D3:9C5A

Затем этот адрес можно еще более сократить, заменив все смежные нулевые блоки двойным двоеточием (::). В отдельном IPv6-адресе это можно сделать только один раз:

2001:DB8:3FA9::D3:9C5A

Поскольку IPv6-адреса состоят из восьми блоков, всегда можно определить, сколько блоков нулей представлены двойными двоеточиями. Например, в предыдущем IPv6-адресе двойные двоеточия представляют три нулевых блока, поскольку в адресе присутствует пять блоков.

IPv6-адреса разделены на две части: 64-битовый компонент сети и 64-битовый компонент узла.

Компонент сети идентифицирует уникальную подсеть, и администрация IANA выделяет эти числа поставщикам ISP или крупным компаниям.

Компонент узла, как правило, основан на уникальном 48-битовом MAC-адресе сетевого адаптера или генерируется случайным образом.

Для одноадресных типов IPv6 не поддерживает идентификаторы подсетей переменной длины, а число битов, используемых для идентификации сети одноадресного типа IPv6-адреса, всегда равно 64 (первая половина адреса). Поэтому для представления одноадресных типов IPv6 нет необходимости указывать маску подсети, поскольку компьютеры распознают идентификатор /64.

IPv6-адреса используют сетевые префиксы, выражаемые в представлении с косой чертой, однако лишь для описания маршрутов и диапазонов адресов, а не для указания ID сети. Например, в таблице маршрутизации IPv6 можно встретить такую запись: 2001:DB8:3FA9::/48.

В отличие от IPv4 версия IPv6 не использует широковещание в сети. Вместо широковещания в IPv6 применяется многоадресная или групповая передача.

Версия IPv6 изначально проектировалась для обеспечения более простого конфигурирования узлов, чем IPv4. Хотя IPv6 можно конфигурировать и вручную (обычно это требуется для маршрутизаторов), конфигурирование IPv6 на компьютерах практически всегда выполняется автоматически. Компьютеры могут получать IPv6-адреса от соседних маршрутизаторов или DHCPv6-серверов. Кроме того, компьютеры всегда сами назначают себе адрес для использования исключительно в локальной подсети.

Версия IPv6 описывает три типа адресов: *глобальные адреса, канальные и уникальные локальные адреса*.

Глобальные IPv6-адреса (GA) аналогичны публичным адресам в сетях IPv4 и используются для области IPv6 сети Интернет. Для глобальных адресов в настоящее время применяется префикс 2000::/3, который преобразуется в стандартное шестнадцатеричное значение первого блока между 2000 и 3FFF. Например:

2001:db8:21da:7:713e:a426:d167:37ab.

Канальные адреса (Link-Local Address, LLA) аналогичны автоматически назначаемым частным адресам APIPA (Automatic Private IP Addressing) в IPv4 (например, 169.254.0.0/16). Они конфигурируются самостоятельно и могут использоваться лишь для коммуникаций в локальной подсети. Но в отличие от адреса APIPA канальный адрес LLA назначается интерфейсу как вспомогательный даже после получения маршрутизируемого адреса для этого интерфейса. Канальный адрес LLA всегда начинается с fe80. Пример канального адреса – fe80::154d:3cd7:b33b:1bc1%13.

Уникальные локальные адреса (Unique Local Address, ULA) в IPv6 аналогичны частным адресам в IPv4 (10.0.0.0/8, 172.16.0.0/12 и 192.168.0.0/16). Эти адреса маршрутизируются между подсетями в частной сети и не маршрутизируются в сети Интернет. Они позволяют создавать комплексные внутренние сети. Такие адреса начинаются с fd, как, например, локальный уникальный адрес fd65:9abf:efb0:0001::0002.

Узлы IPv6, как правило, автоматически конфигурируют IPv6-адреса, взаимодействуя с IPv6-маршрутизатором. В течение короткого промежутка времени между первым назначением адреса и проверкой его уникальности адрес называется пробным. Компьютеры используют обнаружение дубликатов адресов, чтобы идентифицировать другие компьютеры с тем же IPv6-адресом, отправляя запрос обнаружения соседей (Neighbor Solicitation) с предварительным адресом. Если какой-либо компьютер ответил на запрос, адрес считается недействительным. Если на запрос не ответил ни один компьютер, адрес считается уникальным.

и действительным. Действительный адрес называется основным в течение срока действия, назначенного маршрутизатором или в автоматической конфигурации. По истечении этого жизненного цикла действительный адрес считается устаревшим. В существующих сеансах коммуникаций может использоваться устаревший адрес.

2.1.4. Символьные адреса

Каждый компьютер в сети имеет уникальный адрес. При использовании IP-адресации – это IP-адрес. Однако пользователю достаточно трудно оперировать длинными наборами цифр, не несущих смысловой нагрузки, поэтому применяются системы преобразования имен, ставящие в соответствие цифровому адресу компьютера его символьное имя. В глобальных сетях и сети Интернет это служба DNS (Domain Name System) – распределенная база данных, поддерживающая иерархическую систему имен для идентификации узлов в сети Интернет. Определенные части базы данных доменных имен хранятся на специальных серверах – DNS-серверах. Они обрабатывают запросы компьютера и определяют имя, соответствующее IP-адресу, и наоборот. В каждой локальной сети, подключенной к сети Интернет, работает по крайней мере один DNS-сервер. База данных DNS имеет структуру дерева, называемого доменным пространством имен, в котором каждый домен (узел дерева) имеет имя и может содержать поддомены. Имя домена идентифицирует его положение в этой базе данных по отношению к родительскому домену, а точки в имени отделяют части, соответствующие узлам домена.

Для именованя компьютеров в локальных сетях используются линейные (не имеющие иерархии) символьные имена, так называемые NetBIOS-имена. Протокол NetBIOS (Network Basic Input/Output System) как расширение стандартных функций базовой системы ввода-вывода был разработан в 1984 г. компанией IBM и широко применяется в ее продуктах, а также продуктах компании Microsoft. В протоколе NetBIOS реализован механизм широковещательного разрешения имен, когда все компьютеры в локальной сети получают запрос на разрешение имени, соответствующего некоторому IP-адресу. Кроме того, компания Microsoft для своей сетевой операционной системы Windows NT разработала централизованную службу разрешения имен WINS (Windows Internet Name Service). WINS-сервер, работающий в локальной сети, централизованно обрабатывает все запросы, касающиеся разрешения имен в сетях Windows. При большом числе компьютеров в локальной сети WINS-сервер необходим. Однако в малых сетях, содержащих менее 10 компьютеров, часто используется широковещательный механизм разрешения имен протокола NetBIOS, упрощающий административное обслуживание таких сетей. В сетях без поддержки NetBEUI/NetBIOSover TCP/IP для разрешения имен используют DNS-серверы.

2.1.5. Разбиение IPv4 на подсети

Ранее при развертывании сети организации все компьютеры и другие сетевые устройства часто подключали к одной IP-сети. Всем устройствам в организации назначались IP-адреса с одинаковой сетевой частью. Конфигурация такого типа называется линейной архитектурой сети. В небольшой сети с небольшим количеством устройств такая архитектура не представляет проблемы. Однако по мере расширения сети с такой конфигурацией могут возникнуть серьезные трудности.

В Ethernet-сети устройства выполняют поиск необходимых служб и устройств с помощью широковещательной рассылки. Широковещательное сообщение доставляется всем узлам данной сети. Протокол DHCP – пример сетевой службы, которая зависит от широковещательной рассылки. Устройства отправляют по сети широковещательные запросы, чтобы определить местонахождение DHCP-сервера. В крупной сети из-за этого может создаваться значительный трафик, который уменьшит пропускную способность сети. Кроме того, поскольку широковещательная рассылка выполняется по всем устройствам, им необходимо принять и обработать трафик, что приводит к повышению требований к обработке. Если устройство должно обработать значительный объем широковещательных рассылок, это может привести к замедлению работы устройства. По этой причине крупные сети необходимо разделить на более мелкие подсети, предназначенные для небольших групп устройств и служб.

Процесс сегментации сети путем деления ее на несколько более мелких сетей называется разбиением на подсети. Сетевые устройства и службы могут группироваться в подсети по различным признакам:

- их местоположению;
- организационному подразделению;
- типу устройств;
- другим признакам.

Разбиение на подсети позволяет снизить общую нагрузку на сеть и повысить ее производительность.

Для взаимодействия узлов из разных подсетей необходим маршрутизатор. Устройства в сети используют интерфейс маршрутизатора, подключенный к их локальной сети, в качестве шлюза по умолчанию. Трафик, отправляемый на устройство в удаленной сети, будет обработан маршрутизатором и отправлен в направлении сети назначения. Для определения, является ли трафик локальным или удаленным, маршрутизатор использует маску подсети.

В пространстве подсети этот механизм реализуется аналогичным образом. Подсети образуют несколько логических сетей из одного блока адресов или сетевого адреса. Каждая подсеть рассматривается как отдельное сетевое пространство. Устройства одной подсети должны использовать адрес, маску подсети и шлюз по умолчанию той подсети, которой они принадлежат.

Трафик не может передаваться между подсетями без использования маршрутизатора. У каждого интерфейса маршрутизатора должен быть IPv4-адрес, принадлежащий сети или подсети, к которой подключен этот интерфейс.

Принцип разделения одной сети на подсети показан на рис. 2.1.

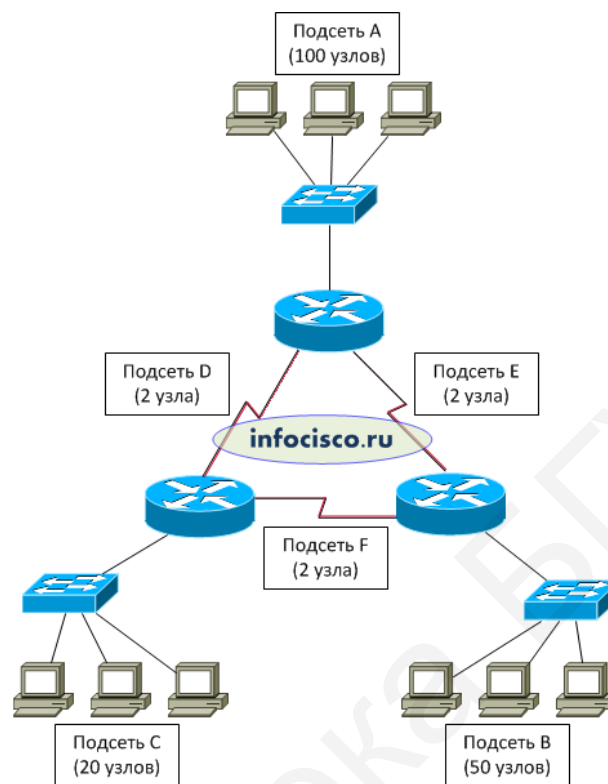


Рис. 2.1. Принцип разбиения сети на подсети

Рассмотрим пример разбиения сети на подсети на примере разбиения исходной IPv4 сети на две подсети.

Количество хостов в подсети определяется как $2^{32-N} - 2$, где N – длина маски. Чем длиннее маска, тем меньше в ней хостов.

Вычитаемая цифра 2 является двумя адресами, которые нельзя присвоить хосту, – адрес сети и адрес широковещательной рассылки.

Адрес сети – в порции хоста все нули, адрес широковещательной рассылки – в порции хоста все единицы.

Количество адресов в сети всегда четное, кроме того, оно всегда равно степени числа 2.

Исходная сеть: 192.168.150.0. Маска: 255.255.255.0.

Network	192.168.150.0	11000000.10101000.10010110.00000000
Netmask	255.255.255.0	11111111.11111111.11111111.00000000

Полужирным шрифтом выделена сетевая часть IP-адреса.

Для того чтобы получить две новые подсети, достаточно изменить маску и 24 на 25, увеличив сетевую часть IP-адреса на 1 бит. Увеличение маски на 1 бит даст

два возможных варианта подсетей – когда новый бит равен 0 и когда новый бит равен 1. Таким образом, получаем две новые подсети:

Network 1	192.168.150.0	11000000.10101000.10010110.00000000
Netmask	255.255.255.128	11111111.11111111.11111111.10000000
Network 2	192.168.150.128	11000000.10101000.10010110.10000000
Netmask	255.255.255.128	11111111.11111111.11111111.10000000

Для первой подсети новый бит равен 0 – это подсеть 192.168.150.0/25. Для второй подсети новый бит равен 1 – это подсеть 192.168.150.128/25.

Теперь рассчитаем адрес широковещательной рассылки (Broadcast). Адрес широковещательной рассылки – это такой IP-адрес, в хостовой части которого все биты равны 1.

Network 1	192.168.150.0	11000000.10101000.10010110.00000000
Netmask	255.255.255.128	11111111.11111111.11111111.10000000
Broadcast	192.168.150.127	11000000.10101000.10010110.01111111
Network 2	192.168.150.128	11000000.10101000.10010110.10000000
Netmask	255.255.255.128	11111111.11111111.11111111.10000000
Broadcast	192.168.150.255	11000000.10101000.10010110.11111111

Теперь рассмотрим пример расчета маски подсети, содержащей 10 компьютеров.

Необходимо подобрать степень числа 2, равную или больше 10 (необходимо помнить про два зарезервированных адреса).

$2^3 = 8$, $2^4 = 16$. Это четвертая степень. Следовательно, последние 4 бита маски приравниваются к 0: 11111111.11111111.11111111.11110000.

Маска подсети будет иметь вид 255.255.255.240.

Network	192.168.150.0	11000000.10101000.10010110.00000000
Netmask	255.255.255.240	11111111.11111111.11111111.11110000
Broadcast	192.168.150.15	11000000.10101000.10010110.00001111
Host _{min}	182.168.150.1	11000000.10101000.10010110.00000001
Host _{max}	192.168.150.14	11000000.10101000.10010110.00001110
Hosts	14	

2.2. Задание для выполнения лабораторной работы

1. Требуется создать сеть стандарта IPv4, имеющую адрес 192.168.1.0 и состоящую из четырех подсетей. Количество хостов в каждой подсети выбрать согласно варианту из табл. 2.1.

Таблица 2.1

Количество хостов в подсети

Подсеть	Номер варианта							
	1	2	3	4	5	6	7	8
1	8	7	6	9	20	7	5	15
2	5	9	15	13	5	16	19	18
3	12	23	19	12	14	21	11	10
4	21	19	22	22	2	4	3	7

Для каждой подсети определить маску сети, минимальный и максимальный адрес в подсети, адрес широковещательной рассылки. Результаты вычислений свести в таблицу (пример заготовки таблицы представлен табл. 2.2).

Таблица 2.2

Пример таблицы для заполнения

Параметр	Десятичная запись	Двоичная запись
Сеть (Network)		
Маска сети (Netmask)		
Адрес широковещательной рассылки (Broadcast)		
Минимальный адрес устройства в сети ($Host_{min}$)		
Максимальный адрес устройства в сети ($Host_{max}$)		
Количество устройств (хостов) в сети		

2. В программе Cisco Packet Tracer реализовать две-три подсети по указанию преподавателя. Проверить прохождение пакетов данных между устройствами внутри подсети и между подсетями. Объяснить полученные результаты.

2.3. Содержание отчета

1. Титульный лист.
2. Цель работы.
3. Результаты расчета разбиения сети на подсети.
4. Топология сети.
5. Результаты прохождения пакета данных между устройствами сети.
6. Выводы.

2.4. Контрольные вопросы

1. Что такое IP-адресация?
2. Какие стандарты IP-адресации существуют?
3. Что такое классовая и бесклассовая адресация?
4. Что такое маска сети и для чего она предназначена?
5. Что такое IP-адреса специального назначения?
6. Что такое «белые» и «серые» IP-адреса?
7. Может ли в одной сети быть два одинаковых IP-адреса?
8. Как осуществляется обмен данными между подсетями? Какое оборудование необходимо для этого?

Библиотека БГУИР

Лабораторная работа №3 ПРОТОКОЛ ПЕРЕДАЧИ ФАЙЛОВ FTP

Цель работы:

- изучить модель протокола передачи файлов FTP по сети;
- научиться принимать и загружать файлы с/на FTP-сервера;
- изучить основы работы с межсетевой операционной системой Cisco IOS в программном пакете Cisco Packet Tracer;
- научиться производить первоначальную настройку коммутатора в программном пакете Cisco Packet Tracer.

3.1. Краткие теоретические сведения

3.1.1. Общие сведения, принцип работы протокола FTP

FTP (File Transfer Protocol, протокол передачи файлов) – является одним из базовых протоколов передачи файлов по сети. Применение данного протокола удобно для загрузки и скачивания файлов большого объема. Принцип работы протокола FTP основан на использовании модели «клиент – сервер».

В передаче файлов по протоколу FTP участвуют два устройства: сервер и клиент. Модель взаимодействия между клиентом и сервером по протоколу FTP показана на рис. 3.1.

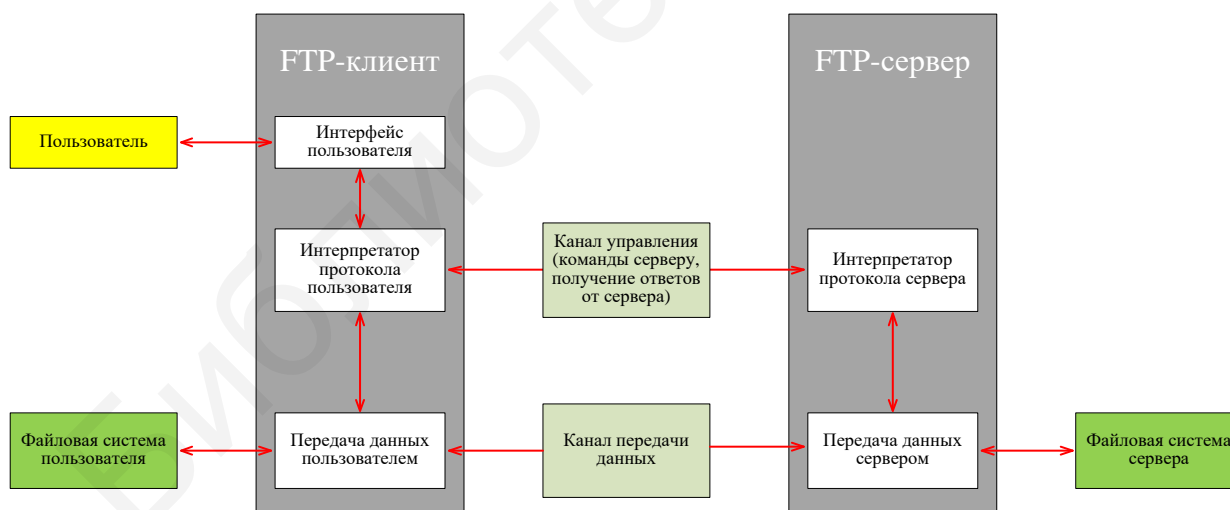


Рис. 3.1. Модель протокола FTP

FTP-сервер – это сервер, который имеет возможность использования протокола передачи файлов и предназначен для хранения файлов. Доступ пользователей к нему обеспечивается при помощи специального программного обеспечения – FTP-клиента.

FTP-клиент – программное обеспечение или приложение, предназначенное для упрощения работы пользователя с FTP-сервером. Оно может быть представлено в виде автономного приложения, веб-клиента, расширений для веб-браузеров или как интерфейс командной строки.

FTP-сервер может поддерживать два типа соединений с FTP-клиентами: активное и пассивное. При использовании активного способа подключения устройств FTP-клиент должен открыть порт и прослушивать его, пока FTP-сервер выполняет активное подключение к нему. При пассивном способе подключения производится обратная процедура – FTP-сервер открывает порт и пассивно прослушивает его. В этом случае клиенты имеют возможность самостоятельно подключиться к серверу.

Протокол FTP относится к протоколам прикладного уровня и для передачи данных использует транспортный протокол TCP. Команды и данные по этому протоколу передаются по разным портам. Порт 20 используется для передачи данных, порт 21 – для передачи команд.

При использовании активного режима клиент инициирует управляющее соединение с портом 21 сервера, передавая команду *«port»*, при помощи которой указывается адрес и порт трансферта информации. После получения этой команды сервер запускает соединение собственного 20-го порта с указанным портом пользователя.

Основным недостатком данного метода является обязательное наличие у пользователя для работы выделенного IP-адреса в сети Интернет. Кроме того, некоторые проблемы могут возникнуть, если клиент находится за брандмауэром либо сеть использует технологию NAT для преобразования сетевых адресов.

При использовании пассивного режима абсолютно все соединения инициируются клиентом. Для того чтобы установить пассивное соединение, пользователь (FTP-клиент) должен передать FTP-серверу специальную команду *«pasv»*. В качестве ответа на эту команду FTP-сервер передает информацию об адресе и номере порта, с которым FTP-клиент должен установить соединение. После получения этих данных пользователь осуществляет подключение к FTP-серверу и проводит прием или передачу информации. При этом пользователь может использовать технологию NAT и брандмауэр, а также не использовать выделенный IP-адрес. Поэтому сегодня в качестве основной разновидности доступа и передачи файлов по протоколу FTP в сети Интернет используется именно пассивный режим. Также использование пассивного способа соединения между FTP-сервером и FTP-клиентом является наиболее предпочтительным и безопасным. Данный способ позволяет избежать внешних подключений из сети.

При использовании брандмауэра и активного режима у пользователей могут возникнуть проблемы с доступом к файлам, хранящимся на внешнем (не находящимся в одной сети с клиентом) FTP-сервере. В случае настройки брандмауэра на отклонение неиницированных изнутри входящих соединений FTP-серверу не удастся установить связь и начать передачу информации. А в связи с тем, что порт, по которому передается информация, динамически выделяется FTP-сервером, появляются и некоторые трудности во время настройки

брандмауэра. Оптимальным вариантом в данном случае является указание диапазона используемых портов и организация специального разрешающего правила брандмауэра для них.

В случае использования пассивного режима с подобной сложностью рискует столкнуться FTP-сервер. При этом можно использовать аналогичное решение – указать в опциях некий диапазон используемых портов и создать для этого диапазона специальное правило.

Для корректного функционирования протокола FTP через технологию NAT и успешной передачи файлов недостаточно просто настроить перенаправление рабочих портов, так как FTP-сервер, работающий из-под NAT, будет передавать внутренний адрес порта, и у клиента просто не получится подключиться и совершить передачу информации.

Некоторые современные реализации технологии NAT могут следить за управляющим каналом FTP-соединения и подменять для нормальной работы передачи данных внутренний адрес внешним. Кроме того, FTP-серверы имеют возможность указывать внешний порт, который должен фигурировать в управляющей сессии.

Чаще всего для нормальной передачи файлов по протоколу FTP через NAT достаточно перенаправления данных, отправляемых по каналу управления по 21-му порту, для реализации управляющей сессии, а также указания и перенаправления диапазона динамических адресов, используемых с целью передачи данных в сеть Интернет.

Алгоритм работы по протоколу FTP заключается в выполнении следующего ряда действий:

1. Пользователь запускает приложение FTP-клиента и устанавливает соединение с FTP-сервером. После установления соединения производится идентификация пользователя путем ввода имени пользователя и пароля.

2. Далее устанавливается управляющее соединение между соответствующими программными модулями клиента и сервера – интерпретаторов протокола.

3. Пользователь через клиентское приложение посылает команды FTP-серверу, которые определяют различные параметры FTP-соединения (например, активный или пассивный режим будет использоваться, номер порта для передачи данных, вид передачи данных, тип передаваемых данных). Кроме этого, по каналу управления передаются команды, которые пользователь будет осуществлять с данными, хранящимися на FTP-сервере.

4. После настройки всех параметров соединения, в зависимости от настроек, один из участников (клиент или сервер), являющийся пассивным, переходит в режим ожидания открытия соединения по порту, указанному для передачи информации. После этого активный участник может установить соединение и начать передавать данные по соответствующему каналу.

5. По завершении передачи данных это соединение закрывается, однако канал управления между интерпретаторами остается открытым. При этом пользователь в рамках той же сессии передачи данных может вновь открыть канал для передачи данных.

Недостатком FTP-протокола является низкая защищенность передаваемых данных. Это связано с тем, что все данные, включая имя пользователя и пароль, передаваемые между клиентом и сервером, передаются в незашифрованном виде, вследствие чего при перехвате трафика в сети злоумышленник может получить доступ к передаваемой информации. Причиной этого является простота использования протокола FTP. Для устранения этой проблемы при использовании FTP-соединения разработаны защищенные версии FTP-протокола, основанные на криптографических протоколах шифрования, таких как SSL (Secure Sockets Layer) и SSH (Secure Shell): FTPS, SFTP, FTP через SSH и др.

FTPS (FTP + SSL) – протокол передачи данных, представляющий собой расширенную версию протокола FTP, который использует шифрование передаваемых данных и команд управления криптографическим протоколом SSL (Secure Sockets Layer – уровень защищенных сокетов) либо же его современным аналогом – протоколом TLS (Transport Layer Security – защита транспортного уровня). При этом существует два метода предоставления безопасности FTP-соединения: явный и неявный.

При использовании неявного метода используется стандартный протокол FTP совместно с протоколами SSL и TSL и используются порты для передачи команд и данных между FTP-клиентом и FTP-сервером, отличные от обычных. Это создает неудобства, поскольку не обеспечивается совместимость FTP-клиентов и FTP-серверов, не поддерживающих FTPS. Поэтому данный метод является устаревшим.

Второй метод, явный, использует команды стандартного протокола FTP, но при этом клиент должен явным образом запросить защищенную передачу данных сервером. Это позволяет сохранить совместимость, поскольку в этом случае применяются одни и те же порты как для FTPS, так и для FTP. При этом для шифрования данных клиентом отправляется команда «AUTH TLS» или «AUTH SSL».

SFTP (SSH FTP) – протокол прикладного уровня для передачи файлов, копирования и выполнения других операций над ними поверх защищенного канала связи. Если FTPS является просто расширением FTP, то SFTP это отдельный и никак не связанный с FTP протокол, который снабжен SSH. В отличие от обычного FTP-протокола он шифрует все команды и данные, защищая передаваемую информацию от открытой передачи через сеть.

FTP через SSH – выполняет обычную сессию передачи данных протоколом FTP через SSH-соединение. Этот метод не является полностью безопасным ввиду того, что если несколько SSH-клиентов устанавливают туннель для управляющего канала, который изначально осуществляется через 21-й порт (а такая ситуация практически всегда и наблюдается), то защищенным окажется именно этот канал. При передаче же данных клиентское программное обеспечение открывает новые TCP-соединения, которые будут находиться уже вне воздействия защитной оболочки SSH.

3.1.2. Работа с межсетевой операционной системой Cisco IOS

в программном пакете Cisco Packet Tracer

Cisco IOS (Internetwork Operating System) – межсетевая операционная система, используемая в сетевом оборудовании компании Cisco (коммутаторах и маршрутизаторах), предназначенная для выполнения задач организации сетевой структуры, маршрутизации трафика в сети, коммутации и передачи данных.

Работа с Cisco IOS осуществляется с использованием интерфейса командной строки при помощи многословных команд. Все команды имеют определенный уровень привилегий (задается в пределах от 0 до 15) и выполняются только в том случае, когда определенный пользователь имеет соответствующий уровень привилегий. Посредством интерфейса командной строки можно произвести настройку разрешенных команд для каждого из уровней.

С использованием интерфейса командной строки осуществляется настройка сетевого оборудования Cisco под нужды конкретной сети.

Сетевое оборудование других производителей также имеет подобного рода интерфейсы управления.

Для настройки сетевого оборудования возможно использование следующих вариантов:

- с использованием консольного подключения по интерфейсам RS-232C, RS-485 или USB;

- с использованием сетевого протокола Telnet (с применением протокола шифрования SSH);

- посредством веб-интерфейса;

- с использованием специализированного программного обеспечения.

Последние три варианта требуют начальной настройки сетевого оборудования через консольное соединение. Рассмотрим процесс настройки коммутатора через консольный порт.

У каждого сетевого коммутатора, маршрутизатора или сетевого экрана присутствует консольный порт (рис. 3.2). Для подключения к коммутатору по консольному порту в программном пакете Cisco Packet Tracer следует соединить компьютер и коммутатор при помощи консольного кабеля (рис. 3.3).

Далее общий алгоритм действий следующий:

- необходимо задать пароль на привилегированный режим командой *enable*;

- создать пользователя (или нескольких пользователей с различными уровнями привилегий);

- установить авторизацию на подключение к консоли;

- задать IP-адрес устройства;

- выбрать тип удаленного подключения (например, Telnet/SSH);

- включить авторизацию для удаленных подключений.

Разъёмы для подключения
консольного кабеля

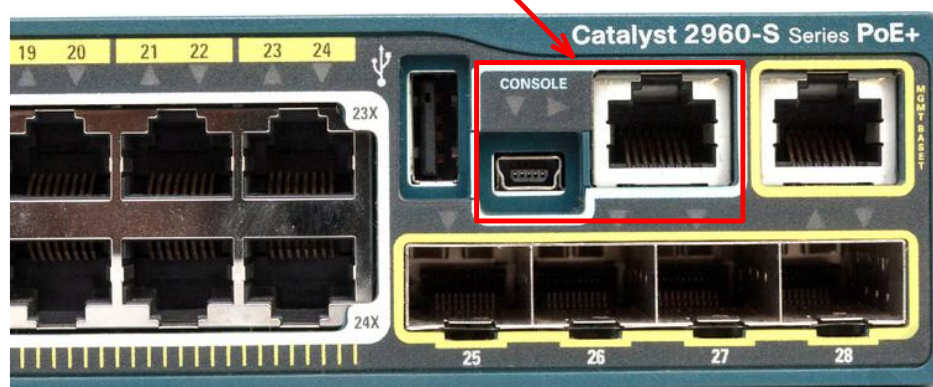


Рис. 3.2. Разъемы на коммутаторе Cisco серии 2960-S для подключения консоли



Рис. 3.3. Выбор консольного подключения на вкладке «Соединения»

Настройка соединения коммутатора с компьютером с использованием последовательного порта (COM-порта или интерфейса RS-232C) в программном пакете Cisco Packet Tracer выполняется следующим образом.

В меню компьютера необходимо перейти на вкладку «Рабочий стол» («Desktop»). На этой вкладке запустить программу Terminal. В открывшемся окне будут отражены текущие параметры последовательного порта (их оставить без изменений) и далее отразится процесс загрузки операционной системы коммутатора Cisco IOS (рис. 3.4).

После загрузки операционной системы можно начинать работу по настройке коммутатора.

Для вызова списка доступных для исполнения команд в терминале следует ввести символ «?» (рис. 3.5).

Доступных команд немного, это связано с тем, что в данный момент времени работа производится в пользовательском режиме. Для полного доступа к командам необходимо войти в привилегированный режим. Выполняется это с помощью команды *enable*. Знак # свидетельствует о том, что мы находимся в привилегированном режиме (рис. 3.6).

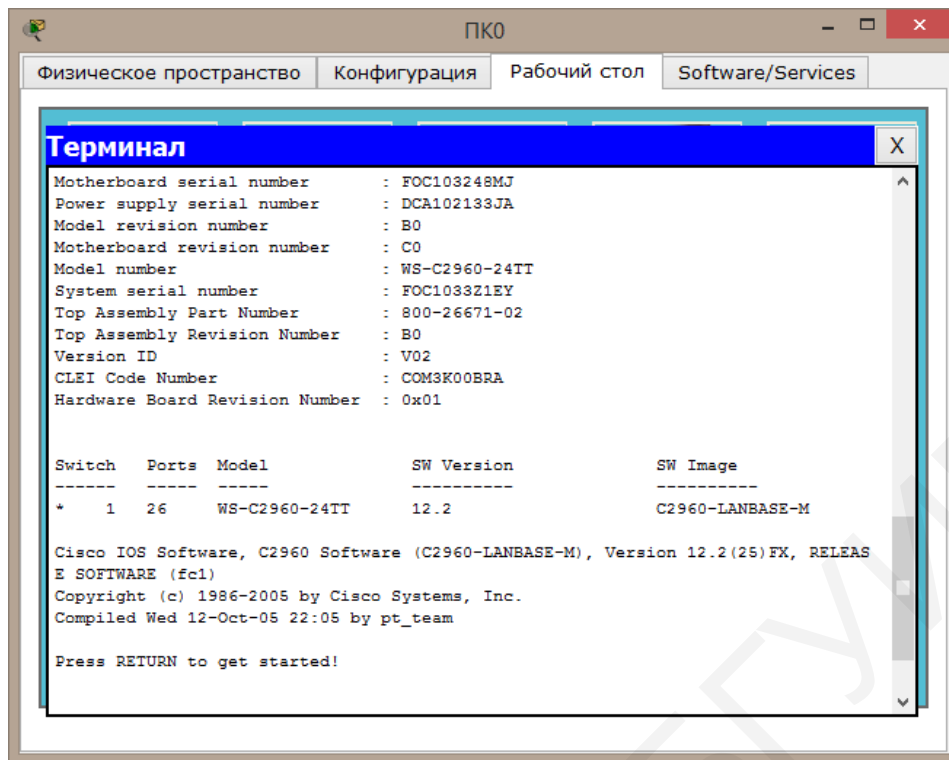


Рис. 3.4. Отображение процесса загрузки операционной системы коммутатора Cisco IOS в окне консоли

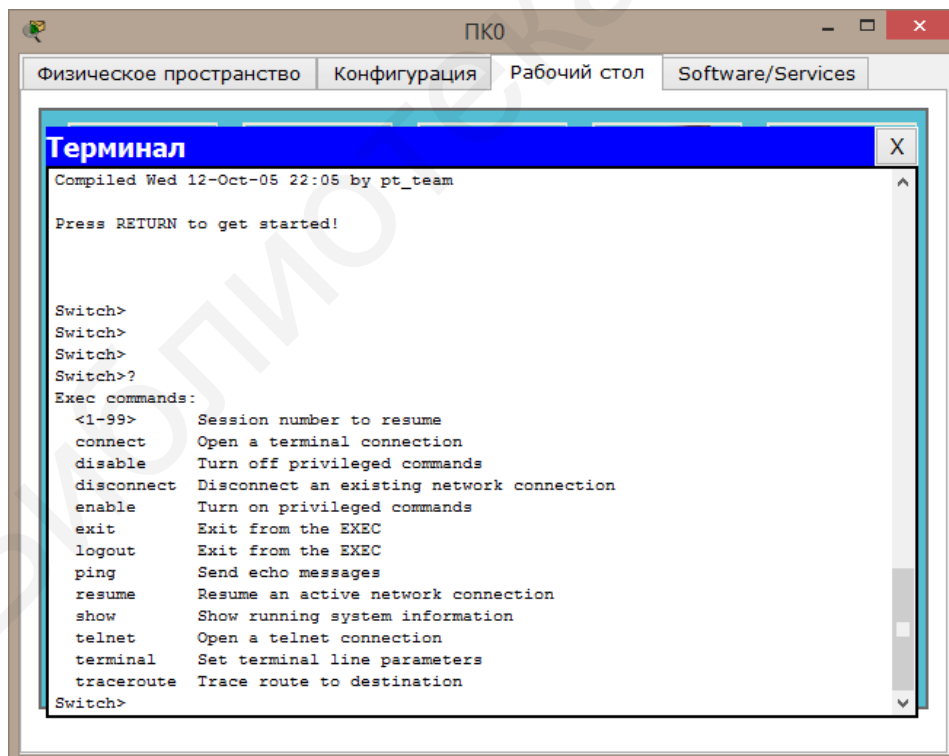


Рис. 3.5. Список доступных команд

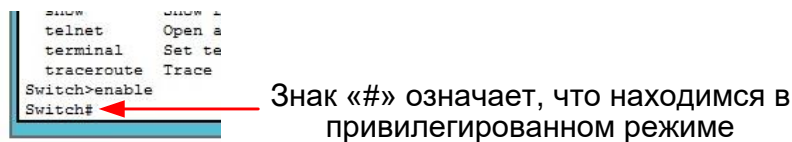


Рис. 3.6. Привилегированный режим работы

Разделение на пользовательский и привилегированный режимы используется для обеспечения безопасности. Часть списка команд управления, доступных в привилегированном режиме, показана на рис. 3.7.

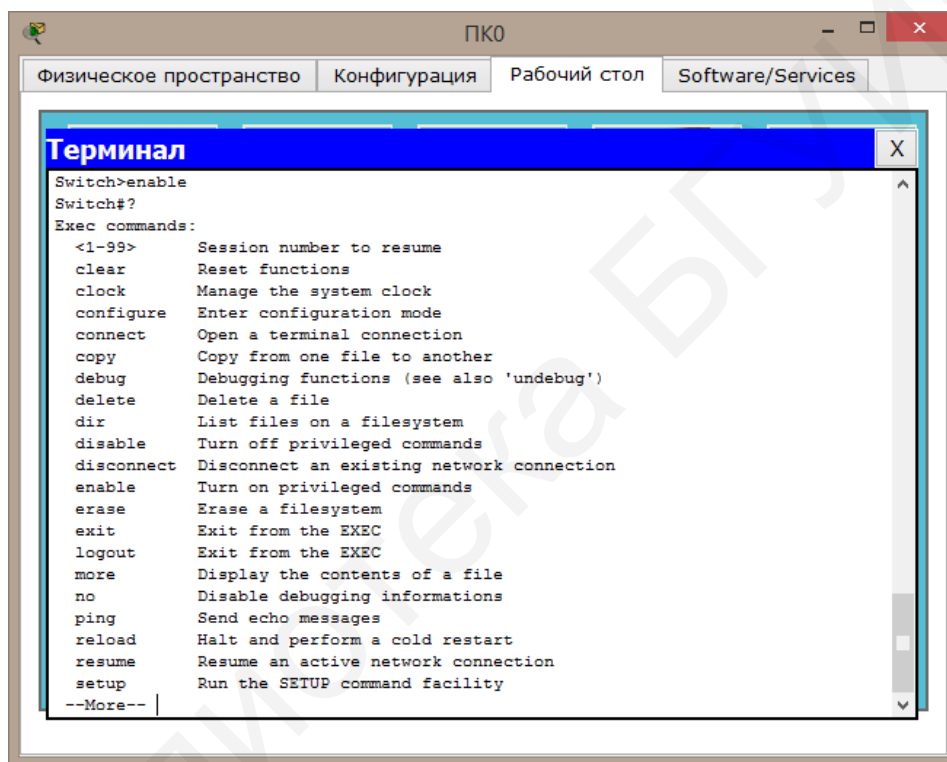


Рис. 3.7. Команды управления, доступные в привилегированном режиме

Для пролистывания страниц используется клавиша «пробел» клавиатуры.

Для возвращения в пользовательский режим используется команда *disable* или команда *exit*.

Автоматическое дописывание команд доступно посредством клавиши табуляции на клавиатуре. Например, после набора *en* и нажатия клавиши TAB Cisco IOS автоматически допишет команду.

Если по нажатии клавиши TAB ничего не происходит, это свидетельствует о наличии нескольких команд, начинающихся на введенные символы. Чтобы увидеть эти команды, необходимо ввести знак вопроса (рис. 3.8).

```
Switch>e
Switch>e?
enable  exit
Switch>e|
```

Рис. 3.8. Отображение списка доступных команд, имеющих одинаковое начало

Находясь в привилегированном режиме, можно посмотреть текущие настройки коммутатора командой *show run* или *show running-config*.

Перед началом настройки необходимо войти в режим глобального конфигурирования с помощью команды *configure terminal* или *conf t* (рис. 3.9).

```
Switch#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#|
```

Рис. 3.9. Переход в режим глобального конфигурирования командой *configure terminal*

Для безопасности необходимо задать пароль для входа в привилегированный режим командой *enable*. После появления приглашения *Password:* требуется ввести пароль (рис. 3.10). Теперь для входа в привилегированный режим необходимо будет вводить пароль.

```
Switch>enable
Password:|
```

Рис. 3.10. Задание пароля для входа в привилегированный режим

Данный пароль хранится в открытом виде и виден при вводе команды *show run*. Для исправления этого необходимо в режиме глобального конфигурирования ввести команду *service password-encryption*. Теперь при вводе команды *show run* пароль будет отображаться в зашифрованном виде. Второй способ – задать пароль в режиме глобального конфигурирования с использованием команды *enable secret <пароль>*, после чего использовать команду *service password-encryption*. При наличии обоих паролей приоритет имеет последний (введенный с командой *enable secret <пароль>*).

Создание пользователей производится с помощью команды *username <имя пользователя> privilege <уровень привилегий 1...15, 15 – самый высокий уровень> password <пароль>*.

После создания локальный пользователь хранится в локальной базе. Необходимо настроить коммутатор таким образом, чтобы при аутентификации локальный пользователь использовал именно локальную базу.

Для этого необходимо зайти в режим конфигурирования терминальных линий командой *line console 0* (рис. 3.11).

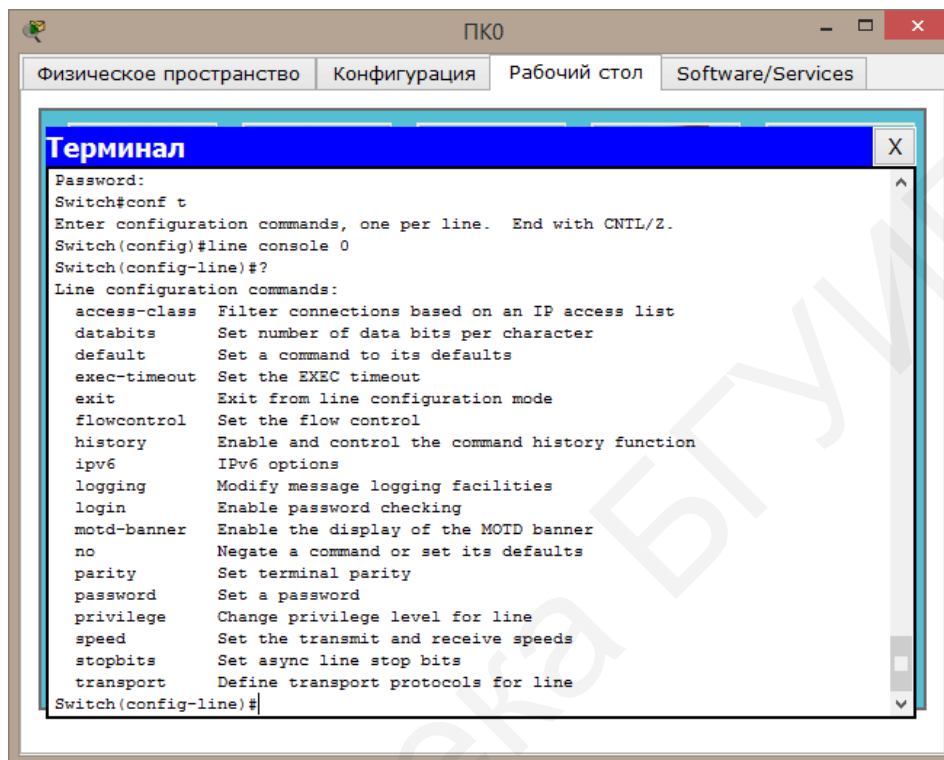


Рис. 3.11. Список команд, доступных в режиме конфигурирования терминальных линий

Для использования локальной базы при аутентификации необходимо ввести команду *login local*.

Для выхода из всех режимов конфигурации используется команда *end*.

Теперь при попытке входа в консоль будет запрос на имя пользователя и пароль.

Далее следует задать IP-адрес коммутатора.

Просмотреть доступные интерфейсы можно с помощью команды *show run*. Доступны физические интерфейсы FastEthernet 0/1...0/24 и GigabitEthernet 1/1...1/2, а также логический интерфейс Vlan1 (Vlan – виртуальная локальная сеть, виртуальные локальные сети более подробно будут рассмотрены в лабораторной работе №4). По умолчанию все порты коммутатора включены во Vlan1. В коммутаторах IP-адреса всегда настраиваются на логических интерфейсах (не на физических).

Для настройки IP-адреса необходимо войти в режим глобального конфигурирования, далее войти в режим конфигурирования интерфейсов (рис. 3.12) командой *interface <указать конфигурируемый интерфейс>*.

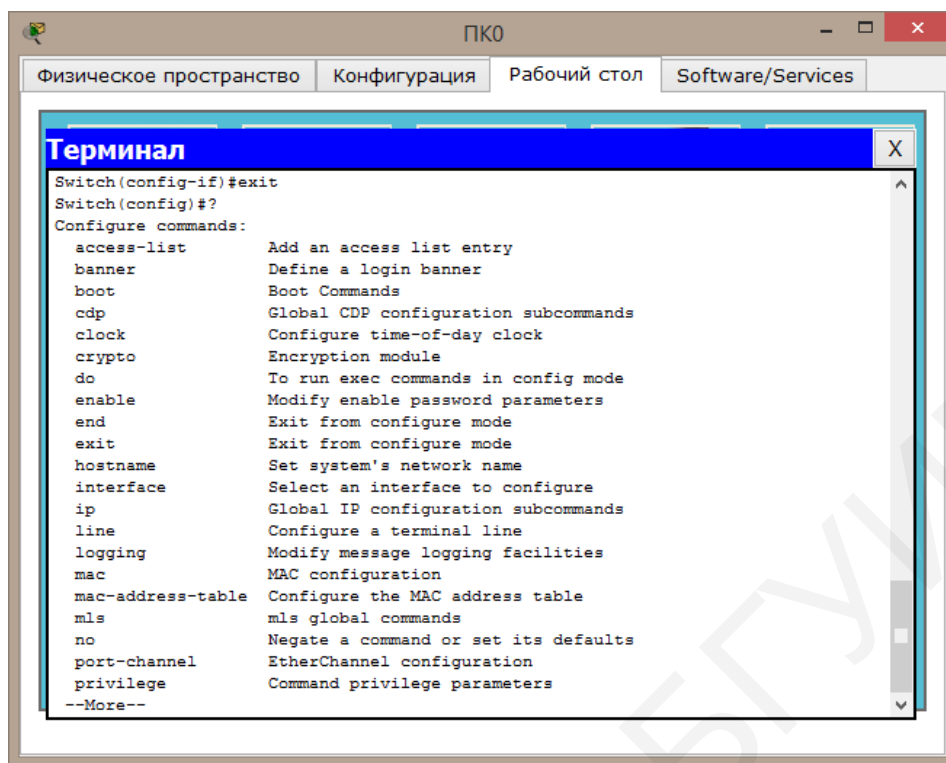


Рис. 3.12. Список команд, доступных в режиме конфигурирования интерфейсов

Задание IP-адреса производится с помощью команды *ip address <IP-адрес и маска сети>*. Далее нужно ввести команду *no shutdown*, которая необходима для включения интерфейса. Выход из режима конфигурирования интерфейсов – командой *exit*.

Далее необходимо настроить виртуальные терминальные линии. Для этого необходимо войти в режим конфигурирования терминальных линий – команда *line vty 0 4* (рис. 3.13).

Затем необходимо определить транспортный протокол с помощью команды *transport input <название протокола>*. В Cisco Packet Tracer возможна работа только по протоколу Telnet.

Далее необходимо задать аутентификацию при входе – *login <название базы данных>*, в данном случае – *local*. Затем следует выйти из режима конфигурирования и сохранить конфигурацию, делается это командой *write memory* или *wr mem*.

Для проверки правильности настроек необходимо произвести подключение компьютера по высокоскоростному соединению FastEthernet (компьютеру назначить IP-адрес из той же подсети, в которую входит коммутатор) и проверить связь командой *ping* и командой *telnet*. При проверке командой *telnet* коммутатор запросит аутентификационные данные, при правильном вводе которых открывается удаленное подключение к коммутатору.

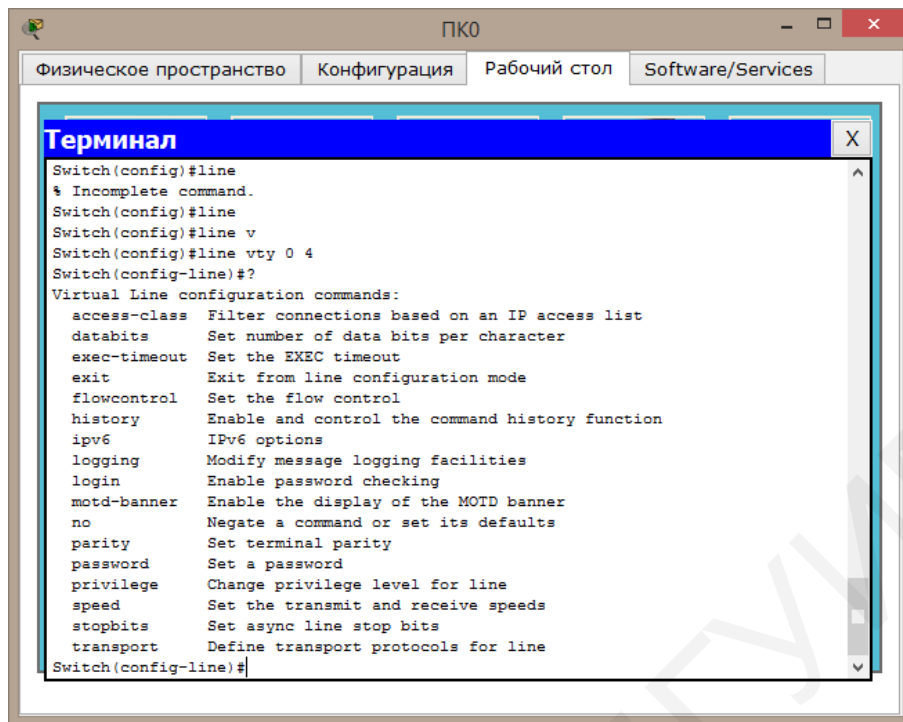


Рис. 3.13. Список команд, доступных в режиме конфигурирования терминальных линий

3.1.3. Работа с FTP-сервером в программном пакете Cisco Packet Tracer

Для настройки FTP-сервера в программном пакете Cisco Packet Tracer следует зайти в меню конфигурации сервера (рис. 3.14) и выбрать *Службы* → *FTP*.

Для удаления существующего пользователя необходимо его выделить в окне 4 и нажать кнопку со значком « – », а для создания нового пользователя необходимо заполнить поля 1–3 и нажать кнопку со значком « + ».

Подключение к FTP-серверу со стороны FTP-клиента производится с использованием интерфейса командной строки. Для этого на компьютере пользователя запустить командную строку (зайти в основное меню, затем перейти на вкладку «Рабочий стол» и там запустить интерфейс командной строки) и в ней ввести команду для подключения к FTP-серверу:

ftp <IP-адрес FTP-сервера> или *ftp* <имя FTP-сервера>.

Основные команды для работы с файлами на FTP-сервере это:

- *dir* – вывести список файлов, хранящихся на сервере;
- *put* <имя файла.расширение> – загрузить файл на сервер;
- *get* <имя файла.расширение> – загрузить файл с сервера;
- *delete* <имя файла.расширение> – удалить файл с сервера;
- *rename* <старое имя.расширение> <новое имя.расширение> – переименовать файл на сервере.

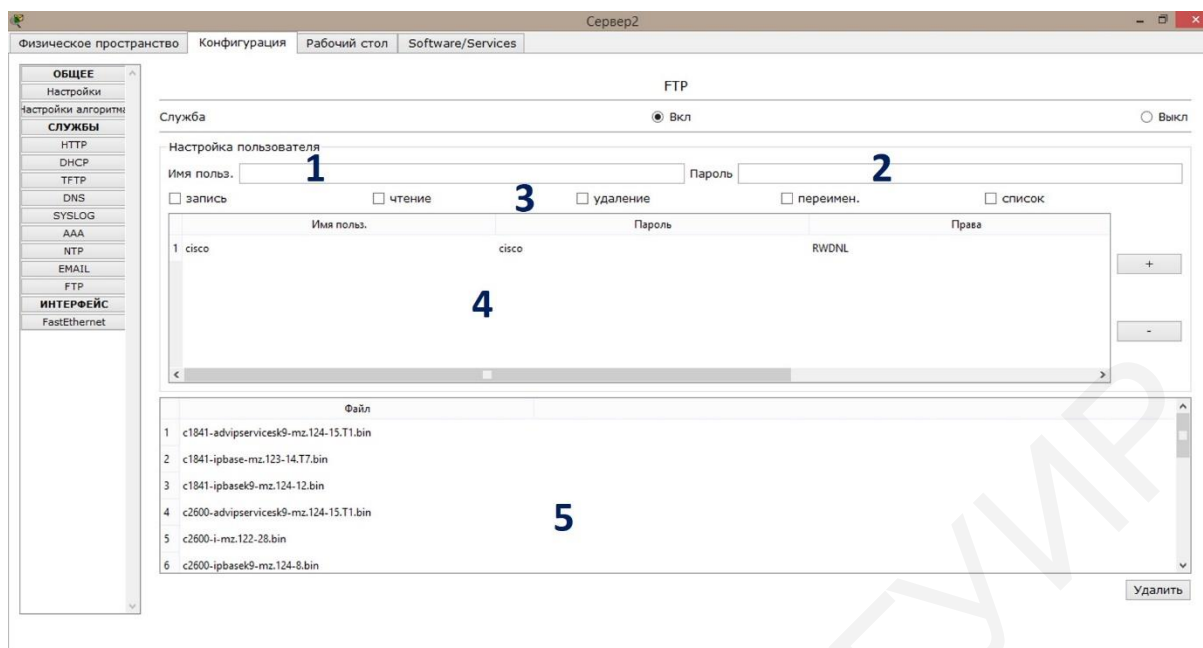


Рис. 3.14. Окно конфигурации FTP-сервера:

1 – поле для ввода имени нового пользователя; 2 – поле для ввода пароля нового пользователя; 3 – набор флажковых переключателей для задания прав пользователей; 4 – поле со списком пользователей FTP-сервера и их параметрами; 5 – поле со списком файлов, находящихся на FTP-сервере

Приведенный список команд является неполным. Данные команды наиболее часто используются при работе с FTP-сервером.

В данной лабораторной работе предполагается, что работа будет осуществляться с текстовыми файлами, имеющими расширение *.txt. Работа с текстовыми файлами производится в «Блокноте». Для запуска приложения следует зайти в основное меню персонального компьютера, далее перейти на вкладку «Рабочий стол» и запустить приложение «Блокнот».

Через данное приложение можно проверить работу с файлами, хранящимися на FTP-сервере.

3.2. Задание для выполнения лабораторной работы

1. Используя сеть передачи данных, построенную в предыдущей лабораторной работе (согласно заданному варианту), в программном пакете Cisco Packet Tracer создать сеть, в состав которой будет входить FTP-сервер (дополнить имеющуюся сеть сервером). Другими словами, в сеть, которая была построена при выполнении предыдущей лабораторной работы, необходимо добавить FTP-сервер.

При настройке сети произвести отключение служб HTTP, DHCP и DNS; настроить IP-адреса всех хостов. FTP-серверу задать имя (допускаются только латинские буквы и цифры).

Полученную схему сети включить в отчет.

2. Настроить коммутатор согласно алгоритму, приведенному в п. 3.1.2. Для этого следует подключить один из компьютеров в данной сети к коммутатору посредством консольного кабеля и, используя интерфейс командной строки (в меню подключенного компьютера выбрать вкладку «Рабочий стол» и запустить программу Terminal), произвести настройку коммутатора.

Проверить прохождение пакетов данных между устройствами. Для этого в дополнительной панели инструментов выбрать «Добавить простой PDU», левой кнопкой мыши указать отправителя и получателя пакета. При правильной настройке коммутатора в окне «Переданные пакеты» будет отображена новая запись со статусом «успешно».

В отчет включить текущие настройки коммутатора и результаты проверки прохождения пакета данных между устройствами в сети.

3. Настроить работу FTP-сервера. Для этого в окне настроек FTP-сервера (см. рис. 3.14) удалить всех имеющихся пользователей и создать новых пользователей, имеющих следующие права.

Пользователь 1 – права на чтение, запись, удаление, переименование и просмотр списка файлов, имеющихся на сервере.

Пользователь 2 – права только на чтение файлов.

Пользователь 3 – права на чтение, переименование и на просмотр списка файлов.

Имена пользователей задаются произвольно (допускаются только латинские буквы и цифры).

4. Проверить связь с FTP-сервером по IP-адресу и по имени сервера. Для этого на одном из компьютеров, входящих в сеть, следует использовать интерфейс командной строки (программа Terminal). Результаты проверки включить в отчет.

5. Создать на одном из компьютеров, являющемся клиентом с правами Пользователя 1, текстовый файл и загрузить его на FTP-сервер. Просмотреть с данного клиента находящиеся на FTP-сервере файлы.

6. Подключиться к FTP-серверу со второго клиента с правами Пользователя 2, вывести список файлов, находящихся на сервере, скопировать файл, ранее загруженный с первого клиента, просмотреть этот файл в текстовом редакторе, изменить и загрузить на сервер (в отчет включить результаты чтения списка файлов, открытого файла, результаты загрузки файла на FTP-сервер).

7. Подключиться к FTP-серверу со второго клиента с правами Пользователя 3, вывести список файлов, находящихся на сервере. Переименовать ранее загруженный текстовый файл и снова вывести список файлов (в отчет включить результат чтения списка файлов).

8. Подключиться к FTP-серверу с первого клиента с правами Пользователя 1, скопировать переименованный файл на компьютер, затем удалить его с

сервера (в отчет включить список файлов, хранящихся на компьютере, и окончательный список файлов, хранящихся на FTP-сервере).

3.3. Содержание отчета

1. Титульный лист.
2. Цель работы.
3. Схема топологии сети.
4. Результаты настройки сетевого коммутатора.
5. Результаты прохождения пакета данных между устройствами сети.
6. Результаты настройки параметров FTP-сервера.
7. Результаты действий с файлами на FTP-сервере по пп. 4–8 подразд. 3.2.
8. Выводы.

5. Контрольные вопросы

1. Что такое FTP-протокол?
2. Поясните модель работы FTP-протокола.
3. Насколько протокол FTP является безопасным?
4. Какие методы используют для повышения безопасности передачи данных с использованием FTP-протокола?
5. Приведите примеры команд управления FTP-сервером из командной строки.
6. Что необходимо для первоначальной настройки коммутатора?
7. Возможна ли первоначальная удаленная настройка коммутатора? Почему?
8. Как можно узнать текущие настройки коммутатора?
9. Режимы настройки коммутатора по уровням доступа: какие бывают, для чего нужны?
10. Каким образом выполняется вход в привилегированный режим? Что свидетельствует о нахождении в данном режиме?

Лабораторная работа №4 ВИРТУАЛЬНЫЕ ЛОКАЛЬНЫЕ СЕТИ

Цель работы:

- изучить, что такое виртуальные локальные сети и необходимость их применения;
- изучить типы виртуальных локальных сетей;
- изучить способы построения сетей передачи данных с применением виртуальных локальных сетей.

4.1. Краткие теоретические сведения

4.1.1. Общие сведения о виртуальных локальных сетях

Локальная вычислительная сеть представляет собой совокупность различных устройств, предназначенных для передачи и приема информации. В настоящее время наиболее распространенным является вариант построения локальной сети на основе технологии Ethernet. Основу построения сети по этой технологии составляет применение оборудования канального уровня в виде коммутаторов. Логика работы коммутатора основана на частичной фильтрации проходящего через него трафика, т. е. информационный пакет, отправленный от источника сообщения на один из портов коммутатора, будет отправлен только лишь на другой порт коммутатора, который соединяет его с адресатом, либо же отправляет его на другой коммутатор или маршрутизатор для последующей поставки конечному адресату. В любом случае прием и отправка информационного пакета, имеющего конкретные адреса источника и получателя, через коммутатор производится только через два порта и изолируется от всего другого трафика, т. е. работа осуществляется по схеме «точка – точка». Однако же в основе работы сети лежит использование различных протоколов, которые при своей логике работы требуют рассылки информационных пакетов на все устройства, подключенные к сети. Иначе говоря, в сети Ethernet существуют широковещательные кадры. Они необходимы для работы многих сетевых протоколов (ARP, BOOTP, DHCP). С их помощью сетевые устройства могут оповещать другие компьютеры о своем появлении в сети. Кроме того, рассылка широковещательных кадров может возникать из-за некорректной работы сетевого адаптера.

Широковещательные кадры могут привести к нерациональному использованию полосы пропускания, особенно в крупных сетях. Для того чтобы этого не происходило, важно ограничить область распространения широковещательного трафика (эта область называется широковещательным доменом) – организовать небольшие широковещательные домены, или виртуальные локальные сети.

Виртуальная локальная сеть, или VLAN (Virtual Local Area Network), – логическая группа узлов сети, трафик в которой, в том числе и

широковещательный, на канальном уровне полностью изолирован от других узлов сети. Это означает, что передача кадров между разными виртуальными сетями на основании MAC-адреса невозможна независимо от типа адреса – уникального, группового, широковещательного. В то же время внутри виртуальной сети кадры передаются по технологии коммутации, т. е. только на тот порт, который связан адресом назначения кадра. Таким образом, с помощью виртуальных сетей решается проблема распространения широковещательных кадров и вызываемых ими последствий вплоть до развития «широковещательных штормов», что существенно снижает производительность сети.

Преимуществами использования VLAN являются:

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

Рассмотрим пример, доказывающий эффективность использования логической сегментации сети с помощью технологии VLAN при решении типовой задачи организации доступа в сеть Интернет сотрудникам офиса (трафик каждого отдела должен быть изолированным).

Предположим, что в офисе имеется несколько комнат, в каждой из которых располагается определенное количество сотрудников. Каждая комната представляет собой отдел (рис. 4.1).

При стандартном подходе к решению задачи с помощью физической сегментации трафика каждого отдела потребовалось бы в каждую комнату устанавливать отдельный коммутатор, который бы подключался к маршрутизатору, предоставляющему доступ в сеть Интернет. При этом маршрутизатор должен обладать достаточным количеством портов, обеспечивающих возможность подключения всех физических сегментов сети. Данное решение плохо масштабируемо и является дорогостоящим, так как при увеличении количества отделов увеличивается количество необходимых коммутаторов, интерфейсов маршрутизатора и магистральных кабелей.

При использовании виртуальных сетей уже не придется подключать пользователей одного отдела к отдельному коммутатору, что позволяет сократить количество используемых устройств и магистральных кабелей. Коммутатор, программное обеспечение которого поддерживает функцию виртуальных локальных сетей, позволяет выполнять логическую сегментацию сети путем соответствующей программной настройки. Это дает возможность подключать пользователей, находящихся в разных сегментах, к одному коммутатору, а также сокращает количество необходимых физических интерфейсов на маршрутизаторе (рис. 4.2).

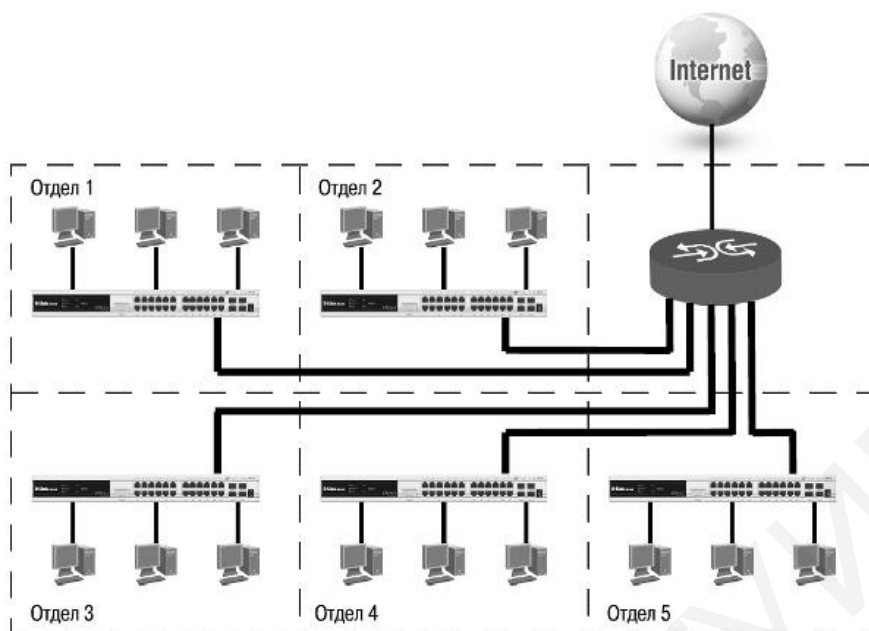


Рис. 4.1. Пример организации локальной вычислительной сети с физической сегментацией

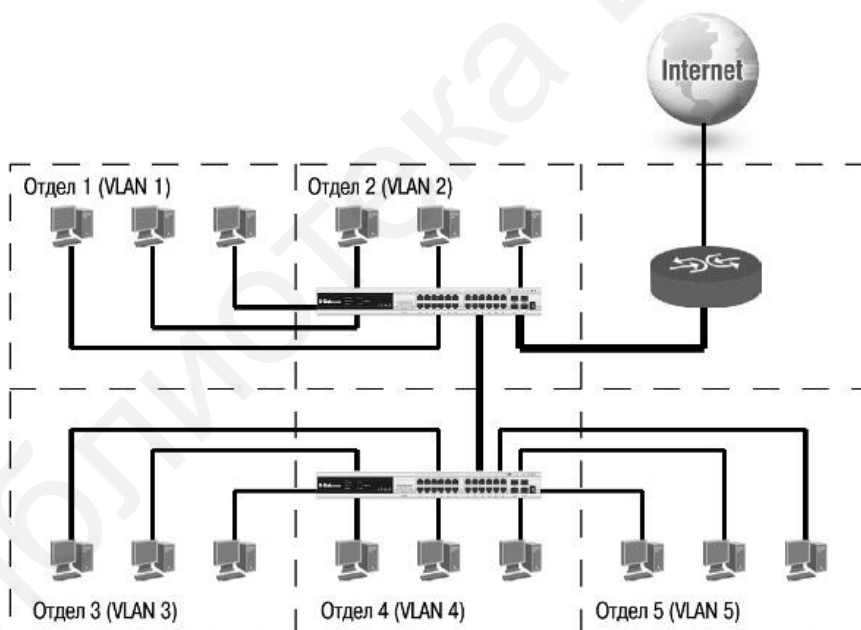


Рис. 4.2. Пример организации локальной вычислительной сети с логической сегментацией

В локальных сетях, построенных на основе коммутаторов, могут быть построены следующие типы виртуальных локальных сетей:

- на основе портов;
- на основе MAC-адресов;
- на основе стандарта IEEE 802.1Q;

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

При построении *виртуальной локальной сети на основе портов* (Port-based VLANs) сетевой администратор назначает каждый порт коммутатора в определенную виртуальную локальную сеть. Например, порты 1...3 коммутатора могут быть определены для VLAN1, 4...6 – для VLAN2, 7...14 – для VLAN3. При приеме пакета коммутатор определяет, к какой VLAN принадлежит порт, на который этот пакет прибыл, и может отправить этот пакет только лишь на те порты коммутатора, которые принадлежат той же VLAN.

Когда компьютер пользователя подключается к другому порту коммутатора, сетевой администратор может просто переназначить новый порт для той же VLAN, в которую входил пользователь. В этом случае сетевые изменения полностью прозрачны для пользователя и администратору не требуется изменять топологию сети. Однако при таком подходе конфигурация портов статическая и может быть изменена только вручную.

Основные достоинства виртуальной локальной сети на основе портов:

- как правило, применяются в пределах одного коммутатора при необходимости организации нескольких рабочих групп в пределах небольшой локальной сети;
- простота настройки, которая заключается в назначении одинакового идентификатора портам, входящим в одну виртуальную локальную сеть;
- возможность изменения логической топологии сети без изменения физической топологии;
- каждый порт коммутатора может принадлежать только лишь одной виртуальной логической сети.

Однако использование виртуальных сетей на основе портов имеет недостаток: для объединения виртуальных сетей как внутри одного коммутатора, так и между несколькими коммутаторами требуется применение оборудования, работающего на сетевом уровне модели OSI, так как в этом случае необходимо иметь один дополнительный порт в каждой виртуальной локальной сети для подключения к маршрутизатору, который хранит таблицу маршрутизации для пересылки пакетов из одной подсети в другую. Для устранения этой проблемы можно использовать коммутаторы, которые на основе реализаций фирменных технологий внутреннего программного обеспечения могут включать порт в несколько виртуальных локальных вычислительных сетей или же использовать коммутаторы третьего уровня (L3 Switch).

Виртуальные локальные сети на основе MAC-адресов (MAC-address-based VLANs) устроены следующим образом. В каждом коммутаторе хранится таблица MAC-адресов подключенных к нему устройств. При использовании этого метода построения виртуальной локальной сети кроме таблицы MAC-адресов коммутатор хранит таблицу соотношений этих адресов с имеющимися виртуальными локальными сетями. Ключевым преимуществом использования данного метода является то, что сетевому администратору не требуется выполнять

переконфигурацию коммутатора при изменении подключения пользователей к другим физическим портам коммутатора. Однако же при этом способе организации виртуальной локальной сети присвоение MAC-адресов может потребовать значительных временных затрат, с одной стороны, а с другой – присвоение отдельных MAC-адресов нескольким виртуальным сетям может быть непростой задачей. Это будет являться существенным ограничением для совместного использования ресурсов, например, сервера, а также между несколькими различными VLAN. Теоретически MAC-адрес может быть присвоен множеству VLAN, однако это может вызвать проблемы с существующей маршрутизацией трафика в сети и ошибки, связанные с таблицами пересылки пакетов в коммутаторе.

Пересылка пакетов между различными VLAN при данном способе организации сети возможна двумя способами:

- с использованием дополнительного сетевого адаптера, подключенного к другой сети;
- с использованием маршрутизатора.

Первый вариант объединения нескольких сетей имеет существенный недостаток, так как при большом количестве устройств в сети, подключаемых к нескольким VLAN, требуется большое число дополнительных сетевых адаптеров, что экономически нецелесообразно.

При втором способе имеются свои ограничения для применения VLAN на основе MAC-адресов, аналогичные предыдущему варианту построения VLAN на основе портов: маршрутизатор должен иметь отдельный порт для каждой подключаемой VLAN. Также при этом нельзя объединять сети в одном сегменте, так как маршрутизатор является устройством третьего уровня модели OSI.

Виртуальные сети на основе стандарта IEEE 802.1Q. Построение VLAN на основе портов основано только на добавлении дополнительной информации к адресным таблицам коммутатора и не использует возможности встраивания информации о принадлежности к виртуальной сети в передаваемый кадр. Виртуальные локальные сети, построенные на основе стандарта IEEE 802.1Q, также как и VLAN на основе портов, работают на втором уровне сетевой модели, но в отличие от VLAN на основе портов используют дополнительные поля кадра Ethernet для хранения информации о принадлежности к VLAN при его перемещении по сети. Эту технологию построения виртуальных локальных сетей используют для создания сетей, охватывающих множество коммутаторов.

Преимуществами применения данного варианта построения являются:

- гибкость и удобство в настройке и применении – возможность создания необходимых комбинаций VLAN как в пределах одного коммутатора, так и в пределах всей сети;
- использование дополнительных полей в кадре Ethernet («тегов») позволяет передавать информацию из одной VLAN через множество коммутаторов по одному физическому соединению (магистральному каналу);
- возможность использования алгоритма связующего дерева Spanning Tree (алгоритма, позволяющего избежать появления замкнутых маршрутов передачи информации в сети) на всех портах и работать в обычном режиме;

– способность добавления и извлечения тегов из заголовков кадров позволяет использовать в сети коммутаторы и сетевые устройства, которые не поддерживают стандарт IEEE 802.1Q;

– возможность использования в сети оборудования разных производителей, поддерживающего стандарт IEEE 802.1Q, независимо от какого-либо фирменного решения;

– для простых случаев, например для организации доступа к серверу из различных VLAN, – возможность обойтись без применения маршрутизатора или коммутатора третьего уровня, если сетевой адаптер поддерживает стандарт IEEE 802.1Q.

Любой порт коммутатора может быть настроен как Tagged (маркированный) или как Untagged (немаркированный). Функция untagging позволяет работать с теми сетевыми устройствами виртуальной сети, которые не понимают тегов в заголовке кадра Ethernet. Функция tagging позволяет настраивать VLAN между несколькими коммутаторами, поддерживающими стандарт IEEE 802.1Q.

Стандарт IEEE 802.1Q определяет изменения в структуре кадра Ethernet, позволяющие передавать информацию о VLAN по сети. На рис. 4.3 изображен формат тега стандарта IEEE 802.1Q. К кадру Ethernet добавлены 32 бита (4 байта), которые увеличивают его размер до 1522 байт. Первые 2 байта (поле Tag Protocol Identifier, TPID) с фиксированным значением 0x8100 определяют, что кадр содержит тег протокола 802.1Q.

Обычный (немаркированный) кадр

Адрес назначения (DA)	Адрес источника (SA)	Данные (Data)	Контрольная последовательность кадра (CRC)
-----------------------	----------------------	---------------	--

Маркированный кадр 802.1p/802.1Q

Адрес назначения (DA)	Адрес источника (SA)	Тег (Tag)	Данные (Data)	Контрольная последовательность кадра (CRC)
-----------------------	----------------------	-----------	---------------	--

Идентификатор протокола тега (TPID) 0x8100	Приоритет (Priority)	Индикатор канонического формата (CFI)	Идентификатор VLAN (VID)
16 бит	3 бита	1 бит	12 бит

Рис. 4.3. Формат кадра Ethernet, содержащий в своем составе тег

Остальные 2 байта содержат следующую информацию:

– Priority (приоритет) – 3 бита поля приоритета передачи кодируют до восьми уровней приоритета (от 0 до 7, где 7 – наивысший), которые используются в стандарте IEEE 802.1p;

– Canonical Format Indicator (CFI), идентификатор канонического формата – 1 бит – зарезервирован для обозначения кадров сетей других типов (TokenRing, FDDI), передаваемых по каналам сетей Ethernet;

– VID (VLAN ID) – 12-битный идентификатор VLAN определяет, какой VLAN принадлежит трафик.

Так как под поле VID отведено 12 бит, то максимальное количество виртуальных локальных сетей составляет 4094 (VID = 0 и VID = 4095 зарезервированы).

Для определения, в какую VLAN необходимо передать кадр информации, каждому физическому порту коммутатора ставится в соответствие некоторый параметр, называемый идентификатором порта виртуальной локальной вычислительной сети – PVID. Этот параметр используется для определения VLAN, в которую коммутатор направит входящий немаркированный (нетегированный) кадр с подключенного к порту сегмента локальной сети. Для этого внутри коммутатора в заголовок всех немаркированных кадров будет добавлен идентификатор VID, равный PVID порта, на который он был принят. Этот механизм позволяет одновременно существовать в одной сети устройствам с поддержкой и без поддержки стандарта IEEE 802.1Q.

Коммутаторы, поддерживающие протокол IEEE 802.1Q, должны хранить таблицу, связывающую идентификаторы портов PVID с идентификаторами VID сети. При этом каждый порт такого коммутатора может иметь только один PVID и столько идентификаторов VID, сколько поддерживает данная модель коммутатора.

Если на коммутаторе не настроены VLAN, то все порты по умолчанию входят в одну VLAN с PVID, равной 1.

Решение о продвижении кадра внутри виртуальной локальной сети принимается на основе трех следующих видов правил:

– правила входящего трафика (ingress rules) – классификации получаемых кадров относительно принадлежности к VLAN;

– правила продвижения между портами (forwarding rules) – принятии решения о продвижении или отбрасывании кадра;

– правила исходящего трафика (egress rules) – принятии решения о сохранении или удалении в заголовке кадра Ethernet тега стандарта IEEE 802.1Q перед его передачей.

Правила входящего трафика выполняют классификацию каждого получаемого кадра относительно принадлежности к определенной виртуальной локальной сети, а также могут служить для принятия решения о приеме кадра для дальнейшей обработки или его отбрасывании на основе формата принятого кадра.

Классификация кадра по принадлежности к определенной виртуальной локальной сети осуществляется следующим образом:

– если кадр не содержит информацию о VLAN (немаркированный кадр), то в его заголовок коммутатор добавляет тег с идентификатором VID, равным идентификатору PVID порта, через который этот кадр был принят;

– если этот кадр содержит информацию о виртуальной локальной сети (маркированный или тегированный кадр), то его принадлежность к конкретной виртуальной локальной сети определяется по идентификатору VID в заголовке кадра (т. е. значение тега в этом кадре не изменяется).

Следует отметить, что внутри коммутатора, поддерживающего работу по стандарту IEEE 802.1Q, все кадры являются маркированными (тегированными).

Правила продвижения между портами осуществляют принятие решения об отбрасывании или передаче кадра на порт назначения на основе его информации о принадлежности конкретной виртуальной локальной сети и MAC-адреса узла-приемника.

Если входящий кадр маркированный, то коммутатор определяет, является ли входной порт членом той же виртуальной локальной сети, путем сравнения идентификатора VID в заголовке кадра и набора идентификаторов VID, ассоциированных с портом, включая его PVID. Если нет, то кадр отбрасывается. Этот процесс называется входной фильтрацией и используется для сохранения пропускной способности внутри коммутатора путем отбрасывания кадров, не принадлежащих той же виртуальной локальной сети, что и входной порт, на стадии их приема. Если кадр немаркированный, входная фильтрация не выполняется.

Далее определяется, является ли порт назначения членом той же виртуальной локальной сети. Если нет, то кадр отбрасывается. Если же выходной порт входит в данную виртуальную локальную сеть, то коммутатор передает кадр в подключенный к нему сегмент сети.

Правила исходящего трафика определяют формат исходящего кадра – маркированный или немаркированный. Если выходной порт является немаркированным (Untagged), то он будет извлекать тег стандарта IEEE 802.1Q из заголовков всех выходящих через него маркированных кадров. Если выходной порт настроен как маркированный (Tagged), то он будет сохранять тег стандарта IEEE 802.1Q в заголовках всех выходящих через него маркированных кадров.

Виртуальные сети на основе стандарта IEEE 802.1ad. Стандарт IEEE 802.1ad является расширением стандарта IEEE 802.1Q и позволяет добавлять в маркированные кадры Ethernet второй тег стандарта IEEE 802.1Q. Эту особенность стандарта называют функцией Q-in-Q или Double VLAN.

Благодаря функции Q-in-Q провайдеры могут использовать их собственные уникальные идентификаторы VLAN (называемые Service Provider VLAN ID или SP-VLANID) при оказании услуг пользователям, в сетях которых настроено несколько VLAN. Это позволяет сохранить используемые пользователями идентификаторы VLAN (Customer VLAN ID или CVLAN ID), и избежать их совпадения, а также изолировать трафик разных клиентов во внутренней сети провайдера. Формат кадра Ethernet, содержащий в своем составе два тега стандарта IEEE 802.1Q, представлен на рис. 4.4.

Обычный (немаркированный) кадр

Адрес назначения (DA)	Адрес источника (SA)	Данные (Data)	Контрольная последовательность кадра (CRC)
-----------------------	----------------------	---------------	--

Кадр с одним тегом 802.1Q

Адрес назначения (DA)	Адрес источника (SA)	Тег (Tag)	Данные (Data)	Контрольная последовательность кадра (CRC)
-----------------------	----------------------	-----------	---------------	--

Кадр с двумя тегами 802.1Q

Адрес назначения (DA)	Адрес источника (SA)	Тег (Tag)	Тег (Tag)	Данные (Data)	Контрольная последовательность кадра (CRC)
-----------------------	----------------------	-----------	-----------	---------------	--

Рис. 4.4. Формат кадра Ethernet, содержащий в своем составе два тега стандарта IEEE 802.1Q

Существует две реализации стандарта IEEE 802.1ad:

- на основе портов – Port-based Q-in-Q;
- выборочная, или избирательная, реализация – Selective Q-in-Q.

При использовании реализации на основе портов Port-based Q-in-Q коммутатор по умолчанию присваивает любому кадру, поступившему на порт доступа граничного коммутатора провайдера, идентификатор SP-VLAN, равный идентификатору PVID порта. Порт маркирует кадр независимо от того, является он маркированным или немаркированным. При поступлении маркированного кадра в него добавляется второй тег с идентификатором, равным SP-VLAN. Если на порт пришел немаркированный кадр, в него добавляется только тег с порта SP-VLAN.

Технология Selective Q-in-Q является более гибкой по сравнению с Port-based Q-in-Q и позволяет производить следующие действия:

- маркировать кадры внешними тегами с различными идентификаторами SP-VLAN в зависимости от значений внутренних идентификаторов CVLAN;
- задавать приоритеты обработки кадров внешних SP-VLAN на основе значений приоритетов внутренних пользовательских CVLAN;
- добавлять к немаркированным пользовательским кадрам помимо внешнего тега SP-VLAN внутренний тег CVLAN.

Базовая архитектура виртуальной локальной сети, построенная на основе стандарта IEEE 802.1ad, работающая по технологии Port-based Q-in-Q, показана на рис. 4.5.

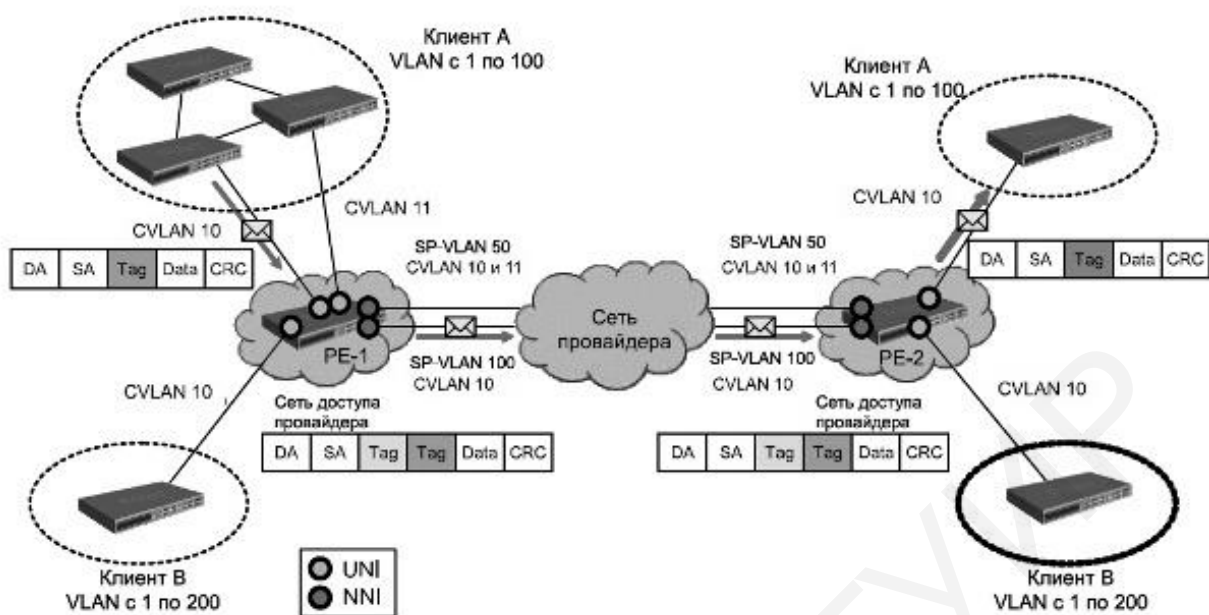


Рис. 4.5. Пример базовой архитектуры виртуальной локальной сети на основе стандарта IEEE 802.1ad

Граничные коммутаторы сети провайдера услуг PE-1 и PE-2 позволяют обрабатывать трафик виртуальных локальных сетей двух подключенных к ним клиентских сетей. Каждому клиенту провайдером присвоен уникальный идентификатор VLAN: SP-VLAN 50 – для клиента А и SP-VLAN 100 – для клиента В. При передаче кадра из клиентской сети в сеть провайдера в его заголовок будет добавляться второй тег 802.1Q: для сети А – SP-VLAN 50, для сети В – SP-VLAN 100. При передаче кадра из сети провайдера в клиентскую сеть второй тег будет удаляться граничным коммутатором.

Все порты граничного коммутатора, на котором используются функции Port-based Q-in-Q или Selective Q-in-Q, должны быть настроены как порты доступа (UNI) или Uplink-порты (NNI):

- UNI (User-to-Network Interface) – порты коммутатора, предназначенные для взаимодействия граничного коммутатора провайдера с клиентскими сетями;
- NNI (Network-to-Network Interface) – порты коммутатора, предназначенные для подключения к внутренней сети провайдера или к другим граничным коммутаторам.

Виртуальные локальные сети на основе стандарта IEEE 802.1v. Данный тип виртуальных локальных сетей также является развитием стандарта IEEE 802.1Q и позволяет объединять узлы в локальной сети в виртуальные локальные сети на основе поддерживаемых ими протоколов (виртуальные локальные сети на основе портов и протоколов).

Для определения членства в виртуальной локальной сети стандарт IEEE 802.1v классифицирует немаркированные кадры по типу протокола и порту, с

которого пришел кадр. Формат тега стандарта IEEE 802.1v аналогичен формату тега стандарта IEEE 802.1Q.

Принцип работы виртуальной локальной сети по этому стандарту следующий. При поступлении на порт коммутатора немаркированного кадра осуществляется проверка заголовка канального уровня и типа протокола вышележащего уровня. Если тип протокола соответствует типу виртуальной локальной сети стандарта IEEE 802.1v на этом порту, то в заголовок кадра добавляется тег с идентификатором VID, равным идентификатору соответствующей виртуальной локальной сети стандарта IEEE 802.1v. Если совпадения не найдены, то в заголовок кадра добавляется тег с идентификатором VID, равным идентификатору входного порта PVID.

Если же на порт коммутатора приходит маркированный кадр, то значение тега VLAN в нем не изменяется.

Внутри коммутатора все кадры являются маркированными. Передача кадров осуществляется на основе таблицы VLAN путем сравнения значений идентификаторов VID. Если порт назначения является членом той же виртуальной локальной сети, что и входной порт, то он передает кадр в подключенный к нему сегмент сети. В противном случае кадр отбрасывается.

Для выходных портов действуют такие же правила, как для стандарта IEEE 802.1Q.

Механизм классификации стандарта IEEE 802.1v требует, чтобы на коммутаторе были настроены группы протоколов. Каждый протокол в группе определяется типом кадра (Ethernet II, IEEE 802.3 SNAP или IEEE 802.3 LLC) и значением поля идентификации протокола в нем. Порт может быть ассоциирован с несколькими группами протоколов, что позволяет классифицировать поступающие немаркированные кадры по принадлежности к разным виртуальным локальным сетям в зависимости от их содержимого. Одна и та же группа протоколов может быть ассоциирована с разными портами коммутатора, при этом на каждом входном порте ей должны быть присвоены уникальные идентификаторы VLAN.

Асимметричные виртуальные локальные сети. Асимметричные виртуальные локальные сети представляют собой разновидность виртуальных локальных сетей, в которых прием и передача трафика между абонентами производится по различным правилам.

4.1.2. Построение виртуальных локальных сетей на основе нескольких коммутаторов

Зачастую в состав локальной сети входит не один, а несколько коммутаторов (рис. 4.6). В этом случае при создании виртуальных локальных сетей на основе портов или MAC-адресов на каждом из коммутаторов следует выделить дополнительную пару портов, принадлежащую к одноименным виртуальным локальным сетям. Количество дополнительно выделенных пар портов должно соответствовать количеству виртуальных локальных сетей. Кроме этого, при соединении виртуальных локальных сетей через маршрутизатор для каждой

виртуальной сети необходимо иметь отдельный порт для его подключения. Это приводит к неэффективному использованию оборудования и кабельных линий связи при большом количестве виртуальных локальных сетей, а также увеличивает экономические затраты на создание и эксплуатацию локальной сети.

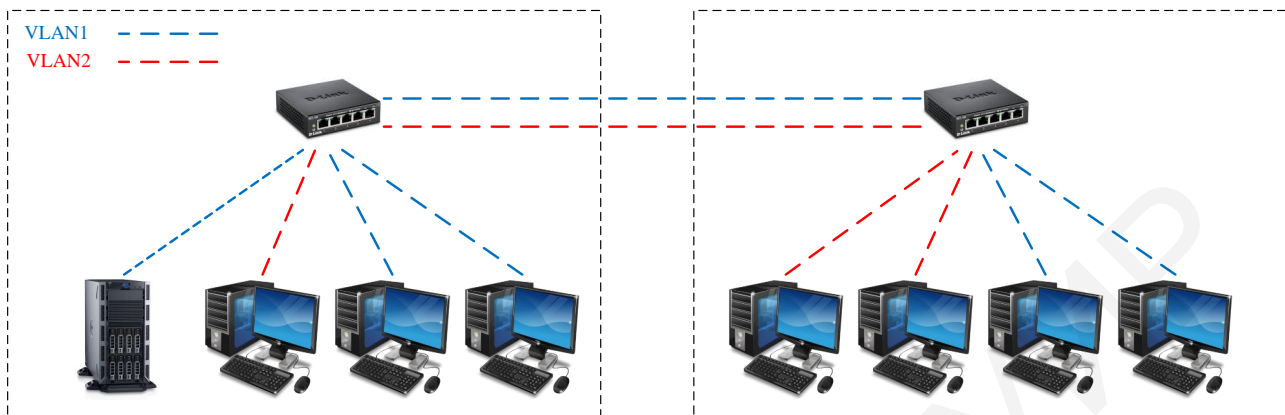


Рис. 4.6. Построение виртуальных сетей на основе нескольких коммутаторов на основе портов

Виртуальные локальные сети, построенные на основе стандарта IEEE 802.1Q, позволяют упростить создание и конфигурирование виртуальных локальных сетей.

Согласно принципу, представленному на рис. 4.6, в виртуальных локальных сетях для соединения нескольких коммутаторов между собой задействуют несколько физических портов. Совокупность физических каналов между двумя устройствами может быть заменена одним агрегированным логическим каналом, получившим название транк (trunk), или транковое соединение (рис. 4.7).

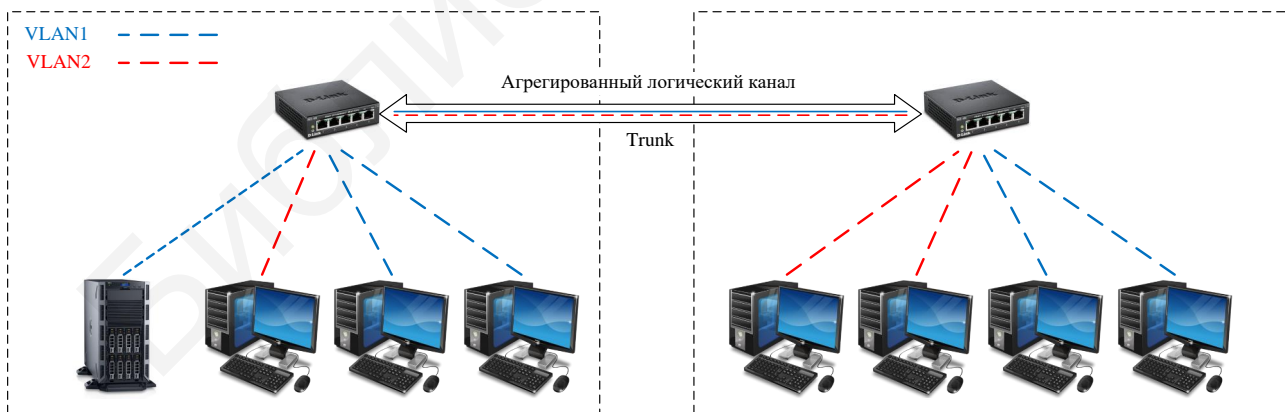


Рис. 4.7. Соединение коммутаторов при помощи транкового соединения

Транковые соединения используются и для подключения маршрутизатора к коммутатору или для подключения высокоскоростных серверов. При этом на

интерфейсе маршрутизатора формируются несколько субинтерфейсов (по количеству виртуальных сетей).

Пропускная способность агрегированного логического канала равна сумме пропускных способностей физических каналов.

Настройка оборудования (коммутаторов) в этом случае будет дополнительно включать процесс конфигурирования портов коммутатора. Для этого в настройках порта в этом случае нужно выбрать один из вариантов:

- Access port – порт доступа;
- Trunk port – порт, передающий маркированный (тегированный) трафик одной или нескольких виртуальных сетей.

В случае выбора access port порт коммутатора будет настроен на работу с окончательным оборудованием. Трафик между этим портом коммутатора и оборудованием – немаркированный (нетегированный).

Во втором случае работа может осуществляться между двумя коммутаторами или коммутатором и маршрутизатором, а трафик будет содержать теги.

4.1.3. Алгоритм работы с виртуальными локальными сетями в программном пакете Cisco Packet Tracer

В операционной системе коммуникационного оборудования Cisco IOS настройку виртуальных локальных сетей можно произвести двумя способами:

- в режиме «Config-vlan»;
- в режиме «VLAN database».

Алгоритм конфигурации для режима «Config-vlan» выглядит следующим образом:

1. Войти в режим глобального конфигурирования с помощью команды «*configure terminal*».
2. Создать виртуальную локальную сеть с помощью команды «*vlan <номер VLAN>*».
3. Перейти в режим конфигурирования интерфейса посредством выполнения команды «*interface [тип интерфейса] mod/port*».
4. Далее следует выполнить конфигурацию режима работы порта с помощью команды «*switchport mode access*».
5. Настроить принадлежность интерфейсов и виртуальной локальной сети с помощью команды «*switchport access <vlan-id>*».
6. Проверить конфигурацию оборудования, выполнив команду «*show vlan*».
7. Выйти из режима конфигурирования VLAN с помощью команды «*end*» или «*exit*».

Конфигурация виртуальных локальных сетей в режиме «VLAN database» производится по следующему алгоритму:

1. Для входа в режим редактирования «VLAN database» выполнить команду «*vlan database*».

2. Выполнить команду «*vlan <vlan-id> name <vlan name>*» для добавления новой виртуальной локальной сети и назначения ей номера и имени.

3. Применить изменения конфигурации оборудования с помощью команды «*apply*», после чего выполнить выход из режима конфигурации командой «*exit*».

4. Перейти в режим конфигурирования интерфейса посредством выполнения команды «*interface [тип интерфейса] mod/port*» и выполнить конфигурацию соответствующего интерфейса.

5. Выполнить конфигурацию режима работы порта с помощью команды «*switchport mode access*».

6. Настроить принадлежность интерфейсов и виртуальной локальной сети с помощью команды «*switchport access <vlan-id>*».

7. Проверить конфигурацию оборудования, выполнив команду «*show vlan*».

Второй способ конфигурации оборудования в настоящее время используется редко, ввиду того что не все коммутаторы поддерживают этот способ настройки.

Настройку подключенных интерфейсов можно выполнить следующим способом:

1. Войти в режим конфигурации интерфейса с помощью команды «*interface <название интерфейса>*», например *interface fastEthernet 0/1* или *intfa 0/1*.

2. Определить режим функционирования выбранного порта (*access* или *trunk*) командой «*switchport mode <режим функционирования>*».

3. Определить принадлежность порта к VLAN с помощью команды «*switchport access vlan <номер VLAN>*».

4. Сохранить настройки интерфейсов.

5. Выйти из режима настроек порта с помощью команды «*exit*» и выйти из режима глобального конфигурирования (в привилегированный режим) – команда «*end*».

Просмотреть существующие VLAN и интерфейсы, определенные к ним, можно с помощью команды «*show vlan*» или «*show vlanbrief*».

Просмотреть таблицу MAC-адресов интерфейсов коммутатора можно посредством выполнения команды «*show mac address-table*» (выполняется в привилегированном режиме).

С помощью команды «*show run*» можно посмотреть текущие настройки коммутатора.

Для создания сети с несколькими коммутаторами на обоих коммутаторах должны быть созданы виртуальные локальные сети с соответствующими номерами и названиями.

Для определения порта в качестве транкового необходимо определить режим его функционирования как *trunk*. Далее необходимо определить, трафик из каких виртуальных сетей необходимо передавать через данный порт с помощью

команды `switchport trunk allowed vlan <указать номера VLAN через запятую>`».

4.2. Задание для выполнения лабораторной работы

Выполнение лабораторной работы состоит из двух этапов. В начале требуется организовать локальную сеть с использованием одного коммутатора и на ее базе построить виртуальные локальные сети. На втором этапе выполнения лабораторной работы требуется расширить полученную сеть путем добавления еще одного коммутатора и нескольких компьютеров и исследовать работу виртуальных сетей с применением транкового канала связи между коммутаторами.

Для реализации первого этапа лабораторной работы следует использовать локальную сеть передачи данных, полученную и настроенную в лабораторных работах №2 и 3. В составе сети должен быть один коммутатор.

1. Используя сеть, построенную в лабораторной работе №2, произвести первоначальную настройку статических IP-адресов оборудования.

2. Используя консольное подключение или веб-интерфейс, произвести настройку коммутатора. На коммутаторе создать три виртуальных локальных сети (не менее двух хостов на сеть). Определить интерфейсы, к которым подключены хосты, в эти виртуальные локальные сети согласно алгоритму действий, изложенному в п. 4.1.3.

Следует отметить, что для названия сетей используются символы латинского алфавита и цифры.

3. Проверить связь между хостами одной виртуальной локальной сети и между хостами, принадлежащими различным виртуальным сетям, используя отправку простого пакета данных или команду *ping*.

4. Просмотреть список существующих VLAN на коммутаторе и интерфейсов, назначенных им при помощи команды *show vlan* или *show vlanbrief*.

5. Просмотреть таблицу MAC-адресов интерфейсов коммутатора и текущие настройки коммутатора с помощью команд *show mac address-table* и *show run* соответственно.

6. Исследовать работу виртуальных локальных сетей, основанных на нескольких коммутаторах. Для этого добавить в существующую сеть еще один коммутатор и минимум шесть хостов и произвести настройку IP-адресов.

7. На втором коммутаторе настроить виртуальные сети, аналогичные созданным на первом коммутаторе. В каждую сеть определить минимум по два хоста из числа подключенных хостов к коммутатору.

8. Настроить между коммутаторами транковое соединение общее для всех виртуальных локальных сетей. Названия виртуальных локальных сетей на обоих коммутаторах должны совпадать для обеспечения правильного функционирования сети. Для транкового соединения следует использовать самый производительный интерфейс.

Для настройки транкового соединения необходимо настроить режим работы порта коммутатора как *trunk* и определить с помощью команды *switchport trunk allowed vlan*, трафик каких VLAN будет проходить через этот порт.

9. Проверить связь между хостами одной VLAN и между хостами, принадлежащими различным VLAN (хосты подключены к различным коммутаторам).

10. Просмотреть существующие на коммутаторах VLAN и интерфейсы, определенные к ним.

11. Просмотреть таблицу MAC-адресов интерфейсов второго коммутатора и текущие настройки коммутатора.

12. Исключить одну из виртуальных сетей из настроек транкового порта.

13. Проверить связь между хостами, включенными в данную VLAN и подключенными к различным коммутаторам, выполнить пп. 10 и 11.

4.3. Содержание отчета

1. Титульный лист.

2. Цель работы.

3. Схема топологии сети с использованием одного коммутатора.

4. Конфигурация коммутатора (списки настроенных виртуальных локальных сетей, интерфейсов, таблица MAC-адресов).

5. Результаты прохождения пакетов данных между хостами одной VLAN и между хостами, принадлежащими различным VLAN.

6. Схема топологии сети с использованием двух коммутаторов.

7. Конфигурация второго коммутатора (списки настроенных виртуальных локальных сетей, интерфейсов, таблица MAC-адресов).

8. Результаты прохождения пакетов данных между хостами одной VLAN и между хостами, принадлежащими различным VLAN (хосты должны быть подключены к различным коммутаторам).

9. Конфигурация второго коммутатора после исключения одной виртуальной сети (списки настроенных виртуальных локальных сетей, интерфейсов, таблица MAC-адресов).

10. Результаты прохождения пакетов данных между хостами одной VLAN и между хостами, принадлежащими различным VLAN (хосты должны быть подключены к различным коммутаторам).

11. Выводы.

4.4. Контрольные вопросы

1 Что такое виртуальная локальная сеть?

2. Какие бывают типы виртуальных локальных сетей?
3. Что такое тегированный (маркированный) кадр Ethernet?
4. Что такое двойное тегирование?
5. Как производится настройка виртуальных локальных сетей на основе одного коммутатора?
6. Какие бывают режимы настройки портов коммутатора?
7. Что такое транковое соединение (транковый канал)?
8. Как производится настройка транкового порта на коммутаторе?
9. Что такое порт доступа и как производится его настройка?

Библиотека БГУИР

Лабораторная работа №5

МАРШРУТИЗАЦИЯ ТРАФИКА С ИСПОЛЬЗОВАНИЕМ КОММУТАТОРОВ ТРЕТЬЕГО УРОВНЯ

Цель работы:

- ознакомиться с основами маршрутизации трафика в локальных вычислительных сетях;
- ознакомиться с оборудованием, предназначенным для маршрутизации трафика на третьем уровне модели взаимодействия открытых систем;
- научиться производить конфигурирование сетевого оборудования третьего уровня.

5.1. Краткие теоретические сведения

5.1.1. Общие сведения о маршрутизации. Маршрутизация трафика в сети с использованием коммутаторов третьего уровня

Маршрутизация сетевого трафика – это процесс определения маршрута передачи данных в вычислительных сетях. Маршрутизацию сетевого трафика выполняют специальные сетевые устройства, называемые маршрутизаторами. Они осуществляют решение задачи маршрутизации и перенаправляют пакеты передаваемых данных между сетями.

Изначально построение локальных вычислительных сетей базируется на основе сетевых коммутаторов, работающих на втором (канальном) уровне. Коммутатор второго уровня обеспечивает прямую передачу данных между двумя устройствами в локальной сети. Маршрутизация трафика в этом случае производится на основе MAC-адресов.

Как только в локальной вычислительной сети появляется хотя бы два сегмента (организованы виртуальные локальные сети), то сразу появляется необходимость использования маршрутизации трафика. Маршрутизация трафика производится на третьем уровне модели взаимодействия открытых систем. Для маршрутизации трафика на данном уровне применяются маршрутизаторы (Router) и коммутаторы третьего уровня (L3 Switch).

Изначально у этих двух устройств различное предназначение.

Маршрутизатор предназначен для подключения локальной сети (LAN – Local Area Network) к глобальной компьютерной сети (WAN – Wide Area Network), т. е. осуществляет маршрутизацию трафика во «внешний мир» (сеть Интернет, филиалы, удаленные сотрудники) и обратно.

Коммутатор третьего уровня (L3 Switch) – это прежде всего устройство для локальной вычислительной сети, т. е. данный коммутатор должен маршрутизировать трафик в локальной сети между существующими сегментами. Обычно он используется на уровне распределения (Distribution Layer) в иерархической модели сети.

Может возникнуть вопрос: зачем нужен коммутатор третьего уровня, если его функции может выполнять маршрутизатор?

Исторически первым способом для решения задачи маршрутизации пакетов между виртуальными локальными сетями (VLAN) было использование маршрутизаторов с несколькими физическими интерфейсами. Для маршрутизации трафика каждый физический интерфейс маршрутизатора в этом случае подключается к отдельной виртуальной локальной сети. Этому интерфейсу присваивается IP-адрес той подсети, которая соответствует подключенной VLAN. В этой конфигурации сетевые устройства используют маршрутизатор в качестве шлюза для доступа к устройствам, подключенным к другим виртуальным локальным сетям.

Этот метод маршрутизации трафика в локальной сети на данный момент является неэффективным ввиду существенных ограничений, вызванных тем, что маршрутизаторы не имеют большого числа физических интерфейсов для подключения к разным виртуальным локальным сетям. По мере роста числа виртуальных локальных сетей в сети растет также и число задействованных интерфейсов маршрутизатора. Поэтому в больших сетях используют транковые каналы и подынтерфейсы для связи коммутаторов второго уровня и маршрутизаторов. Такой метод маршрутизации трафика в сети получил название «router-on-a-stick».

Использование транкового канала позволяет передавать трафик от нескольких VLAN по одному физическому интерфейсу. Ограничение количества используемых физических интерфейсов на маршрутизаторе в этом случае устраняется путем создания виртуальных подынтерфейсов. Каждому такому подынтерфейсу назначаются свои уникальные IP-адрес и маска подсети в соответствии с его подсетью или сетью VLAN. Таким образом, маршрутизатор может отделять трафик для каждого подынтерфейса по мере прохождения по транковому каналу обратно в коммутатор второго уровня.

По логике работы метод «router-on-a-stick» мало чем отличается от предыдущего метода маршрутизации на основе физических интерфейсов. Отличие состоит лишь в том, что вместо физических интерфейсов работа ведется с виртуальными интерфейсами одного физического канала связи.

Однако при росте трафика в локальной сети использование маршрутизаторов в сети становится невыгодным ввиду того, что они предназначены в первую очередь для маршрутизации трафика между высокоскоростной локальной сетью и низкоскоростной распределенной сетью (WAN), т. е. для организации связи с внешними сетями. У обычных маршрутизаторов процесс маршрутизации пакетов, передаваемых между сетями, организован программным способом, что сильно сказывается на быстродействии. Поэтому для маршрутизации трафика в локальных сетях используются коммутаторы третьего уровня, или так называемые маршрутизирующие коммутаторы, которые коммутируют маршрутизаторы или многоуровневый коммутатор.

По своей сути коммутатор третьего уровня является вариантом сетевого оборудования, которое совмещает функции канального и сетевого уровней модели OSI, и предназначен для маршрутизации пакетов на больших скоростях в

пределах локальной вычислительной сети.

Отличием коммутатора третьего уровня от коммутатора второго уровня является наличие функции маршрутизации. Коммутатор второго уровня работает только с MAC-адресами, игнорируя элементы пакетов более высоких уровней.

Коммутатор третьего уровня позволяет осуществлять статическую и динамическую маршрутизацию. Если коммутатор может выполнять только статическую маршрутизацию, то он обычно называется коммутатором уровня 2+ (Layer 2+ Switch или L2+ Switch) или коммутатором третьего уровня с ограниченными функциями (Layer 3 Lite Switch, L3 Lite Switch).

С функциональной точки зрения коммутаторы третьего уровня представляют собой маршрутизаторы с высоким быстродействием. При обработке пакета они выполняют те же самые действия: используя информацию третьего уровня, определяют лучший путь передачи пакета, с помощью контрольной суммы проверяют целостность пакета и т. д. В то же время такие устройства полностью совместимы с традиционными маршрутизаторами и могут взаимодействовать с ними по стандартным протоколам.

В отличие от маршрутизаторов коммутаторы третьего уровня строятся на основе распределенной архитектуры, когда обработкой отдельных кадров и пакетов занимается не один процессор, а несколько специализированных. Организация управления производится при помощи отдельного управляющего процессора.

Каждый порт коммутатора третьего уровня имеет собственный специализированный процессор, реализованный в виде одной или нескольких интегральных микросхем специализированного применения (ASIC – Application-specific integrated circuit), предназначенных для обработки передаваемых пакетов. В этом случае маршрутизация производится на аппаратном уровне. На программном уровне остается только обработка данных, не связанных напрямую с обработкой трафика: расчет таблиц маршрутизации, списков доступа и др.

Ввиду того что коммутатор третьего уровня производит анализ заголовков IP-пакетов, можно устанавливать гибкую политику в локальной сети. Последняя предусматривает такие особенности обработки потока информации в локальной сети, как классы и качество обслуживания. С помощью коммутаторов третьего уровня можно устанавливать приоритеты для трафика, выделять определенную ширину полосы пропускания и назначать величину задержки распространения конкретного вида трафика.

В отличие от традиционных маршрутизаторов, которые определяют конкретную подсеть только для одного порта, коммутаторы третьего уровня позволяют выделить в отдельную подсеть каждый порт коммутатора. Маршрутизация в коммутаторах третьего уровня осуществляется над уровнем коммутации, что обеспечивает более гибкую и масштабируемую сетевую архитектуру.

5.1.2. Построение сетей передачи данных с коммутатором третьего уровня и маршрутизатором в программном пакете Cisco Packet Tracer

Настройка коммутатора третьего уровня в программном пакете Cisco Packet Tracer в принципе не отличается от настройки коммутатора второго уровня, за исключением особенностей, о которых будет сказано далее.

Программный пакет Cisco Packet Tracer содержит одну модель коммутатора третьего уровня Cisco Catalyst 3560-24PS (рис. 5.1).



Рис. 5.1. Внешний вид коммутатора третьего уровня Cisco Catalyst 3560-24PS

Вначале рассмотрим вариант организации локальной сети с несколькими виртуальными сетями с использованием прямых подключений к коммутатору третьего уровня (рис. 5.2).

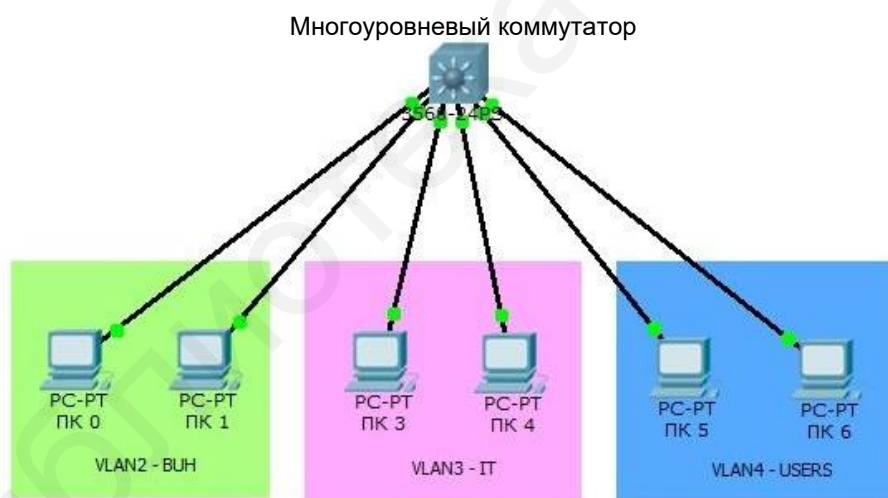


Рис. 5.2. Топология сети с прямым подключением хостов к коммутатору третьего уровня

В процессе настройки IP-адресов сети, помимо задания IP-адреса и маски сети, необходимо в настройках указать адрес шлюза по умолчанию (Default Gateway), на который будут передаваться пакеты для дальнейшей передачи в другие сети при отсутствии в подсети искомого адресата. В качестве шлюза необходимо указывать IP-адрес интерфейса, принадлежащего оборудованию, на котором будет производиться маршрутизация трафика.

Создание виртуальных локальных сетей и назначение им физических интерфейсов осуществляется таким же образом, как в случае использования коммутатора второго уровня. Каждому виртуальному интерфейсу необходимо назначить IP-адрес.

На данном этапе маршрутизация внутри коммутатора не производится и коммутатор выполняет роль обычного коммутатора второго уровня. Для ее включения необходимо в режиме глобального конфигурирования ввести команду «*ip routing*». Данная команда предназначена для включения автоматической маршрутизации трафика между виртуальными сетями.

Конфигурация оборудования в конечном итоге должна быть сохранена в памяти коммутатора.

При организации сети, имеющей в своем составе несколько коммутаторов второго уровня, соединенных с использованием агрегированного канала (транкового соединения), включение коммутатора третьего уровня для маршрутизации производится, как показано на рис. 5.3.

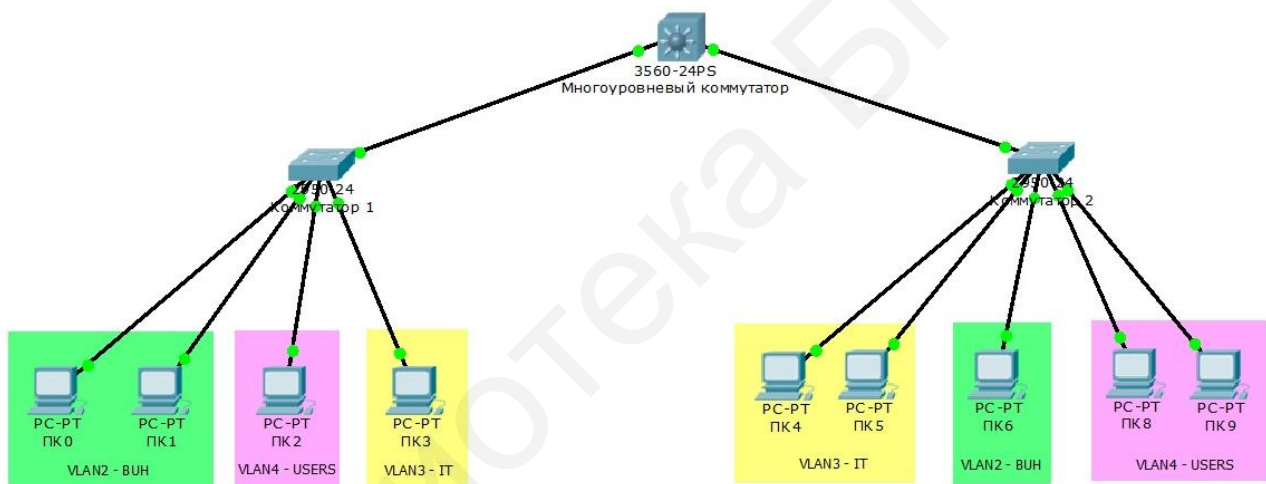


Рис. 5.3. Топология сети с использованием коммутаторов второго и третьего уровней и транковыми соединениями

В данном случае процесс настройки сети имеет следующий алгоритм. Вначале производится настройка коммутаторов второго уровня: создаются необходимые VLAN и назначаются им соответствующие физические интерфейсы.

На коммутаторах второго уровня в данном случае не нужно задавать IP-адреса интерфейсам. Интерфейс, которым коммутатор второго уровня соединяется с коммутатором третьего уровня, назначается в качестве транкового соединения и указывается, какие VLAN он обслуживает.

Далее настраиваются параметры коммутатора третьего уровня. Интерфейсы коммутатора, на которые приходят данные с коммутаторов второго уровня, переводятся также в режим транкового соединения. Следующим за ним действием будет настройка типа инкапсуляции входящих кадров. Это

производится командой «*switchport trunk encapsulation <тип инкапсуляции>*». В качестве используемого типа может быть один из стандартов организации виртуальных локальных сетей, например IEEE 802.1q. Для этого в качестве параметра команды *<тип инкапсуляции>* следует прописать «*dot1q*».

Следующий этап настройки – назначение виртуальных локальных сетей портам командой «*switchport trunk allowed vlan <номера VLAN через запятую>*».

Далее производится назначение IP-адресов созданным виртуальным интерфейсам в соответствии с принадлежностью каждой из VLAN.

Последним шагом настройки является включение автоматической маршрутизации посредством команды «*ip routing*» и сохранение конфигурации оборудования.

Если сеть строится на основе классического подхода с использованием коммутаторов второго уровня и маршрутизатора (рис. 5.4), то алгоритм процесса конфигурации сети будет примерно следующим.

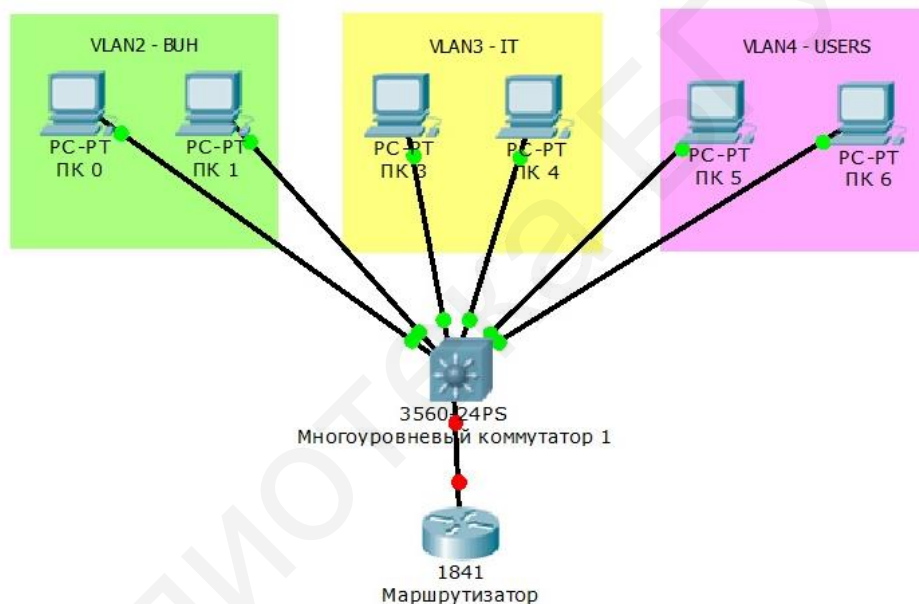


Рис. 5.4. Топология сети с коммутатором второго уровня и маршрутизатором

После соединения всех устройств в сети производится первоначальная настройка коммутатора второго уровня, которая включает создание виртуальных локальных сетей, определение соответствующих физических интерфейсов. Далее настраиваются транковые соединения между коммутатором и маршрутизатором и конфигурация коммутатора сохраняется.

Процедура настройки маршрутизатора производится аналогично настройке коммутатора в режиме глобального конфигурирования.

Вначале включается физический порт (по умолчанию все порты маршрутизатора отключены) посредством команды «*int <название физического интерфейса> no shutdown*».

Затем необходимо на этом порту произвести настройку подынтерфейсов, поскольку по транковому соединению на маршрутизатор приходят пакеты от нескольких виртуальных сетей. Каждому подынтерфейсу будет соответствовать определенная виртуальная локальная сеть. Создание подынтерфейсов производится командой «*interface <название физического интерфейса> .номер подынтерфейса*». В качестве номера подынтерфейса удобно использовать цифры, соответствующие номерам виртуальных локальных сетей, например 2.

Далее указывается тип инкапсуляции – «*encapsulation dot1q <номер VLAN>*» и задается IP-адрес – «*ip address <IP-адрес, маска>*». Включение данного подынтерфейса производится командой «*no shutdown*».

Данную процедуру необходимо провести для каждой виртуальной локальной сети (создать и настроить соответствующее количество подынтерфейсов) и сохранить конфигурацию.

Когда сеть имеет структуру, состоящую из коммутаторов второго, третьего уровня, маршрутизатора и других устройств (рис. 5.5), то в такой сети маршрутизация трафика должна быть налажена таким образом, чтобы внутрисетевое взаимодействие выполнялось на коммутаторе третьего уровня, а межсетевое с внешними сетями – маршрутизатором.

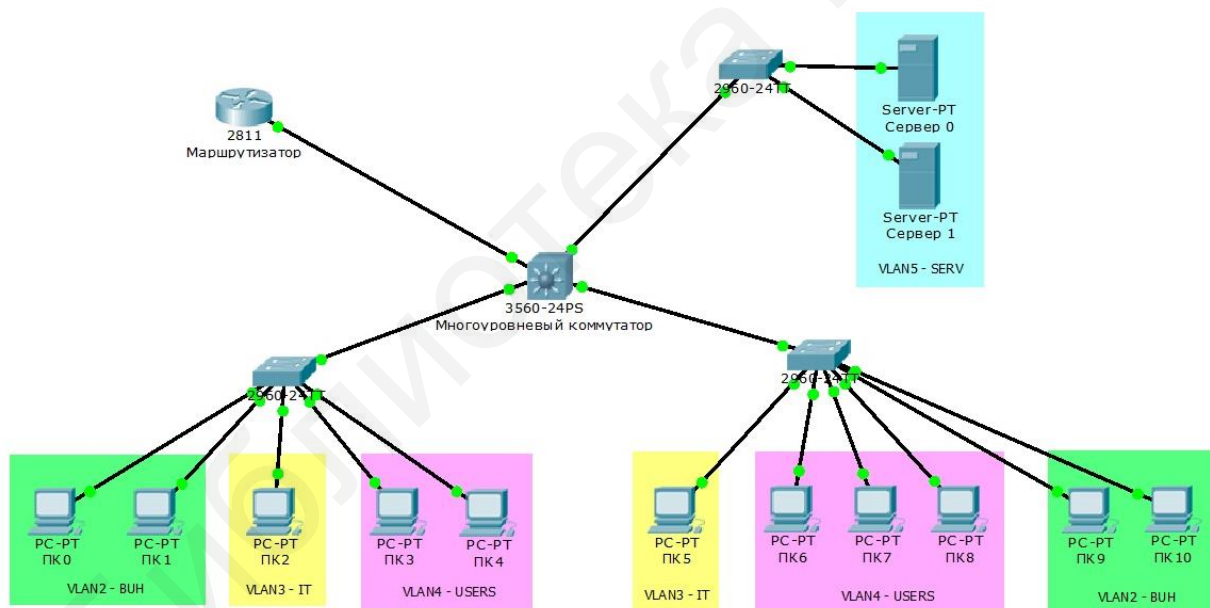


Рис. 5.5. Пример топологии локальной сети с коммутаторами второго и третьего уровней и маршрутизатором

Коммутаторы второго уровня, входящие в состав сети, настраиваются аналогично предыдущему варианту сети. На коммутаторе третьего уровня физические интерфейсы, которые связывают его с коммутаторами второго уровня, переводятся в режим транкового соединения с указанием виртуальных сетей, приходящих на них, и типом инкапсуляции получаемых кадров информации.

Также на коммутаторе третьего уровня выполняется настройка виртуальных интерфейсов с указанием IP-адресов и масок и включается автоматическая маршрутизация трафика посредством команды «*ip routing*».

На следующем этапе производится настройка соединения между коммутатором третьего уровня и маршрутизатором. Для такой схемы сети маршрутизация трафика внутри сети производится коммутатором третьего уровня, а маршрутизатор выполняет передачу пакетов только во внешние сети. Поэтому данный сегмент сети можно считать отдельной виртуальной локальной сетью со своим номером.

Таким образом, на коммутаторе третьего уровня нужно создать отдельную виртуальную сеть, например VLAN6, и соответствующий ей виртуальный интерфейс. Физический порт, который соединяет эту сеть с маршрутизатором, настраивается как порт доступа (access port).

На маршрутизаторе производится настройка физического интерфейса и присваивается IP-адрес. Необходимости создавать подынтерфейсы в данном случае нет, так как на данный порт приходит трафик только одной VLAN.

После настройки оборудования должна выполняться маршрутизация трафика внутри локальной сети коммутатором третьего уровня, а весь исходящий трафик должен маршрутизироваться маршрутизатором.

5.2. Задание для выполнения лабораторной работы

Основываясь на полученных знаниях по настройке коммутаторов второго уровня, требуется произвести настройку локальных сетей, работающих с использованием оборудования третьего уровня модели OSI в различных конфигурациях.

Используя программный пакет Cisco Packet Tracer, требуется реализовать топологию сетей, показанных на рис. 5.2–5.5, и исследовать их работу. Количество хостов для построения виртуальных локальных сетей задается преподавателем (минимальное число хостов – 2).

Для локальных сетей, показанных на рис. 5.4 и 5.5, использовать следующие настройки IP-адресов виртуальных сетей:

– для рис. 5.4:

VLAN 2 – 192.168.2.0 255 255 255.0;

VLAN 3 – 192.168.3.0 255.255.255.0;

VLAN 4 – 192.168.4.0 255.255.255.0;

– для рис. 5.5:

VLAN 2 – 192.168.22.0 255.255.255.0;

VLAN 3 – 192.168.33.0 255.255.255.0;

VLAN 4 – 192.168.44.0 255.255.255.0;

VLAN 5 – 192.168.55.0 255.255.255.0.

В дальнейшем топология этих сетей будет использована в лабораторной работе №6.

1. Исследовать локальную сеть с несколькими виртуальными сетями с использованием прямых подключений к коммутатору третьего уровня (см. рис. 5.2).

Для этого необходимо произвести расчет статических IP-адресов, масок сетей и подсетей (порядок расчета и настройки описан в лабораторной работе №2) в зависимости от числа заданных преподавателем хостов. Далее настроить коммутатор для работы с виртуальными сетями (порядок настройки и работы с VLAN описан в лабораторной работе №4).

Проверить прохождение пакетов данных между различными хостами внутри одной сети и между разными сетями. На данном этапе пакеты данных должны передаваться только между хостами одной сети.

Настроить маршрутизацию на коммутаторе третьего уровня командой *ip routing*. Проверить прохождение пакетов данных между различными хостами разных сетей.

2. Собрать сеть, имеющую в своем составе несколько коммутаторов второго уровня, соединенных с использованием агрегированного канала (см. рис. 5.3). Произвести настройку IP-адресов и виртуальных сетей с учетом того, что не нужно задавать IP-адреса интерфейсам VLAN.

Соединить коммутаторы второго уровня и коммутатор третьего уровня и настроить интерфейсы как транковые. Настроить тип инкапсуляции входящих кадров.

Далее следует назначить виртуальные локальные сети портам посредством команды *switchport trunk allowed vlan* и произвести назначение IP-адресов созданным виртуальным интерфейсам в соответствии с принадлежностью каждой из VLAN.

Включить автоматическую маршрутизацию с помощью команды *ip routing* и сохранить конфигурацию оборудования.

Проверить прохождение пакетов данных между различными хостами сети.

3. Создать сеть на основе коммутатора второго уровня и маршрутизатора (см. рис. 5.4). Произвести настройку сети с учетом требований, указанных в начале подразд. 5.2, и алгоритма действий, изложенного в п. 5.1.2.

Для этого настроить IP-адреса хостов, настроить VLAN на коммутаторах второго уровня. Далее следует настроить транковые соединения между коммутаторами и маршрутизатором и настроить маршрутизацию данных между сетями. При настройке маршрутизатора необходимо учитывать, что процедура создания и настройки соответствующего количества интерфейсов и подынтерфейсов производится для каждой VLAN.

После настройки маршрутизатора необходимо проверить прохождение пакетов данных между различными сетями.

4. Собрать сеть, показанную на рис. 5.5. Настроить данную сеть в соответствии с предложенными в начале подразд. 5.2 настройками и порядком действий, описанным в п. 5.1.2.

При настройке сети следует учитывать, что внутрисетевое взаимодействие выполняется коммутатором третьего уровня, а межсетевое – маршрутизатором.

После настройки оборудования проверить прохождение пакетов внутри сети и между сетями. Убедиться в правильности настройки маршрутизации внутрисетевого и межсетевого трафика.

5.3. Содержание отчета

1. Титульный лист.
2. Цель работы.
3. Схемы реализованных топологий сетей.
4. Конфигурация коммутаторов и маршрутизаторов (списки настроенных виртуальных локальных сетей, интерфейсов, таблица MAC-адресов).
5. Результаты прохождения пакетов данных между хостами одной VLAN и между хостами, принадлежащими различным VLAN.
6. Выводы.

5.4. Контрольные вопросы

1. Что такое маршрутизация?
2. На каком уровне модели взаимодействия открытых систем осуществляется маршрутизация трафика?
3. Какое оборудование может применяться в целях маршрутизации трафика?
4. В чем состоит отличие коммутатора третьего уровня от коммутатора второго уровня?
5. Чем различаются коммутатор третьего уровня и маршрутизатор?
6. Что такое шлюз по умолчанию и для чего он используется?
7. Поясните процесс настройки маршрутизации между различными виртуальными сетями при использовании коммутатора третьего уровня.
8. Произведите сравнение коммутатора третьего уровня и маршрутизатора.

Лабораторная работа №6 СТАТИЧЕСКАЯ МАРШРУТИЗАЦИЯ

Цель работы:

- ознакомиться с основами маршрутизации трафика в локальных вычислительных сетях;
- ознакомиться с основными используемыми протоколами маршрутизации;
- научиться производить конфигурирование сетевого оборудования, предназначенного для маршрутизации трафика, с использованием статической маршрутизации.

6.1. Краткие теоретические сведения

6.1.1. Общие сведения о маршрутизации. Основные типы маршрутизации

Как было отмечено в предыдущей работе, *маршрутизация сетевого трафика* – это процесс определения маршрута передачи данных в вычислительных сетях.

Основными целями маршрутизации являются:

- обеспечение минимального времени задержки передачи пакета информации от отправителя к получателю в сети;
- обеспечение минимальной вариации времени задержки;
- обеспечение максимальной пропускной способности сети;
- обеспечение надежности доставки пакета адресату;
- обеспечение максимальной защиты пакета от различных угроз безопасности.

Объединение нескольких локальных вычислительных сетей в глобальную распределенную сеть производится на третьем уровне семиуровневой модели взаимодействия открытых систем OSI при помощи специализированного оборудования и протоколов, позволяющих производить перенаправление трафика. В качестве оборудования, позволяющего производить маршрутизацию трафика в сети, могут быть использованы маршрутизаторы, модемы, коммуникационные серверы.

Для решения задачи маршрутизации пакета передаваемых данных маршрутизатор должен владеть определенным набором информации:

- об адресе назначения перенаправляемого пакета;
- об адресах соседних маршрутизаторов;
- о пути доступа ко всем удаленным сетям;
- о наилучшем пути к каждой удаленной сети;
- о методах обслуживания и проверки информации о маршрутизации.

Информацию об удаленных сетях маршрутизатор может получить от соседних маршрутизаторов или же от системного администратора. На основе данной информации маршрутизатор составляет таблицу маршрутизации, которая описывает, как найти другие сети.

Если локальная вычислительная сеть подключена напрямую к маршрутизатору, то он знает пути доставки пакетов информации из других сетей. Если же сеть не имеет прямого доступа, то маршрутизатор должен определить пути доступа к удаленной сети. Это можно выполнить либо при помощи задания вручную путей доставки пакетов в другие сети (статическая маршрутизация) или же с использованием автоматического определения маршрута доставки пакетов при помощи алгоритмов динамической маршрутизации.

Процесс маршрутизации можно разделить на два взаимосвязанных уровня:

- уровень маршрутизации;
- уровень передачи пакетов.

На первом верхнем уровне производится работа с таблицей маршрутизации. Она служит для определения сетевого адреса следующего маршрутизатора или непосредственного получателя пакета.

На втором уровне осуществляется работа с пакетами передаваемой информации. Здесь выполняются процедуры проверки контрольной суммы, определения адреса назначения на канальном уровне, производится отправка пакета с учетом очередности, фрагментации и фильтрации пакетов. Данные действия выполняются на основе информации, получаемой с уровня маршрутизации.

Процесс определения маршрута следования пакета производится на уровне маршрутизации. Как правило, определение маршрута передачи данных производится программными методами, основанными на алгоритмах маршрутизации. Совокупность методов определения маршрута носит название протоколов маршрутизации.

Различают следующие способы маршрутизации:

- централизованный;
- распределенный;
- смешанный.

При централизованном способе маршрутизации выбор маршрута для передачи осуществляется в центре управления сетью. В этом случае узлы в сети только принимают и реализуют результаты решения задачи маршрутизации. Недостатками данной маршрутизации являются уязвимость к отказам центрального узла и низкая гибкость управления.

При распределенном (децентрализованном) способе маршрутизация в сети выполняется узлами сети, имеющими соответствующие средства для осуществления функций маршрутизации. В отличие от централизованной маршрутизации распределенная маршрутизация более сложная, но в то же время обладает большей гибкостью.

Смешанная маршрутизация представляет собой смесь в определенном соотношении принципов централизованной и распределенной маршрутизации.

Для выбора оптимального маршрута каждый узел в сети передачи данных должен иметь информацию о состоянии сети в целом. Задача маршрутизации решается при выполнении условия, что кратчайший маршрут, обеспечивающий передачу пакета за минимальное время, зависит от набора параметров: топологии сети, пропускной способности каналов связи, нагрузки на линии связи и др.

Основными требованиями, предъявляемыми к алгоритмам маршрутизации, могут быть:

- оптимальность выбора маршрута;
- простота реализации алгоритма;
- устойчивость;
- быстрая сходимость;
- гибкость реализации.

Под оптимальностью понимают способность алгоритма маршрутизации выбирать наилучший маршрут с точки зрения определенных параметров, например: количество переходов, минимальное время задержки и др.

Простота реализации алгоритма подразумевает эффективность выполнения своих функциональных возможностей с минимальными затратами.

Устойчивость алгоритмов маршрутизации показывает, насколько способны функционировать эти алгоритмы в условиях непредвиденных ситуаций, например: отказов оборудования, высокой нагрузки на каналы связи и др.

Под сходимостью алгоритмов маршрутизации понимают процесс согласования между маршрутизаторами информации о топологии сети. Если определенное событие в сети приводит к тому, что некоторые маршруты становятся недоступны или возникают новые маршруты, маршрутизаторы рассылают сообщения об этом друг другу по всей сети. После получения этих сообщений маршрутизаторы производят переназначение оптимальных маршрутов, что в свою очередь может породить новый поток сообщений. Этот процесс должен завершиться, причем достаточно быстро, иначе в сетевой топологии могут появиться петли, или сеть вообще может перестать функционировать. Алгоритмы маршрутизации должны быстро и правильно учитывать изменения в состоянии сети (например, отказ узла или сегмента сети).

Гибкость реализации алгоритмов маршрутизации подразумевает адаптацию алгоритмов в процессе работы сети при наличии различного рода изменений.

Все алгоритмы маршрутизации можно классифицировать по ряду признаков. Например, выделяют следующие типы алгоритмов:

- статические и динамические;
- одномаршрутные и многомаршрутные;
- одноуровневые и многоуровневые;
- внутридоменные и междоменные;
- алгоритмы состояния канала связи и алгоритмы вектора расстояний.

Статические (неадаптивные) алгоритмы используют маршруты, выбранные заранее и занесенные вручную в таблицу маршрутизации сетевым администратором. Протоколы, которые разработаны на этом типе алгоритмов,

называются немаршрутизируемыми. Примерами этих протоколов могут быть протоколы LAT, NetBIOS.

Динамические алгоритмы базируются на автоматическом обновлении таблицы маршрутизации при изменении топологии сети или изменении трафика в ней. Динамические алгоритмы маршрутизации могут также дополнять статические маршруты.

Одномаршрутные алгоритмы определяют только один маршрут доставки информационного пакета. Данный маршрут не всегда может быть оптимальным. Многомаршрутные алгоритмы могут иметь несколько маршрутов доставки пакета получателю и позволяют передавать информацию получателю по нескольким каналам одновременно. Это означает, что пропускная способность и надежность будет выше.

Сети передачи данных могут иметь одноуровневую (одноранговую) или иерархическую архитектуру. В сетях с одноуровневой организацией все сегменты сети имеют одинаковый приоритет. В иерархической сети выделяются фрагменты сети – подсети. В этом случае маршрутизаторы нижнего уровня обеспечивают связь фрагментов сети между собой, а маршрутизаторы верхнего уровня образуют магистраль – опорную часть сети, через которую передаются пакеты между сетями нижнего уровня.

Иерархическая структура в больших и сложных сетях позволяет значительно упростить процесс управления сетью, облегчает изоляцию сегментов сети и т. д.

Основным преимуществом иерархической маршрутизации является то, что она имитирует организацию большинства компаний и, следовательно, очень хорошо поддерживает их схемы трафика. Большая часть сетевой связи имеет место в пределах групп небольших сетей (доменов). Внутридоменным маршрутизаторам необходимо знать только о других маршрутизаторах в пределах своего домена, поэтому их алгоритмы маршрутизации могут быть упрощенными. Соответственно может быть уменьшен и трафик обновления маршрутизации, зависящий от используемого алгоритма маршрутизации. Междоменные протоколы обеспечивают маршрутизацию пакетов между различными доменами.

Алгоритмы состояния канала связи функционируют на основе информации о каналах связи, передаваемой каждым маршрутизатором в сеть. Маршрутизатор сообщает другим только ту часть маршрутной таблицы, которая касается только тех каналов связи, которые он использует.

Алгоритмы вектора расстояния (либо же дистанционно-векторные протоколы) требуют обмена всей или частью своей таблицы маршрутизации со своими соседями.

То есть алгоритмы состояния каналов связи производят обмен небольшими корректировками маршрутной информации между всеми участниками сети, а алгоритмы вектора расстояния отправляют более крупные корректировки только в соседние маршрутизаторы.

Эффективность выбора алгоритмов маршрутизации можно оценить следующими показателями:

- длиной маршрута;
- надежностью;
- временем доставки пакета адресату;
- нагрузкой на сеть передачи данных.

Длина маршрута может определяться количеством пересылок пакета от источника до адресата через сеть или задаваться администратором сети в зависимости от стоимости расходов, связанной с каждым задействованным каналом связи.

Надежность зависит от типа используемого канала связи и характеризуется числом битовых ошибок, возникающих в процессе передачи пакета информации через определенный канал связи.

Время доставки пакета адресату зависит от многих факторов, например таких, как скорость передачи через канал связи (полосы пропускания канала связи), задержка в промежуточных маршрутизаторах при обработке поступающих пакетов, загрузка сети, физическое расстояние, на которое надо передать пакет.

Нагрузка на сеть передачи данных определяется длиной очередей пакетов в узлах сети.

Кроме этих факторов имеются и другие, которые используются алгоритмами при расчете оптимального пути доставки пакета.

Совокупность параметров, на основе которых производится выбор маршрута доставки пакета, называется метрикой маршрута.

Различают несколько подходов к выбору маршрута:

- одношаговый подход;
- маршрутизация от источника информации.

При маршрутизации от источника информации выбор маршрута производится конечным узлом или же первым маршрутизатором на пути следования пакета. Остальные маршрутизаторы только занимаются пересылкой пакета по заданному маршруту. Данный подход используют в целях отладки.

Наиболее распространенным подходом при выборе маршрута является одношаговый. Он заключается том, что каждый конечный узел и каждый маршрутизатор выбирают маршрут передачи пакета информации только на один шаг – IP-адрес следующего маршрутизатора (или конечного узла). Этому маршрутизатору передается ответственность за выбор следующего участка маршрута доставки пакета адресату. Такой подход распределяет задачу выбора маршрута и снимает ограничение на максимальное число промежуточных маршрутизаторов в пути доставки пакета.

При одношаговом подходе различают следующие группы алгоритмов построения таблиц маршрутизации:

- алгоритмы фиксированной маршрутизации;
- алгоритмы простой маршрутизации;
- алгоритмы адаптивной маршрутизации.

Разница между данными группами алгоритмов заключается в степени учета изменения топологии и нагрузки сети при решении задачи вычисления маршрута передачи данных.

Первая группа алгоритмов основана на составлении таблиц маршрутизации непосредственно администратором сети и применяется в сетях с простой топологией. При выборе маршрута при таком способе маршрутизации производится учет только изменения топологии сети без учета степени загруженности. Маршрут выбирается исходя из расстояния, т. е. по кратчайшему пути. Изменение нагрузки в сети может приводить к задержке передачи пакетов в сети. Таблицы маршрутизации составляются каждый раз при изменении топологии сети.

Алгоритмы простой маршрутизации при выборе маршрута не учитывают топологию сети и ее состояние (загруженность). Данный тип маршрутизации не обеспечивает направленной передачи пакетов и имеет низкую эффективность, однако обладает простотой реализации алгоритмов и обеспечивает устойчивую работу сети при выходе из строя отдельных элементов. Выделяют следующие типы простой маршрутизации: случайная, лавинная и адаптивная.

При случайной маршрутизации для передачи пакета из узла выбирается одно случайное направление для дальнейшей передачи, кроме исходного. Пакет в этом случае «блуждает» по сети и с какой-то конечной вероятностью когда-нибудь достигает адресата.

При лавинной маршрутизации производится передача пакета из узла по всем свободным выходным линиям. При этом данный процесс происходит в каждом узле, что приводит к появлению дублирующих пакетов в сети, что сильно отражается на пропускной способности сети, т. е. возникает так называемый «лавинный эффект». Однако этот способ маршрутизации дает гарантированное обеспечение оптимального времени доставки информационного сообщения адресату, поскольку одно из возможных направлений доставки сообщения может обеспечить минимальное время. Данный метод можно использовать в незагруженных сетях.

Простой адаптивный алгоритм маршрутизации использует при построении таблицы маршрутизации данные, содержащиеся в проходящих через данный маршрутизатор пакетах.

Адаптивные алгоритмы маршрутизации используют информацию об изменении топологии сети и ее загруженности при принятии решения о выборе маршрута доставки пакета. Существуют локальные, распределенные, централизованные и гибридные алгоритмы маршрутизации.

При использовании алгоритмов локальной адаптивной маршрутизации производится анализ информации в данном узле о текущем состоянии таблицы маршрутизации, состоянии каналов связи, подключенных к узлу, количестве пакетов в очереди на передачу. На основе этого анализа производится построение маршрута доставки пакета за минимальное время. Достоинством данного способа адаптации является выбор маршрута по самым последним данным о состоянии узла. Однако этот способ не учитывает состояния всей сети, что может

приводить к передаче пакета по перегруженным каналам и, как следствие, к задержке в доставке пакетов адресату.

Алгоритмы распределенной адаптивной маршрутизации учитывают текущее состояние узла и информацию, получаемую от соседних узлов. В этом случае производится формирование таблицы маршрутизации, основанной на определении маршрутов по минимальному времени задержки. В процессе работы устройства в сети (маршрутизаторы) обмениваются информацией с соседними узлами о состоянии нагрузки узла, т. е. о длине очереди пакетов узла. После получения такой информации каждый маршрутизатор производит перерасчет задержек и корректирует маршрут передачи пакетов на основе полученных данных. Обмен информацией может производиться как периодически, так и асинхронно, в случаях резких изменений топологии сети или нагрузки. Информация о состоянии соседних узлов повышает эффективность формирования маршрутов передачи данных, однако при этом происходит рост служебного трафика в сети. Сведения об изменении состояния сети проходят относительно медленно, поэтому выбор маршрута передачи данных производится в этом случае на основе устаревших данных.

Алгоритмы централизованной адаптивной маршрутизации основаны на решении задачи маршрутизации для каждого узла в сети передачи данных вычислительного центра сети передачи данных на основе информации, получаемой от узлов с определенной периодичностью. По этой информации производится расчет таблиц маршрутов и отправка их обратно в маршрутизаторы. Все эти действия происходят с определенными задержками, поэтому данные алгоритмы имеют низкую эффективность, особенно при высокой загрузке сети передачи данных. Кроме этого, имеется определенная доля вероятности потери работоспособности сети при выходе из строя управляющего центра сети.

Гибридные адаптивные алгоритмы маршрутизации сочетают в себе черты алгоритмов централизованной и локальной адаптивной маршрутизации. В этом случае используются таблицы маршрутизации, построенные управляющим центром сети для каждого узла, и анализируется информация о текущем состоянии узла. При этом подходе устраняются недостатки централизованной маршрутизации (устаревшая информация о состоянии сети) и локальной маршрутизации (известна информация только о текущем состоянии узла) и используются преимущества обоих способов: маршруты, построенные центром управления сетью, показывают общее состояние сети, а текущее состояние узла решает задачу определения минимального времени доставки пакета получателю.

6.1.2. Протоколы маршрутизации

Под протоколом понимается набор соглашений между устройствами, посредством которых производится обмен информацией между ними в системе передачи данных в рамках одного уровня модели взаимодействия открытых систем.

Протоколом маршрутизации называется сетевой протокол, посредством которого производится управление потоками передаваемой информации в сети. Маршрутизаторы используют протоколы маршрутизации для обмена информацией о состоянии сети.

Как было уже отмечено, все протоколы можно разделить на несколько категорий по ряду признаков.

Исходя из принципов работы протоколов маршрутизации по отношению к автономным системам, можно выделить протоколы внутренних шлюзов (IGP – Interior Gateway Protocol) и протоколы внешних шлюзов (EGP – Exterior Gateway Protocol). Их иначе называют протоколами внутридоменной и междоменной маршрутизации.

Под автономной системой в этом случае понимается сеть или группа сетей, находящихся под единым административным управлением и контролем.

К протоколам внутридоменной маршрутизации, или IGP, относят следующие протоколы: RIP (Routing Information Protocol), RIP v2; IGRP (Interior Gateway Routing Protocol); EIRGP (Enhanced Interior Gateway Routing Protocol); OSPF (Open Shortest Path First); IS-IS (Intermediate System – to – Intermediate System).

Наиболее широко известным протоколом междоменной маршрутизации (EGP) является протокол маршрутизации данных между автономными системами BGP (Border Gateway Protocol).

В зависимости от типа алгоритма маршрутизации выделяют протоколы, основанные на алгоритмах контроля состояния каналов связи, и протоколы, основанные на алгоритмах вычисления вектора расстояний (дистанционно-векторные протоколы).

В случае классификации по типу взаимодействия между автономными системами принадлежность маршрутизаторов к BGP- и EGP-протоколам описывает их физическое взаимодействие, следовательно, классификация протоколов по типу используемых алгоритмов описывает взаимодействие маршрутизаторов между собой при обмене информацией о маршрутах.

Алгоритмы вычисления расстояний или же дистанционно-векторные алгоритмы определяют направление (вектор) и расстояние (количество промежуточных узлов) для каждого из каналов связи. В этом случае маршрутизатор периодически пересылает соседним маршрутизаторам всю или часть своей таблицы маршрутизации. Периодическая рассылка обновлений производится даже в том случае, если в системе не произошло изменений. Получив информацию от своего соседа, маршрутизатор может внести в свою таблицу маршрутизации изменения на основе новых данных. Данная группа алгоритмов является основой следующих протоколов маршрутизации: RIP (Routing Information Protocol), RIP v2; IGRP (Interior Gateway Routing Protocol); EIRGP (Enhanced Interior Gateway Routing Protocol); BGP (Border Gateway Protocol).

Протоколы, использующие алгоритмы контроля состояния канала связи, разработаны с учетом недостатков, имеющих у протоколов, основанных на дистанционно-векторных алгоритмах. Такие протоколы позволяют оперативно реагировать на изменение состояния сети путем рассылки периодических

обновлений только в случаях изменения состояния сети, а также выполнять рассылку периодических обновлений через большие промежутки времени.

Когда происходит изменение состояния одного или нескольких каналов связи, устройство, обнаружившее данное изменение, формирует извещение об изменении состояния канала и рассылает его всем соседним маршрутизаторам. Маршрутизаторы выполняют обновление своей таблицы маршрутизации и отправляют извещение своим соседям и т. д. Массовая рассылка извещения требуется для гарантии того, что все маршрутизаторы в сети передачи данных обновят свои таблицы маршрутизации.

Примерами протоколов, основанных на алгоритмах контроля состояния канала связи, являются протоколы OSPF (Open Shortest Path First), IS-IS (Intermediate System – to – Intermediate System).

Рассмотрим некоторые из приведенных протоколов.

Протокол маршрутной информации RIP (Routing Information Protocol) использует для определения направления и расстояния до любого из каналов сети счетчик количества промежуточных узлов. Если существует несколько маршрутов доставки пакетов к адресату, то выбирается тот из них, который содержит наименьшее число промежуточных узлов. Ввиду того что в качестве метрики маршрута протокол RIP использует только счетчик промежуточных узлов, не всегда выбранный маршрут может быть кратчайшим.

Протокол RIP версии 1 работает только в сетях с классовой маршрутизацией, т. е. все устройства в таких сетях должны иметь одинаковую маску сети, поскольку протокол не включает в маршрутные обновления информацию о ней. Этот недостаток устранен в протоколе RIP версии 2. В данном варианте протокола используется префиксная маршрутизация и пересылка маски сети вместе с анонсами таблиц маршрутизации. Благодаря этому обеспечивается работа с подсетями с разной длины масками внутри одной и той же сети.

Протокол маршрутизации внутреннего шлюза IGRP (Interior Gateway Routing Protocol), разработанный корпорацией Cisco, использует дистанционно-векторный алгоритм и предназначен для решения проблем, возникающих при маршрутизации в больших сетях, где невозможно использовать такие протоколы, как RIP. Протокол IGRP способен выбирать самый быстрый путь на основе задержки, пропускной способности, загрузки и надежности канала. Протокол IGRP использует в качестве метрики только пропускную способность и задержку. Этот протокол имеет значительно большее максимальное значение счетчика узлов, чем протокол RIP, что дает возможность использовать его в более крупных сетях. Протокол IGRP работает в сетях с классовой маршрутизацией.

Расширенный протокол маршрутизации внутреннего шлюза EIGRP (Enhanced Interior Gateway Routing Protocol) был также разработан корпорацией Cisco и представляет собой усовершенствованную версию протокола IGRP. Он использует некоторые функции алгоритмов с контролем состояния канала, что относит его к гибридным протоколам.

Открытый протокол поиска кратчайшего пути OSPF (Open Shortest Path First) использует алгоритм маршрутизации с контролем состояния каналов.

OSPF является протоколом IGP-типа, что означает, что он распространяет маршрутную информацию между маршрутизаторами, находящимися в единой автономной системе. Протокол OSPF был разработан для использования в больших сетях, в которых невозможно использование протокола RIP.

Протокол обмена маршрутной информацией между промежуточными системами IS-IS (Intermediate System – to – Intermediate System) основан на использовании алгоритмов маршрутизации с контролем состояния канала связи для стека протоколов модели OSI. Он распространяет маршрутную информацию для протокола сетевого обслуживания CLNP (Connectionless Network Protocol), для соответствующих ISO-служб сетевого обслуживания без установления соединения CLNS (Connectionless Network Service). Интегрированный протокол IS-IS является вариантом реализации протокола IS-IS для маршрутизации нескольких сетевых протоколов. Интегрированный протокол IS-IS объединяет CLNP-маршруты с информацией об IP-сетях и масках подсетей. Благодаря соединению ISO CLNS и IP-маршрутизации в одном протоколе интегрированный протокол IS-IS предоставляет альтернативу протоколу OSPF при использовании в IP-сетях. Он может быть использован для IP-маршрутизации, ISO-маршрутизации и для комбинации этих двух вариантов.

Протокол граничного шлюза BGP (Border Gateway Protocol) является примером протокола EGP-типа. Протокол BGP обеспечивает обмен маршрутной информацией между автономными системами и гарантирует выбор маршрутов без заикливания. Он является базовым протоколом извещений маршрутизации, используемым большинством крупных компаний и поставщиками услуг доступа к сети Интернет. Протокол BGP-4 является первой версией протокола BGP, в котором встроена бесклассовая междоменная маршрутизация CIDR (Classless InterDomain Routing), и первым, использующим механизм агрегации маршрутов. В отличие от распространенных протоколов IGP-типа, таких как RIP, OSPF и EIGRP, BGP, не использует в качестве метрики счетчик узлов, пропускную способность или задержку в сети. Вместо этого протокол BGP принимает решение о выборе маршрута, руководствуясь указанными сетевыми правилами, используя различные маршрутные BGP-атрибуты. BGP-протокол – это протокол, реализующий ту или иную политику при выборе маршрута передачи данных.

6.2. Задание для выполнения лабораторной работы

Основываясь на полученных знаниях по настройке коммутаторов второго и третьего уровней, топологии сети, полученной в ходе лабораторной работы №5 (рис. 6.1) в программном пакете Cisco Packet Tracer, требуется произвести настройку межсетевой маршрутизации трафика.

В ходе выполнения лабораторной работы №5 была получена сеть передачи данных, состоящая из двух подсетей. Будем считать, что данные подсети образуют сети двух филиалов организации – Филиала 1 и Филиала 2. Данные сети

были настроены, и маршрутизация трафика в них осуществляется только в пределах каждого из филиалов.

Сеть Филиала 1 имеет три виртуальные локальные сети, маршрутизация трафика между которыми производится при помощи Маршрутизатора 0 (1841).

В Филиале 2 есть тоже три виртуальных локальных сети, предназначенных для обмена данными между пользователями и одна виртуальная локальная сеть, объединяющая серверное оборудование (Сервер 0 и Сервер 1).

Виртуальные локальные сети филиалов имеют следующие сетевые адреса и маски:

– Филиал 1:

VLAN2:	192.168.2.0	255.255.255.0;
VLAN3:	192.168.3.0	255.255.255.0;
VLAN4:	192.168.4.0	255.255.255.0;

– Филиал 2:

VLAN2:	192.168.22.0	255.255.255.0;
VLAN3:	192.168.33.0	255.255.255.0;
VLAN4:	192.168.44.0	255.255.255.0;
VLAN5:	192.168.55.0	255.255.255.0.

1. Используя топологии сетей передачи данных, полученных в лабораторной работе № 5, проверить их настройку и функционирование (прохождение пакетов между различными устройствами сети).

2. Убедиться в том, что при проверке связи от Маршрутизатора 1 Филиала 2 к хостам сети пакеты не проходят.

3. Задать статические маршруты доступа на Маршрутизаторе 1 к виртуальным сетям, настроенным на многоуровневом коммутаторе Филиала 2. Необходимо указать маршрутизатору, что компьютеры доступны только через многоуровневый коммутатор (прямого пути от маршрутизатора к компьютерам нет).

Для этого требуется зайти в режим глобального конфигурирования на Маршрутизаторе 1, выполнив команду «*configure terminal*» или «*conf t*», и, используя команду «*ip route*», указать сеть, доступ к которой требуется обеспечить. Формат команды «*ip route*» следующий:

ip route <адрес сети, в которую необходимо обеспечить доступ> <маска сети, в которую необходимо обеспечить доступ> <IP-адрес интерфейса, через который будет осуществляться доступ>.

В данном случае требуется указать IP-адрес интерфейса коммутатора третьего уровня, через который осуществляется связь маршрутизатора и коммутатора.

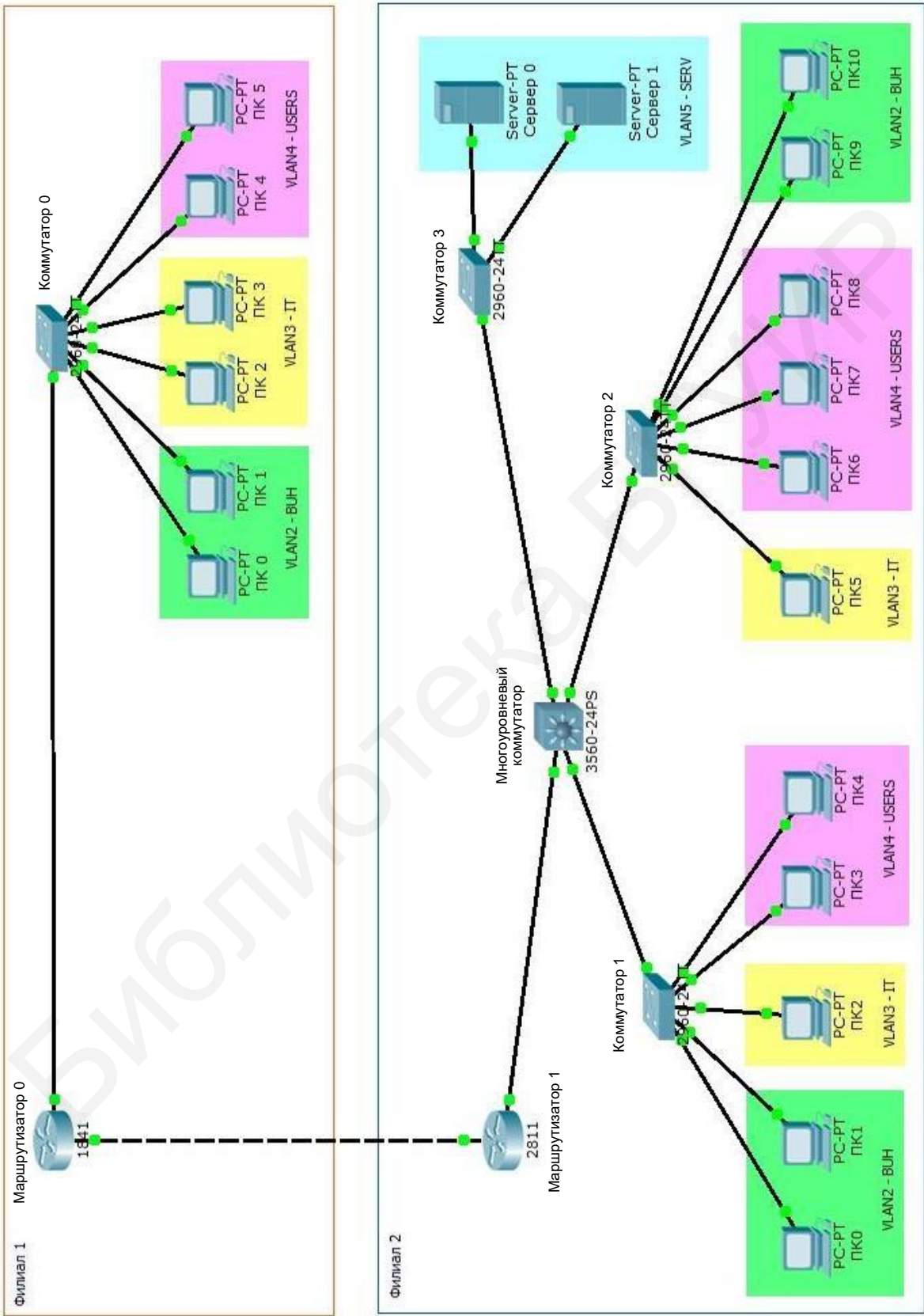


Рис 6.1 Топология сети

Данная процедура выполняется для всех сетей, доступ к которым необходим со стороны маршрутизатора.

4. Объединить две сети, принадлежащие разным филиалам, путем соединения линией связи маршрутизатора 0 (1841) Филиала 1 и Маршрутизатора 1 (2811) Филиала 2.

5. Убедиться в том, что пакеты из сети Филиала 1 не проходят в сеть Филиала 2 и наоборот.

6. Настроить интерфейс взаимодействия между двумя Маршрутизаторами. Для этого необходимо выполнить следующие действия на каждом из маршрутизаторов:

- зайти в режим глобального конфигурирования;
- включить физический интерфейс, через который осуществляется связь между маршрутизаторами;
- присвоить данному интерфейсу IP-адрес и маску сети;
- сохранить конфигурацию.

IP-адрес сети выбирается из диапазона неиспользуемых адресов, например 192.168.10.0. В данном случае организуется сеть передачи данных между двумя маршрутизаторами, поэтому необходимое число IP-адресов для сети равно двум, а маска сети может быть равна 30 бит.

7. Проверить связь между маршрутизаторами при помощи команды «*ping*». Убедиться, что обмен пакетами между ними происходит, а связь между компьютерами филиалов отсутствует.

8. Настроить маршруты прохождения пакетов из сетей Маршрутизатора 0 Филиала 1 в сети Филиала 2. Для этого можно, используя команду «*ip route*», прописать маршруты в каждую сеть второго филиала (в каждую виртуальную сеть) или же, ввиду того что в данной сети имеет место быть только одно внешнее подключение к другой сети, использовать маршрут по умолчанию.

Если использовать маршрут по умолчанию, то формат команды будет следующим:

ip route 0.0.0.0 0.0.0.0 <IP-адрес физического интерфейса, которым маршрутизатор Филиала 2 подключен к маршрутизатору Филиала 1>.

После этого можно сохранить конфигурацию и просмотреть таблицу маршрутизации с помощью команды «*show ip route*».

9. Убедиться, что пакеты, передаваемые компьютерами Филиала 1, проходят в сети Филиала 2, а в обратном направлении – нет.

Это связано с тем, что при текущей настройке многоуровневого коммутатора Филиала 2 в нем нет сведений о сетях Филиала 1. Кроме этого, Маршрутизатор 1 также не имеет сведений о сетях Филиала 1.

10. Настроить маршруты отправки всех пакетов из сетей Филиала 2 на многоуровневом коммутаторе. Для этого зайти в режим глобального конфигурирования и командой «*ip route*» прописать маршруты доставки пакетов.

В данном случае сети Филиала 2 также имеют доступ к другим сетям только по одному пути – через Маршрутизатор 1. Поэтому при настройке коммутатора можно указать только один маршрут передачи пакетов по умолчанию на интерфейс Маршрутизатора вместо нескольких.

После прописывания маршрутов сохранить конфигурацию коммутатора.

На данном этапе пакеты должны проходить из сетей Филиала 1 в сети Филиала 2. Пакеты, отправленные из сети Филиала 2, должны доходить только до маршрутизатора второго филиала.

11. Настроить маршруты доставки пакетов из сетей Филиала 2 в сети Филиала 1 на маршрутизаторе Филиала 2 (производится аналогично п. 8).

12. Проверить прохождение пакетов между сетями. После указанных действий пакеты должны передаваться между компьютерами как из сети Филиала 1 в сеть Филиала 2, так и наоборот.

6.3. Содержание отчета

1. Титульный лист.
2. Цель работы.
3. Схема реализованной топологии сети.
4. Конфигурация коммутаторов и маршрутизаторов (списки настроенных виртуальных локальных сетей, интерфейсов, таблицы маршрутизации).
5. Результаты прохождения пакетов данных между хостами в ходе выполнения маршрутизации.
6. Выводы.

6.4. Контрольные вопросы

1. Что такое маршрутизация?
2. Какие бывают виды маршрутизации?
3. Какие алгоритмы одношаговой маршрутизации существуют?
4. Что такое «метрика» маршрута?
5. Что такое протокол маршрутизации?
6. Назовите основные протоколы маршрутизации.
7. Что такое таблица маршрутизации?
8. Что такое маршрут по умолчанию?
9. Что такое сходимость маршрута?
10. Как производится настройка маршрутизации?

Лабораторная работа №7

ПРОТОКОЛ ДИНАМИЧЕСКОЙ НАСТРОЙКИ УЗЛА ДНСП И СИСТЕМА ДОМЕННЫХ ИМЕН DNS

Цель работы:

- ознакомиться с необходимостью использования в компьютерных сетях системы доменных имен DNS и протокола ДНСП;
- ознакомиться с принципами работы DNS и ДНСП;
- научиться производить настройку DNS и ДНСП в программном пакете Cisco Packet Tracer.

7.1. Краткие теоретические сведения

7.1.1. Общие сведения о системе доменных имен DNS

Числовое представление адреса хоста в системе передачи данных удобно только для восприятия со стороны вычислительной машины, но не совсем приемлемо для восприятия человеком. Запоминать наборы цифр адреса интересующего хоста в сети труднее, чем текстовые осмысленные имена.

Изначально для облегчения взаимодействия пользователей с удаленными информационными ресурсами использовались таблицы соответствия числовых адресов и символьных имен, т. е. пользователь имел у себя файл с такой таблицей и использовал ее для поиска необходимого адреса в сети. Для доступа по мнемоническому имени следовало вначале найти в таблице IP-адрес, а только лишь за тем по установленному адресу установить соединение с хостом.

Такой способ использования символьных имен удобен был до тех пор, пока в сети было мало пользователей. Основной проблемой, возникшей при росте сети, стал вопрос содержания и синхронизации согласованных списков соответствий. Решением данной проблемы стало появление системы доменных имен DNS в 1983 г.

Система доменных имен DNS (Domain Name System) – это распределенная иерархическая система, предназначенная для определения взаимного соответствия доменного имени хоста и его IP-адреса в сети, а также предоставления данных о маршрутизации электронной почты.

Под *доменным именем* понимается уникальное символьное имя хоста, служащее для его идентификации в сети в рамках определенной зоны.

Домен – это определенная зона в сети Интернет, выделенная владельцу (физическому или юридическому лицу, международной организации или стране) для обеспечения доступа к информации, размещенной ее владельцем.

Доменные имена используются для представления адресации в удобной для человека форме. Полное доменное имя (FQDN – Fully Qualified Domain Name) состоит из нескольких частей (полей) – собственно имени хоста и имен

всех доменов, входящих в иерархическую структуру DNS, разделенных точками. Максимальная длина доменного имени не должна превышать 63 символов, а общая длина имени, включая имена поддоменов, разделительные точки и имя зоны домена – до 255 символов.

Крайнее правое поле в доменном имени называется доменом верхнего уровня TLD (Top-Level Domain).

В системе доменных имен используется иерархическая структура (рис. 7.1). Данная структура напоминает древовидную структуру с узлами, формирующими «дерево». Узел верхнего уровня называют корневым доменом или доменом нулевого уровня. В системе имен сети Интернет корневой домен обозначается пустым именем, т. е. не содержащим символов.



Рис. 7.1. Иерархическая структура DNS

Выделяют следующие группы доменных имен:

- общие домены верхнего уровня gTLD (generic Top-Level Domain) (например, .com, .edu, .gov, .mil, .net, .org, .int);
- национальные домены верхнего уровня ccTLD (country code TLD) (например, .by, .ru, .de);
- интернационализованные домены IDN (Internationalized Domain Names) – доменные имена, содержащие символы национальных алфавитов;
- зарезервированные доменные имена (Reserved Top Level DNS Names) – имена, которые предназначены для использования в качестве примеров и тестирования.

Вопросами, связанными с доменными именами, IP-адресами и другими аспектами функционирования сети Интернет, занимается некоммерческая международная организация Internet Corporation for Assigned Names and Numbers (ICANN).

Управление национальными доменными зонами делегировано администраторам национальных доменных зон под общим управлением Internet Assigned Numbers Authority (IANA) – Администрацией адресного пространства Интернета.

В каждом домене имеются серверы DNS (или иначе NS-серверы – Name Server), ответственные за обслуживание базы данных имен хостов данной области сети (части распределенной базы данных имен).

По мере развития сети Интернет все домены верхнего уровня были поделены на поддомены или зоны. Каждая зона представляет собой независимый домен, но при обращении к базе данных имен запрашивает родительский домен. Родительская зона гарантирует дочерней зоне право на существование и отвечает за ее поведение в сети (точно так же, как и в реальной жизни). Каждая зона имеет по крайней мере два сервера DNS, которые поддерживают базу данных DNS для этой зоны.

Основные условия для работы серверов DNS одной зоны – наличие отдельного соединения с сетью Интернет и размещение их в различных сетях для обеспечения отказоустойчивости. Поэтому многие организации полагаются на провайдеров сети Интернет, которые ведут в их интересах вторичные и третичные серверы DNS.

В системе DNS реализуются три сценария поиска IP-адреса в базе данных:

1. Компьютер, которому необходимо получить соединение с другим компьютером в той же зоне, посылает запрос локальному DNS-серверу зоны на поиск IP-адреса удаленного компьютера. Локальный DNS-сервер, имеющий этот адрес в локальной базе данных имен, возвращает запрашиваемый IP-адрес компьютеру, который послал запрос.

2. Компьютер, которому необходимо получить соединение с компьютером в другой зоне, запрашивает локальный DNS-сервер своей зоны. Локальный DNS-сервер обнаруживает, что нужный компьютер находится в другой зоне, и формирует запрос корневому DNS-серверу. Корневой DNS-сервер спускается по дереву серверов DNS и находит соответствующий локальный DNS-сервер. От него он получает IP-адрес запрашиваемого компьютера. Затем корневой DNS-сервер передает этот адрес локальному серверу DNS, который послал запрос. Локальный DNS-сервер возвращает IP-адрес компьютеру, с которого был подан запрос. Совместно с IP-адресом передается специальное значение – время жизни TTL (Time to live). Это значение указывает локальному DNS-серверу, сколько времени он может хранить IP-адрес удаленного компьютера у себя в кэш-памяти. Благодаря этому увеличивается скорость обработки последующих запросов.

3. Компьютер, которому необходимо повторно получить соединение с компьютером в другой зоне, запрашивает локальный DNS-сервер своей зоны. Локальный DNS-сервер проверяет, нет ли этого имени в его кэш-памяти и не истекло ли еще значение TTL. Если адрес еще в кэш-памяти и значение TTL не истекло, то IP-адрес посылается запрашивающему компьютеру. Это считается неавторизованным ответом, так как локальный DNS-сервер считает, что с момента последнего запроса IP-адрес удаленного компьютера не изменился.

Во всех трех случаях компьютеру для поиска какого-либо компьютера в сети Интернет нужен лишь IP-адрес локального сервера DNS. Дальнейшую работу по поиску IP-адреса, соответствующего запрошенному имени, выполняет локальный DNS-сервер.

По мере роста дерева DNS к серверам системы доменных имен предъявлялись новые требования. В начальный период развития сети Интернет большинство запросов на поиск имен приходилось на локальные имена хостов. Основная часть DNS-трафика проходила внутри локальной зоны и лишь в худшем случае перенаправлялась на серверы DNS более высокого уровня. Однако по мере увеличения количества адресов в сети Интернет все больше DNS-запросов формировалось к удаленным хостам вне локальной зоны. Когда DNS-сервер не находил имя хоста в своей базе данных, он вынужден был запрашивать удаленный DNS-сервер. Наиболее подходящими кандидатами для удаленных DNS-серверов, естественно, стали серверы DNS верхнего уровня, которые обладают полной информацией о дереве доменов и способны найти нужный DNS-сервер, ответственный за зону, к которой принадлежит запрашиваемый хост. Затем они же возвращают IP-адрес нужного хоста локальному DNS-серверу. Все это может привести к колоссальным перегрузкам корневых серверов системы DNS.

Решением проблемы определения IP-адреса хоста, расположенного в другой доменной зоне, стали следующие идеи.

Во-первых, каждая таблица соответствия IP-адреса и доменного имени содержит информацию о том, в течение которого времен данная таблица актуальна и гарантировано не изменяется. То есть существует ее «время жизни» (около недели) и задержка на изменение. Это позволяет поместить запись в кэш-память доменных имен и не обращаться за ней к DNS-серверам вышестоящего уровня. Это сильно снижает нагрузку, поскольку в первый раз происходит взаимодействие по всей иерархии DNS, а в следующие разы взаимодействия с вышестоящими DNS-серверами не происходит, пока время задержки на изменение еще не истекло. И даже когда время актуальности истечет, то, возможно, не будет происходить все преобразование имени, а произойдет всего лишь запрос вышестоящему серверу имен с целью проверки обновления данных. И если он отвечает «нет», то записью можно пользоваться дальше.

Во-вторых, чем больше в распределенной системе узлов, тем ниже совокупная надежность самой системы. Для устранения этого недостатка также используется запоминание ответов последних запросов в кэш-памяти (кэширование) DNS-серверов. Также повышает надежность требование наличия более одного DNS-сервера. И желательно, чтобы IP-адреса у них существенно различались. При этом для удобства администратора система устраивается так, чтобы для пользователя не имело значения, какой из DNS-серверов ему ответил.

И в-третьих, производится некоторое ограничение свободы конечного пользователя. Если производится запрос к какому-либо DNS-серверу относительно преобразования IP-адреса, не имеющего никакого отношения к домену, за который отвечает этот сервер, то он имеет право ответить «не знаю». Такой запрос называется нерекурсивным или итеративным. Все DNS-сервера в сети

отвечают на нерекурсивные запросы. Другой вариант: когда происходит обращение к серверу с так называемым рекурсивным запросом, тогда DNS-сервер самостоятельно обращается к корневому и т. д. В итоге либо преобразование будет произведено, либо будет ответ, что нет такого имени, либо ответ, что время запроса превысило предел. И в отличие от нерекурсивного запроса рекурсивные запросы обычно разрешены только для хостов, расположенных в доменной зоне сервера, т. е. решение, удовлетворяет ли сервер рекурсивные запросы или нет, принимает системный администратор этого сервера. В результате получается так, что далеко не все компьютеры могут посылать далеко не всем DNS-серверам рекурсивные запросы. Как правило, существует два уровня: некая группа адресов, которым доверяют и они могут посылать рекурсивные запросы, и вторая – адреса, которые не могут.

Таким образом, конечные пользователи обычно работают только с кэширующим сервером имен, расположенным в локальной сети, который, в свою очередь, обычно передает рекурсивные запросы к кэширующему серверу имен провайдера, а тот – по аналогии выше на следующий уровень.

По выполняемым функциям выделяют следующие типы серверов DNS:

- авторитативный – DNS-сервер, отвечающий за пространство имен в определенной зоне;

- кэширующий – DNS-сервер, выполняющий обработку запросов пользователей;

- перенаправляющий – DNS-сервер, производящий передачу полученных рекурсивных запросов на вышестоящий кэширующий DNS-сервер в виде рекурсивных запросов;

- корневой DNS-сервер – сервер, отвечающий за корневую зону;

- регистрирующий – сервер, принимающий динамические обновления от пользователей, и зачастую совмещенный с DHCP-сервером;

- DSNBL-сервер (DNS blacklist или DNS blocklist) – сервер, хранящий «черные» списки хостов, с которых обычно производится рассылка спама.

Для каждого домена администратор сети ведет базу данных DNS. Эта база данных представляет собой набор простых текстовых файлов, расположенных на основном (первичном) сервере DNS (вторичные сервера периодически копируют к себе эти файлы). В файлах конфигурации сервера указывается, в каком именно файле содержатся описания каких зон, а также является сервер первичным или вторичным для этой зоны.

Элементы базы DNS часто называют ресурсными записями или RR (Resource Record). Каждая из ресурсных записей состоит из нескольких полей. Базовый формат записи выглядит так:

[NAME] [TYPE] [CLASS] [TLL] [RDLENGTH] [RDATA]

[имя] [тип] [класс] [время жизни] [длина поля данных] [данные]

Поле [NAME] – «имя» содержит информацию о доменном имени хоста, к которому принадлежит. Имя может быть относительным или абсолютным

(FQDN). Если имя относительное (не заканчивается точкой), то к нему автоматически добавляется имя текущего домена.

Поле [TYPE] – «тип» указывает код одного из типов ресурсных записей.

Поле [CLASS] – «класс», определяет класс сети. Для сети Интернет будет IN, обозначающее INternet.

Поле [TTL] – время жизни, задает интервал времени в секундах, в течение которого данные могут сохраняться в кэш-памяти сервера.

В поле [RDLENGTH] – Resource Data Length – длина записи ресурса указывается количество байт поля [RDATA] (поля данных) в формате беззнакового целого 16-битного числа.

Поле [RDATA] – поле данных. Содержание этого поля определяется типом и классом ресурсной записи.

Наиболее часто встречаемые типы ресурсных записей приведены в табл. 7.1.

Таблица 7.1

Основные типы ресурсных записей

Тип записи	Описание
A	Адрес хоста. Ставит в соответствие символьному адресу IP-адрес в формате IPv4
AAAA	Адрес хоста в формате IPv6
CNAME	Каноническая запись имени (псевдонима). Используется для перенаправления на другое имя
MX	Адрес почтового шлюза для домена
NS	Адрес узла, отвечающего за доменную зону
PTR	Соответствие адреса имени – обратное соответствие для A и AAAA
SOA	Указатель начальной записи зоны. Указывает, на каком DNS-сервере расположена эталонная информация о данном домене
SRV	Указывает местоположение серверов для сервисов
TXT	Содержит произвольные двоичные данные объемом до 255 байт

Есть также некоторые другие типы, но они намного менее распространены.

В записях можно использовать символы «#» и «;» для комментариев, «@» – для обозначения текущего домена, «()» – скобки – для написания данных на нескольких строках. Кроме того, можно использовать метасимвол «*» в имени. Порядок записей не имеет значения, за одним исключением: запись SOA должна идти первой. Дальнейшие записи считаются относящимися к той же зоне, пока не встретится новая запись SOA. Как правило, после записи зоны указывают записи DNS-серверов, а остальные записи располагают по алфавиту, но это необязательно. Формат записи SOA следующий:

```
@ IN SOA <primary-name-server> <hostmaster-email> (
    <serial-number>
    <time-to-refresh>
```

```
<time-to-retry>  
<time-to-expire>  
<minimum-TTL> )
```

Пример записи:

```
@ 86400 IN SOA test.example.com. hostmaster.example.com. (  
2009092102 ; Serial  
10600 ; Refresh  
800 ; Retry  
3400000 ; Expire  
86400 ) ; Minimum TTL
```

Здесь для DNS-сервера «test.example.com.» установлены следующие параметры.

2009092102 – поле <Serial> – серийный номер самого файла зоны. Серийный номер увеличивается каждый раз на 1 при изменении данных домена. Когда происходит проверка вторичным DNS-сервером необходимости обновления данных, он проверяет номер записи SOA на первичном сервере.

10600 – поле <Refresh> – задает время в секундах, через которое вторичный сервер будет пытаться обновить данные зоны с первичного сервера.

800 – поле <Retry> – задает время в секундах между попытками связи вторичного сервера с первичным.

3400000 – поле <Expire> – задает значение времени срока обновления в секундах. Если попытки обновления данных не привели к успеху, то по истечении срока вторичный сервер перестает отвечать на запросы для данного домена и стирает свою копию файла зоны. Если контакт установлен, значения срока и обновления сбрасываются и цикл начинается снова.

86400 – поле <Minimum TTL> – время жизни. Минимальное время, на протяжении которого запись ресурса этой зоны действительна в кэш-памяти других серверов.

Структура других ресурсных записей имеет примерно аналогичную структуру, определяемую типом записи.

Обмен данными между хостами и DNS-серверами осуществляется по принципу «запрос – ответ», т. е. хост отправляет DNS-серверу DNS-запрос, а в ответ получает интересующую его информацию. Этот обмен DNS-запросами производится согласно специально разработанному протоколу DNS.

Протокол DNS выполняет две основные функции. Он позволяет клиентским компьютерам запрашивать DNS-сервер об IP-адресе или имени какого-либо хоста в сети, а также позволяет производить обмен информацией между базами данных серверов DNS. В своей работе этот протокол использует порт 53 и хорошо известные протоколы – TCP или UDP. Пакет DNS состоит из пяти полей: заголовка, вопроса, ответа, полномочий и поля дополнительной информации. На рис. 7.2 показана общая структура пакета DNS.

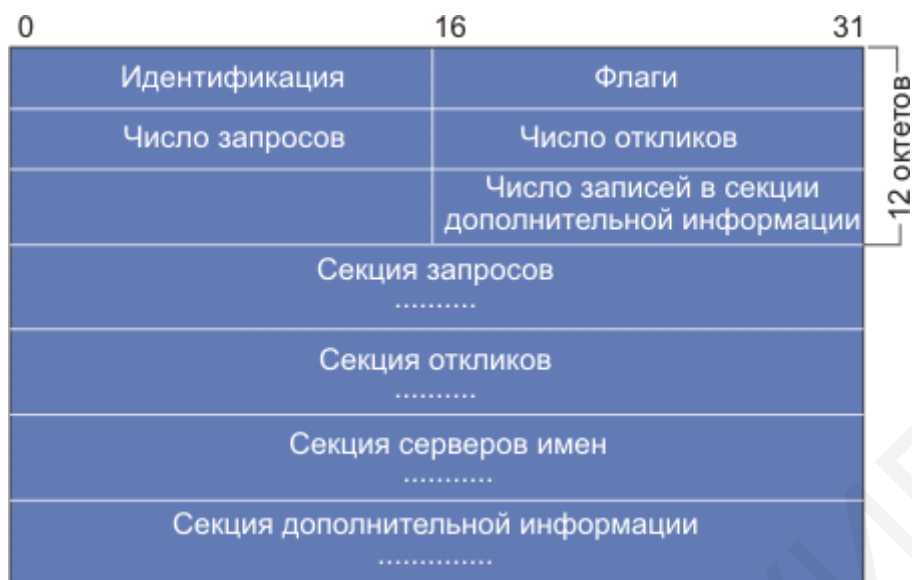


Рис. 7.2. Формат DNS-сообщений

Каждое сообщение начинается с заголовка, который содержит поле идентификации, позволяющее связать в пару запрос и отклик. Поле «флаги» определяет характер запрашиваемой процедуры, а также кодировку отклика. Поля «число записей» определяют число записей соответствующего типа, содержащихся в сообщении. Так число запросов задает число записей в секции запросов, где записаны запросы, требующие ответов. Каждый вопрос состоит из символического имени домена, за которым следует тип запроса и класс запроса.

7.1.2. Общие сведения о протоколе динамической настройки узла DHCP

DHCP (Dynamic Host Configuration Protocol) – протокол динамического конфигурирования хостов – предназначен для автоматического назначения настроек конфигурации через удаленный сервер и выступает в качестве альтернативы ручного конфигурирования IP-адресов, маски подсети, адресов сервера DNS, адресов сервера WINS и других задач адресации.

DHCP позволяет разрешать некоторые из наиболее серьезных проблем, характерных для сетей TCP/IP. Он позволяет обойтись без отдельного конфигурирования каждой рабочей станции и делает практически невозможным назначение дублированных IP-адресов.

DHCP происходит от протокола BOOTP (Bootstrap Protocol), который является «унаследованным» протоколом, разработанным для использования с бездисковыми рабочими станциями. Сервер BOOTP сохранял IP-адреса и другие настройки конфигурации для рабочих станций, образованные в соответствии с MAC-адресом, жестко закодированным в адаптере сетевого интерфейса каждой рабочей станции. При загрузке каждого компьютера сети его настройки TCP/IP доставлялись ему этим сервером. Когда стек TCP/IP начинал действовать,

BOOTP передавал исполняемый файл загрузки операционной системы на рабочую станцию с помощью протокола TFTP (Trivial File Transfer Protocol – UDP-версия FTP); после этого рабочая станция была готова к работе.

Протокол BOOTP позволил разрешить одну из основных проблем TCP/IP, устранив необходимость ручного конфигурирования каждой рабочей станции администратором или конечным пользователем. Но реально он не снял административную проблему назначения IP-адресов, поскольку он обеспечил только централизованное место для хранения настроек конфигурации. IP-настройки каждой отдельной рабочей станции по-прежнему должен был задавать администратор, сохраняя их вручную на сервере. Если в конфигурации двух различных машин случайно вводились дублированные IP-адреса, BOOTP не мог ничего сделать для обнаружения, предотвращения или исправления такой ситуации.

Протокол DHCP был разработан как расширение по сравнению с BOOTP и имеет обратную совместимость с ним. Он сохранил лучшие аспекты своего предшественника, т. е. сохранение и автоматическую доставку данных конфигурации TCP/IP, и при этом был расширен для получения лучшего решения.

DHCP может назначать IP-адреса своим клиентам, используя три различных способа:

1. Ручное выделение. Это фактически эквивалент службы BOOTP. IP-адреса и другие настройки конфигурации вводятся по отдельности администратором, сохраняются на сервере и доставляются заранее определенным клиентам.

2. Автоматическое выделение. Это подход, который называется использованием статического пула. При первой загрузке рабочей станции клиента DHCP в сети сервер DHCP назначает ему IP-адрес и другие настройки конфигурации из пула имеющихся адресов, которые были сконфигурированы администратором для использования этим сервером; они становятся постоянными настройками для данной машины. Этот метод называется отображением резервирования.

3. Динамическое выделение. Это тот же метод, что и автоматическое выделение, за исключением того, что настройки TCP/IP не назначаются как постоянные, они лишь предоставляются в аренду на заданный период времени. Эта аренда должна периодически обновляться посредством (автоматического) согласования между клиентом DHCP и сервером.

Эти три метода можно использовать одновременно, обеспечивая все возможности, которые потребуются сетевым администраторам. Ручное выделение – необходимая часть, унаследованная от BOOTP, так как часто определенным компьютерам в сети требуется определенный постоянно назначенный IP-адрес, например серверам сети Интернет и FTP-серверам. Преимуществом использования DHCP для таких компьютеров (вместо их ручного конфигурирования) является то, что всю информацию по IP-адресам для всей сети можно хранить в одном месте и DHCP не позволит любому другому клиенту DHCP использовать адреса, которые были назначены вручную.

В сети, которая изменяется редко, можно использовать DHCP для автоматического выделения IP-адресов, создавая тем самым постоянную сетевую конфигурацию. Если какой-либо компьютер перемещается из одной подсети в

другую, ему автоматически назначается новый IP-адрес для этой подсети; однако адрес, использовавшийся в старой подсети, останется занятым, пока администратор не удалит вручную эти назначения из таблицы DHCP.

Если компьютеру динамически выделяется IP-адрес, аренда этого адреса должна периодически обновляться, иначе истечет срок ее действия, что вызовет возврат данного адреса в пул свободных IP-адресов. Процесс обновления аренды выполняется автоматически и незаметен для пользователя (кроме случаев сбоя этого процесса). Если данный компьютер перемещается в другую подсеть, ему назначается подходящий IP-адрес для его новой подсети. Старый адрес возвращается в пул, когда истекает срок его аренды.

Таким образом, динамическое выделение позволило разрешить проблему «блуждающего пользователя», работающего на мобильном компьютере, с которого может выполняться вход в сеть из других офисов, других зданий или даже других городов.

Управляемое выделение IP-адресов является наиболее важной функцией DHCP, но сам по себе IP-адрес является недостаточным для полного конфигурирования стека TCP/IP. DHCP может снабжать клиента настройками для более чем 50 связанных с TCP/IP параметров. Основными параметрами являются следующие:

- IP-адрес (IP address) – 32-битный разбитый точками на 4 октета десятичный адрес, используемый для идентификации определенного хоста в сети IP;
- маска подсети – 32-битное разбитое точками на 4 октета десятичное значение, которое отделяет биты адреса сети в IP-адресе от битов адреса хоста;
- IP-адрес маршрутизатора по умолчанию, который будет использоваться клиентом для доступа к удаленным сетям (доступ к этим адресам выполняется в порядке их следования в списке);
- имена серверов DNS – IP-адреса серверов DNS, которые будут использоваться клиентом для разрешения (преобразования) имен хостов сети Интернет в IP-адреса (в порядке их следования в списке);
- имя домена данного клиента.

Протокол DHCP является клиент-серверным, то есть в его работе участвуют клиент DHCP и сервер DHCP. Передача сообщений по протоколу DHCP состоит из пакетов одного типа, которые используются для всех связей между клиентом и сервером DHCP. Передаваемый с помощью протокола UDP (User Datagram Protocol) заголовок пакета содержит поле DHCP Message Type (Тип сообщения DHCP), которое указывает назначение данного пакета среди вариантов, показанных в табл. 7.2.

Таблица 7.2

Типы сообщений DHCP

Значение	Тип сообщения	Назначение
----------	---------------	------------

1	DHCPDISCOVER	Используется клиентами для поиска DHCP-сервера
2	DHCPOFFER	Используется серверами, чтобы предлагать IP-адреса клиентам
3	DHCHREQUEST	Используется клиентами для запроса конкретного IP-адреса
4	DHCPDECLINE	Используется клиентами для отказа от предложенного IP-адреса
5	DHCPACK	Используется серверами для подтверждения согласия клиента с IP-адресом
6	DHCPNACK	Используется серверами для отклонения согласия клиента с IP-адресом
7	DHCPRELEASE	Используется клиентами, чтобы прекратить аренду IP-адреса

До согласования аренды потенциальный клиент DHCP работает со стеком TCP/IP без IP-адреса, что явно ограничивает его возможности обмена информацией. Однако он может отправить широковещательное сообщение DHCPDISCOVER, чтобы попытаться обнаружить какой-либо сервер DHCP. Широковещательные сообщения обычно ограничены локальным сегментом сети, но поскольку протокол DHCP является открытым стандартом, он поддерживается многими маршрутизаторами, что позволяет им распространять широковещательные сообщения DHCP через границы сети. Тем самым один сервер DHCP может поддерживать клиентов в нескольких сетевых сегментах.

Пакет DHCPDISCOVER содержит MAC-адрес рабочей станции, что позволяет серверам DHCP отвечать конкретно направленными, а не широковещательными сообщениями. Все серверы DHCP, получившие это широковещательное сообщение, обязаны ответить данному клиенту пакетом DHCPOFFER, содержащим IP-адрес и другие настройки конфигурации на рассмотрение клиента. Если клиент получил несколько пакетов DHCPOFFER, он выбирает один из них и отправляет широковещательное сообщение DHCPREQUEST, содержащее IP-адрес и настройки, которые он намеревается принять. Это широковещательное сообщение используется, чтобы информировать выбранный сервер о согласии клиента, а также уведомить остальные серверы, что их предложения отклонены.

В течение этого периода IP-адрес, предложенный данным сервером, еще не окончательно выделен данному клиенту. При определенных обстоятельствах за этот период такие же самые настройки могут быть предложены и другому потенциальному клиенту. Однако, получив сообщение DHCPREQUEST, сервер фиксирует предложенные настройки за данным клиентом, записывая их в свою базу данных и устанавливая клиента в состояние bound (привязка). Затем сервер отправляет данному клиенту пакет DHCPACK, информируя его о своем подтверждении. Если по какой-либо причине процесс аренды адреса не может быть

завершен, то сервер отправляет пакет DHCPNACK и клиент начинает весь процесс заново, отправляя новый пакет DHCPDISCOVER.

Получив пакет DHCPACK, клиент выполняет окончательную проверку предложенного IP-адреса, используя протокол ARP – Address Resolution Protocol (протокол разрешения адресов), чтобы выяснить, нет ли дублирования этого адреса в сети. Если такой адрес найден, то клиент отправляет серверу пакет DHCPDECLINE, отменяя всю транзакцию. Если нет, то эти настройки используются для конфигурирования стека TCP/IP, после чего можно начать вход в сеть.

После согласования аренды клиент DHCP имеет право использовать выделенные ему настройки на период, который задан на сервере. По умолчанию период аренды составляет восемь дней. При каждом входе в сеть рабочая станция обновляет эту аренду, отправляя широковещательное сообщение DHCPREQUEST, содержащее cookie-файл с идентификацией аренды (комбинация из MAC-адреса и IP-адреса, которая уникальным образом идентифицирует аренду для сервера).

При обычных условиях сервер отвечает, как и раньше, сообщением DHCPACK. Но если сервер обнаруживает, что данный клиент находится не в той подсети, где он находился во время согласования аренды, то сервер отправляет сообщение DHCPNACK, прекращая текущую аренду и вынуждая начать согласование новой аренды. Если клиент не получает никакого ответа после десяти попыток, то он отправляет широковещательное сообщение DHCPDISCOVER, надеясь получить новую аренду.

Если прошло 50 % времени от текущего периода аренды, то клиент переходит из состояния bound (привязка) в состояние renewing (обновление). После этого сообщения DHCPREQUEST отправляются уже как одиночные (не широковещательные) сообщения на сервер, предоставивший данную аренду. Если прошло 87,5 % времени от периода аренды, то клиент переходит в состояние rebinding (повторение процесса привязки), снова начиная отправлять широковещательные сообщения DHCPREQUEST, запрашивая ответ от любого сервера DHCP. По истечении всего периода аренды без ответа от какого-либо сервера DHCP данный клиент переходит в состояние unbound (без привязки), после чего он проходит через процесс собственного автоматического конфигурирования для получения IP-адреса класса и маски подсети.

Помимо сообщений, необходимых для первоначального получения IP-адреса клиентом, DHCP предусматривает несколько дополнительных сообщений для выполнения иных задач.

Отказ DHCP. Если после получения подтверждения (DHCPACK) от сервера клиент обнаруживает, что указанный сервером адрес уже используется в сети, он рассылает широковещательное сообщение отказа DHCP (DHCPDECLINE), после чего процедура получения IP-адреса повторяется. Использование IP-адреса другим клиентом можно обнаружить, выполнив запрос ARP.

Сообщение отмены DHCP (DHCPNAK). При получении такого сообщения соответствующий клиент должен повторить процедуру получения адреса.

Освобождение DHCP. Клиент может явным образом прекратить аренду IP-адреса. Для этого он отправляет сообщение освобождения DHCP (DHCPRELEASE) тому серверу, который предоставил ему адрес в аренду. В отличие от других сообщений DHCP, DHCPRELEASE не рассылается широковещательно.

Информация DHCP. Сообщение информации DHCP (DHCPINFORM) предназначено для определения дополнительных параметров TCP/IP (например, адреса маршрутизатора по умолчанию, DNS-серверов и т. п.) теми клиентами, которым не нужен динамический IP-адрес (т. е. адрес которых настроен вручную). Серверы отвечают на такой запрос сообщением подтверждения (DHCPACK) без выделения IP-адреса.

7.2. Задание для выполнения лабораторной работы

Используя знания из предыдущих лабораторных работ – о процессах создания и настройки сетей в программном пакете Cisco Packet Tracer, необходимо создать компьютерную сеть и настроить работу DNS- и DHCP-серверов.

1. Реализовать в программном пакете Cisco Packet Tracer сеть передачи данных, топология которой показана на рис. 7.3.

В этой сети необходимо обеспечить доступ к сайту, расположенному на веб-сервере, с любого из компьютеров, входящих в сеть.

2. Произвести настройку адресации в данной сети, используя IP-адреса, указанные в табл. 7.3.

Таблица 7.3

Требуемые настройки сети

Номер сети	Маршрутизатор	Адрес сети	Маска
1	R1	192.168.1.0	255.255.255.128
2	R2	192.168.2.0	255.255.255.128
3	R3	192.168.3.0	255.255.255.128
4	R1, R2	10.0.0.0	255.0.0.0
5	R1, R3	20.0.0.0	255.0.0.0
6	R2,R3	30.0.0.0	255.0.0.0

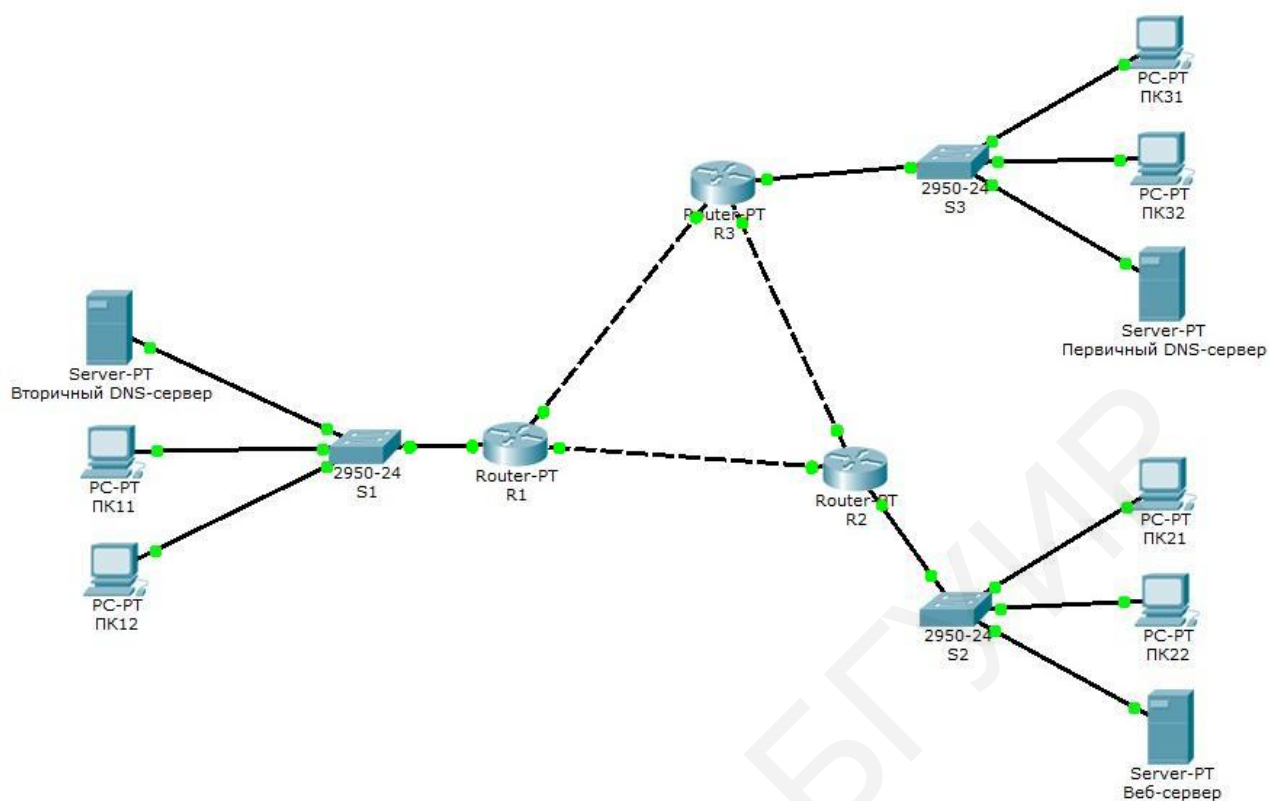


Рис. 7.3. Топология сети передачи данных с DNS-серверами

3. Настроить маршрутизацию трафика в сети. Маршруты можно задать статически или использовать протокол OSPF.

4. Настроить DNS-сервер. Для этого в настройках DNS-сервера (рис. 7.4) требуется создать ресурсную запись А-типа, в которой указать соответствие между DNS-именем и IP-адресом веб-сервера, например:

192.168.1.1 a test.local

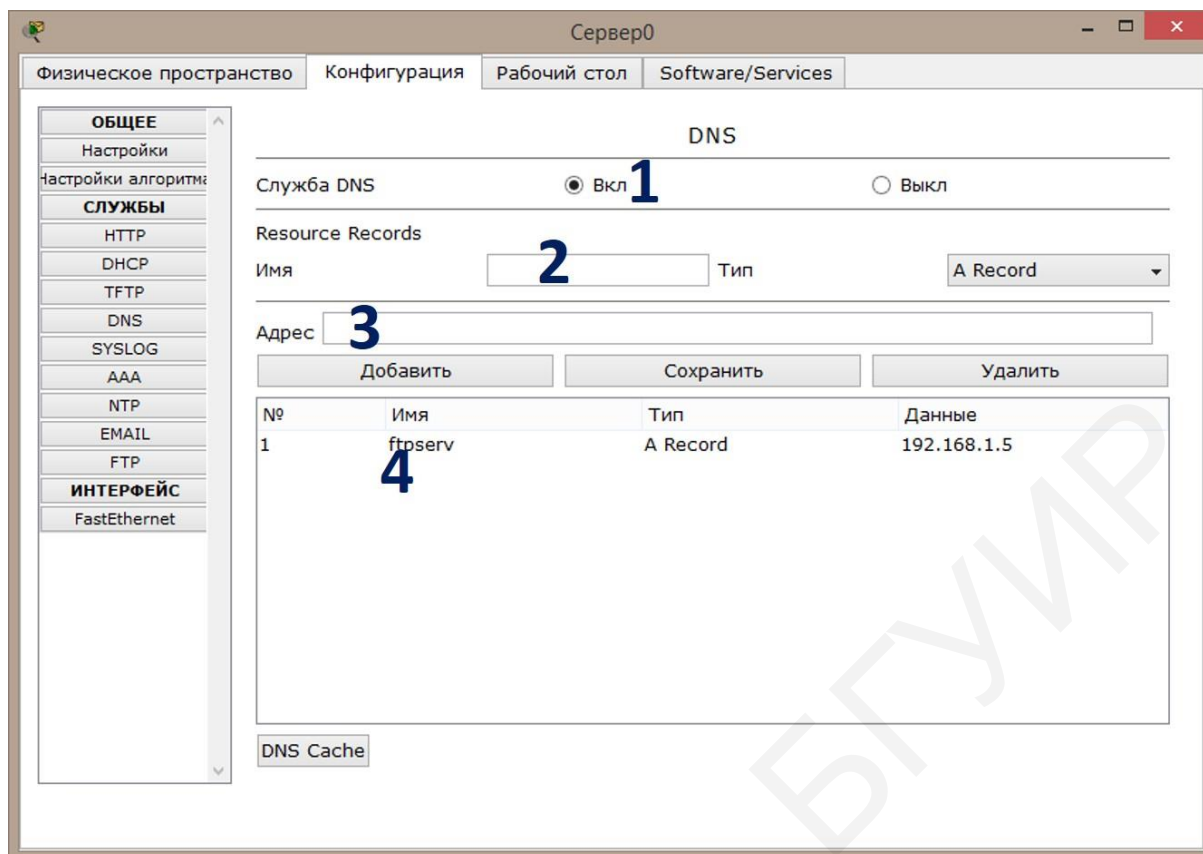


Рис. 7.4. Создание ресурсной записи на DNS-сервере:

1 – включение режима работы в качестве DNS-сервера; 2 – задание имени, по которому будет доступен хост (например, ftpserv); 3 – строка для задания IP-адреса хоста; 4 – после нажатия «Добавить» появляется соответствующая запись

5. Зайти на указанный по DNS-имени сайт с любого компьютера в сети. Убедиться, что данный сайт доступен с любого компьютера.

6. Настроить для созданного домена второе имя. Для этого следует использовать ресурсную запись CNAME. Для этого в поле 2 указать имя сервера, к которому будем обращаться, выбрать тип записи CNAME и в поле 3 указать IP-адрес.

7. Убедиться, что сайт также доступен с любого компьютера в сети по новому адресу.

8. Настроить на первичном DNS-сервере ресурсную запись SOA-типа. Для данной записи требуется указать следующие настройки информационных полей.

Поле Serial Number – серийный номер версии файла зоны. Может быть любым положительным целым числом.

Поле Primary Name Server – адрес первичного DNS-сервера для данного домена. Указать IP-адрес DNS-сервера в сети.

Поле Refresh Time – временной параметр Refresh показывает, как часто вторичные серверы должны запрашивать первичный сервер, чтобы узнать, не

увеличился ли Serial number (серийный номер) зоны и, следовательно, не нужно ли обновить ее у себя. Задать Refresh Time = 3000 с.

Поле Retry Time – показывает, как долго вторичный сервер имен должен ждать, перед тем как повторить попытку запроса первичного сервера (на предмет изменений серийного номера данной зоны), если предыдущая попытка оказалась неудачной. Указать значение 3600 с.

Поле Expire Time – указывает верхнее ограничение по времени, в течение которого вторичный сервер может использовать ранее полученные данные о зоне, до того как они потеряют силу из-за отсутствия обновления (например, вследствие отключения первичного сервера имен на длительное время). Указать значение 86 400 с.

Поле Minimum TTL – определяет «время жизни» отрицательных ответов на запросы о ресурсах, не существующих в DNS. Допустимые значения: не менее 5 мин. В настройках указать 3600 с.

Поле Mail box – почтовый адрес ответственного лица.

9. Настроить запись NS-типа. Запись NS (Name Server) указывает на DNS-сервер для данного домена. Для стабильной работы домена указывается не менее двух NS-записей. В случае недоступности одного из DNS-серверов должен производиться запрос на другой DNS-сервер.

10. Отключить один из DNS-серверов и зайти на веб-сайт с любого компьютера в сети. Пояснить полученный результат.

11. Настроить работу DHCP-сервера в сети. В качестве DHCP-сервера может быть использован выделенный сервер или же маршрутизатор. Для начала настроим работу сети без выделенного DHCP-сервера. В этом случае DHCP-сервер должен быть развернут на маршрутизаторе.

Используем сеть №1, развернутую на маршрутизаторе R1.

Произвести настройку маршрутизатора R1. Проверить настройку интерфейса, которым маршрутизатор R1 соединен с коммутатором S1 (должен быть включен и присвоен IP-адрес).

Далее необходимо создать DHCP-пул, т. е. пространство IP-адресов. Данная процедура выполняется в режиме глобального конфигурирования маршрутизатора R1 с помощью команды «*ip dhcp pool <имя пула>*», после чего задается диапазон IP-адресов командой «*network <адрес сети> <маска сети>*». Хосту необходимо выдать не только IP-адрес, но и шлюз по умолчанию. Это выполняется посредством команды «*default-router <IP-адрес шлюза по умолчанию>*». Далее указывается адрес DNS-сервера посредством команды «*dns-server <IP-адрес DNS-сервера>*».

Для исключения определенных IP-адресов из выдачи можно использовать команду «*ip dhcp excluded-address <IP-адрес>*».

Сохранить конфигурацию маршрутизатора.

12. На хостах сети №1 установить получение IP-адресов по DHCP.

С помощью команды «*ip dhcp binding*» посмотреть, какие IP-адреса и

каким хостам были выданы.

13. Создать сеть с выделенным DHCP-сервером, показанную на рис. 7.5 (или дополнить имеющуюся сеть необходимыми устройствами и произвести ее настройку).

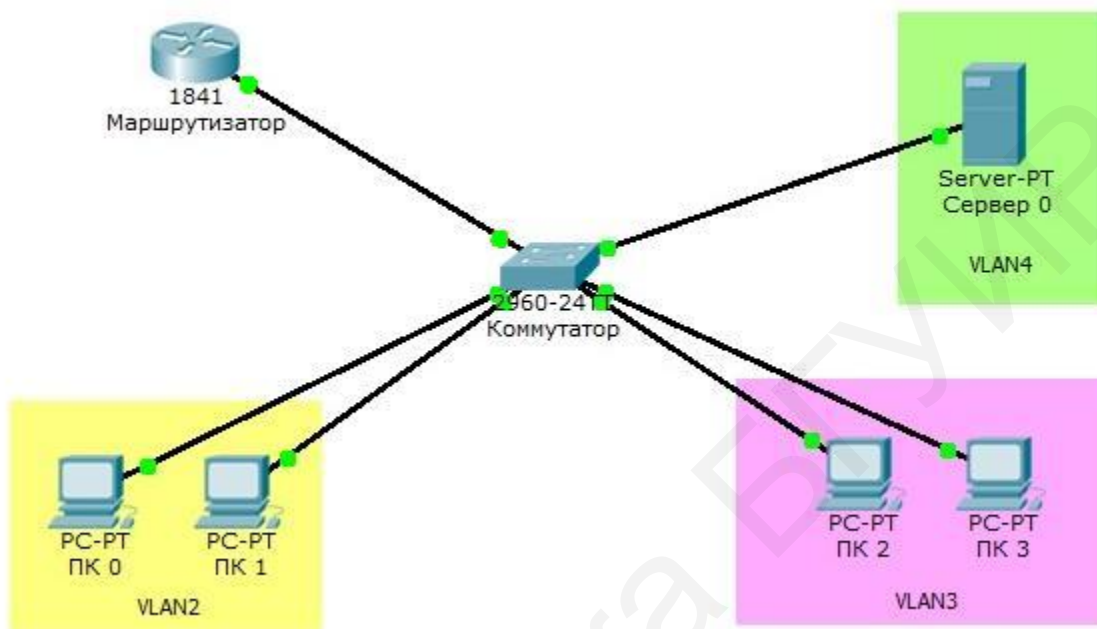


Рис. 7.5. Топология сети с выделенным DHCP-сервером

14. Произвести настройку указанной сети. В данном случае сеть разделена на сегменты – виртуальные сети. Сервер необходимо выделить в отдельный сегмент, отличный от сегмента пользователей. Необходимо сегментировать сеть, т. е. настроить виртуальные локальные сети на коммутаторе. Весь трафик виртуальных локальных сетей перенаправлять для маршрутизации на маршрутизатор. Сохранить конфигурацию на коммутаторе.

На маршрутизаторе необходимо включить физический интерфейс, которым маршрутизатор соединен с коммутатором, настроить субинтерфейсы и сохранить конфигурацию.

15. Настроить DHCP-сервер. Для этого в первую очередь серверу необходимо задать статический IP-адрес, маску сети и шлюз по умолчанию. Далее включить режим работы в качестве DHCP-сервера (позиция 1 на рис. 7.6).

Один пул адресов уже задан. Для создания нового пула необходимо задать его имя (позиция 2 на рис. 7.6), шлюз по умолчанию для данного пула (позиция 3 на рис. 7.6), IP-адрес DNS-сервера (позиция 4 на рис. 7.6), начальный IP-адрес для выдачи и маску сети (позиция 5 на рис. 7.6) и нажать кнопку «Добавить».

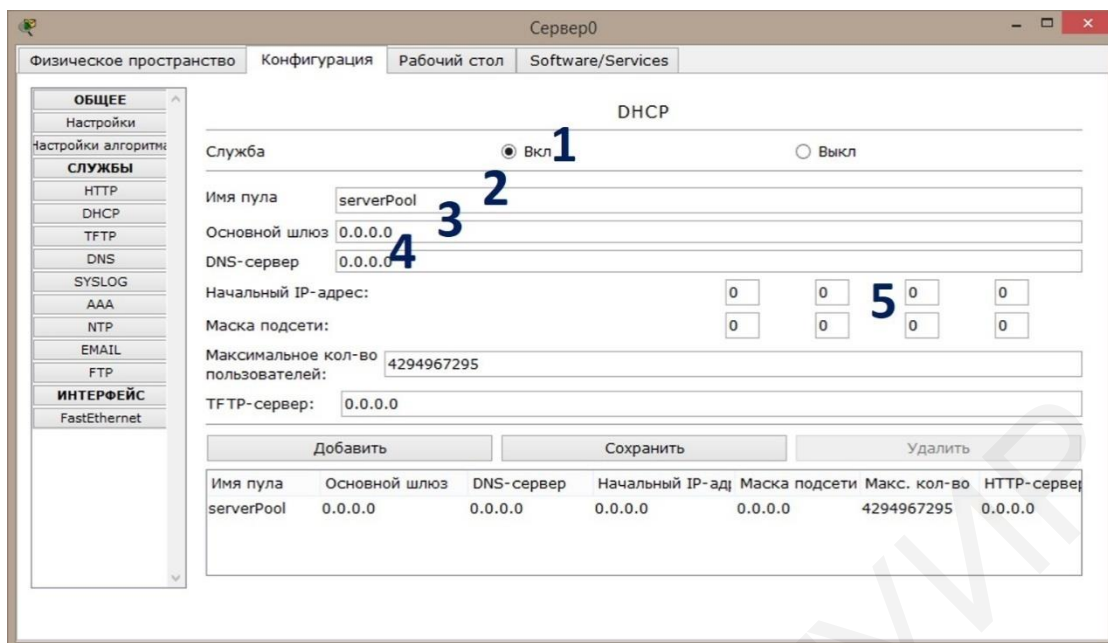


Рис. 7.6. Настройка DHCP-сервера

Так как DHCP-сервер находится в отдельном сегменте, то широковещательные запросы от хостов для поиска DHCP-сервера через маршрутизатор не будут проходить. Для переадресации запросов компьютеров на DHCP-сервер необходимо организовать перенаправление DHCP-запросов (функция DHCP-Relay). Данная функция настраивается на маршрутизаторе. Для каждого подынтерфейса необходимо настроить перенаправление DHCP-запросов на существующий DHCP-сервер. Данная операция выполняется в режиме глобального конфигурирования. Необходимо выбрать подынтерфейс, для которого будет проводиться настройка, например *intgi0/0.2*, и задать перенаправление DHCP-запросов на существующий DHCP-сервер посредством команды «*ip helper-address <IP-адрес DHCP-сервера>*». Данная процедура выполняется для каждого подынтерфейса, для которого необходимо реализовать перенаправление широковещательных запросов. Сохранить конфигурацию маршрутизатора.

16. На компьютерах установить автоматическое получение IP-адресов и проверить функционирование сети.

7.3. Содержание отчета

1. Титульный лист.
2. Цель работы.
3. Схема реализованных топологий сети.
4. Конфигурация коммутаторов и маршрутизаторов (списки настроенных виртуальных локальных сетей, интерфейсов, таблицы маршрутизации) и настройки серверов DNS и DHCP.

5. Результаты прохождения пакетов данных между хостами по ходу выполнения работы.
6. Выводы.

7.4. Контрольные вопросы

1. Что такое система доменных имен?
2. Что такое домен?
3. Какие группы доменных имен существуют?
4. Как осуществляется процедура определения IP-адреса в системе DNS?
5. Что такое ресурсная запись и для чего она предназначена?
6. Поясните структуру DNS-сообщения.
7. Назовите назначение и принцип функционирования протокола DHCP.
8. Назовите способы назначения IP-адресов клиентам, которые можно реализовать, используя протокол DHCP.
9. Поясните принцип настройки DNS-сервера в программном пакете Cisco Packet Tracer.
10. Чем отличается настройка сети, не имеющей выделенного DHCP-сервера, от настройки сети с выделенным сервером?

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Олифер, В. Г. Компьютерные сети. Принципы, технологии, протоколы / В. Г. Олифер, Н. А. Олифер. – 5-е изд. – СПб. : Питер, 2015. – 992 с.
2. Олифер, В. Г. Безопасность компьютерных сетей / В. Г. Олифер, Н. А. Олифер. – М. : Горячая линия – Телеком, 2016. – 643 с.
3. Таненбаум, Э. Компьютерные сети: учеб. пособие / Э. Таненбаум, Д. Уэзеролл. – 5-е изд. – СПб. : Питер, 2012. – 960 с.
4. Cisco Systems, Inc. [Электронный ресурс]. – 2018. – Режим доступа : <https://www.cisco.com>. – Дата доступа : 05.09.2018.
5. Bonaventure, O. Computer Networking : Principles, Protocols and Practice / O. Bonaventure [Электронный ресурс]. – 2018. – Режим доступа : <http://cnp3book.info.ucl.ac.be/2nd/cnp3bis.pdf>. – Дата доступа : 13.12.2018.
6. Internet Corporation for Assigned Names and Numbers [Электронный ресурс]. – 2018. – Режим доступа : <https://www.icann.org>. – Дата доступа : 05.09.2018.
7. Internet Assigned Numbers Authority [Электронный ресурс]. – 2018. – Режим доступа : <https://www.iana.org/>. – Дата доступа : 05.09.2018.

Учебное издание

Листопад Николай Измайлович
Каленкович Евгений Николаевич

***СИСТЕМЫ И СЕТИ ПЕРЕДАЧИ ДАННЫХ. ЗАЩИТА
ИНФОРМАЦИИ В КОМПЬЮТЕРНЫХ СЕТЯХ.
ЛАБОРАТОРНЫЙ ПРАКТИКУМ***

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ

Редактор *Е. И. Костина*
Корректор *Е. Н. Батурчик*
Компьютерная правка, оригинал-макет *О. И. Толкач*

Подписано в печать 24.01.2020. Формат 60×84 1/16. Бумага офсетная. Гарнитура «Таймс».
Отпечатано на ризографе. Усл. печ. л. 6,86. Уч.-изд. л. 7,2. Тираж 50 экз. Заказ 340.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники».
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий №1/238 от 24.03.2014,
№2/113 от 07.04.2014, №3/615 от 07.04.2014.
Ул. П. Бровки, 6, 220013, г. Минск