Белорусский национальный технический университет
Факультет информационных технологий и робототехники
Кафедра «Робототехнические системы»

СОГЛАСОВАНО
Заведующий кафедрой
_____
___ _____ 2018 г.

СОГЛАСОВАНО
Декан факультета
_____Е.Е. Трофименко
___ _____ 2018 г.

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

Локальные вычислительные сети

для специальности:
1-53 01 01 «Автоматизация технологических процессов и производств»

Составитель: Дубинин С.В.

Рассмотрено и утверждено
на заседании совета факультета информационных технологий
и робототехники мая 2018 г.,
протокол N

Перечень материалов

Электронный учебно-методический комплекс включает:
- основные теоретические сведения,
- лабораторные работы,
- экзаменационные вопросы,
- программу дисциплины «Локальные вычислительные сети»
- список литературы.

Пояснительная записка

Электронный учебно-методический комплекс разработан для студентов специальности 1-53 01 01 «Автоматизация технологических процессов и производств».

Информационное наполнение ЭУМК соответствует программе дисциплины «Локальные вычислительные сети».

Комплекс предназначен для студентов дневного и заочного отделений.

Внедрение ЭУМК будет способствовать более эффективному овладению теоретическими и практическими основами разработки, настройки и администрированию локальных вычислительных сетей.

ЭУМК разработан в виде UMKPUTO.pdf - файла, что делает его универсальным и позволяет применять как на локальном компьютере, так и в локальной или глобальной сети. ЭУМК не требует установки специального программного обеспечения. Для работы с ним достаточно иметь операционную систему семейства WINDOWS.

ЭУМК может использоваться для изучения теоретических основ дисциплины, при проведении лабораторных, контрольных работ, а также в ходе подготовки студентов к экзамену по дисциплине «Локальные вычислительные сети».

Содержание

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1. ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ ЛВС

Локальная вычислительная сеть (**ЛВС**, *локальная сеть*; англ. *Local Area Network, LAN*) — компьютерная сеть, покрывающая обычно относительно небольшую территорию или небольшую группу зданий (дом, офис, фирму, институт).

1.1.1. Классификация и характеристики локальных вычислительных сетей

Вопросы для изучения:

- Основные характеристики ЛВС;
- Основные компоненты компьютерных сетей;
- Уровневая организация взаимодействия по сети.

Основные характеристики ЛВС:

- территориальная протяженность сети (длина общего канала связи);
- максимальная скорость передачи данных;
- максимально возможное расстояние между рабочими станциями сети;
- топология сети;
- вид физической среды передачи данных;
- максимальное число каналов передачи данных;
- тип передачи сигналов (синхронный или асинхронный);
- метод доступа абонентов в сети;
- структура программного обеспечения сети;
- возможности передачи речи и видеосигнала;
- условия надежной работы сети;
- возможности связи ЛВС между собой и с сетью более высокого уровня;
- возможность использования процедуры установления приоритетов при одновременном подключении абонентов к общему каналу.

Основные компоненты компьютерных сетей

Интерфейс — в широком смысле — формально определенная логическая и (или) физическая граница между взаимодействующими независимыми объектами. Интерфейс задает параметры, процедуры и характеристики взаимодействия объектов.

Разделяют физический и логический интерфейсы.

Физический интерфейс (называемый также портом) определяется набором электрических связей и характеристиками сигналов. Обычно он представляет собой разъем с набором контактов, каждый из которых имеет определенное назначение, например, это может быть группа конактов для передачи данных, контакт синхронизации данных и т. п. Пара разъемов

соединяется кабелем, состоящим из набора проводов, каждый из которых соединяет соответствующие контакты.

Логический интерфейс (называемый также **протоколом**) — это набор информационных сообщений определенного формата, которыми обмениваются два устройства или две программы (в данном случае компьютер и периферийное устройство), а также набор правил, определяющих логику обмена этими сообщениями.

Стеком протоколов называют иерархически упорядоченный набор протоколов, каждый из которых необходим в процессе обмена данными по сети. Пример стека протоколов: TCP/IP.

Клиент — это модуль, предназначенный для формирования и передачи сообщений-запросов к ресурсам удаленного компьютера от разных приложений с последующим приемом результатов из сети и передачей их соответствующим приложениям.

Сервер — это модуль, который постоянно ожидает прихода из сети запросов от клиентов и, приняв запрос, пытается его обслужить, как правило, с участием локальной ОС; один сервер может обслуживать запросы сразу нескольких клиентов (поочередно или одновременно).

Пара клиент—сервер, предоставляющая доступ к конкретному типу ресурса компьютера через сеть, образует **сетевую службу**.

Услуги, предоставляемые службой, называются **сервисом**.

В процессе передачи сообщения между протоколами соседних уровней, каждый модуль добавляет к сообщению свои управляющие данные и «заворачивает» данные в кадр, формат которого определен данным протоколом. Этот процесс называется **инкапсуляцией** данных. В узле – получателе данных выполняется обратный процесс.

Уровневая организация взаимодействия по сети

В 1984 г. ряд организаций стандартизации разработал модель взаимодействия открытых систем ISO/OSI или **семиуровневую модель.**

Модель OSI описывает следующие семь уровней:

1. Физический;
2. Канальный;
3. Сетевой;
4. Транспортный;
5. Сеансовый;
6. Представительный;
7. Прикладной.

Уровни расположены снизу-вверх в порядке возрастания.

На **физическом уровне** определяются физические характеристики передающей среды, стандарты сетевых разъемов, тип и характеристики кабеля, способ представления двоичной информации при помощи электрических сигналов и т.д. Данные физического уровня представляют собой набор бит.

На **канальном уровне** определяется способ доступа к среде передачи данных и выполняется передача данных внутри сети с заданной топологией.

Способ доступа к среде передачи данных определяет какая станция в какой момент времени может передавать данные в разделяемой среде (по общему куску кабеля, например). На канальном уровне формируется кадр данных. Отправитель и получатель определяются при помощи своего физического (MAC-) адреса, который зашит в сетевой карте производителем.

На **сетевом уровне** данные могут передаваться между сетями с заданной стандартной топологией. Адрес сетевого уровня содержат номер сети и номер узла в сети. Для определения оптимального маршрута между отправителем и получателем данных используется устройство – маршрутизатор. При помощи маршрутизатора выполняется также объединение подсетей.

На **транспортном уровне** выполняется передача данных с требуемой степенью надежности между приложениями. Если необходима надежная передача данных, то выбирается протокол с предварительным установлением соединения, обеспечением подтверждения и проверкой правильности приема данных.

На **сеансовом уровне** выполняется управление диалогом: определяется какая станция является активной, поддерживается механизм контрольных точек (позволяет выполнить «откат» во время сеанса).

На **представительном уровне** выполняется преобразование данных без изменения их содержания (протоколы шифрования/дешифрования).

На **прикладном уровне** пользователь получает доступ к сетевым службам (почта, интернет и т.д.).

1.1.2. Топологии ЛВС

Вопросы для изучения:

- Классификация топологий компьютерных сетей. Принципы работы, достоинства и недостатки.

Сетевая топология — это конфигурация графа, вершинам которого соответствуют конечные узлы сети (компьютеры) и коммуникационное оборудование (маршрутизаторы), а рёбрам — физические или информационные связи между вершинами.

Среди множества возможных конфигураций различают **полносвязные** и **неполносвязные**.

Полносвязная топология соответствует сети, в которой каждый компьютер непосредственно связан со всеми остальными (рис. 1.1, а). Несмотря на логическую простоту, этот вариант оказывается громоздким и неэффективным. Действительно, в таком случае каждый компьютер в сети должен иметь большое количество коммуникационных портов, достаточное для связи с каждым из остальных компьютеров сети. Для каждой пары компьютеров должна быть выделена отдельная физическая линия связи. (В некоторых случаях даже две, если невозможно использование этой линии для двусторонней передачи.) Полносвязные топологии в крупных сетях

применяются редко, так как для связи N узлов требуется N (N-1)/2 физических дуплексных линий связей, то есть имеет место квадратичная зависимость от числа узлов. Чаще этот вид топологии используется в многомашинных комплексах или в сетях, объединяющих небольшое количество компьютеров.

Все другие варианты основаны на **неполносвязных топологиях**, когда для обмена данными между двумя компьютерами может потребоваться транзитная передача данных через другие узлы сети.

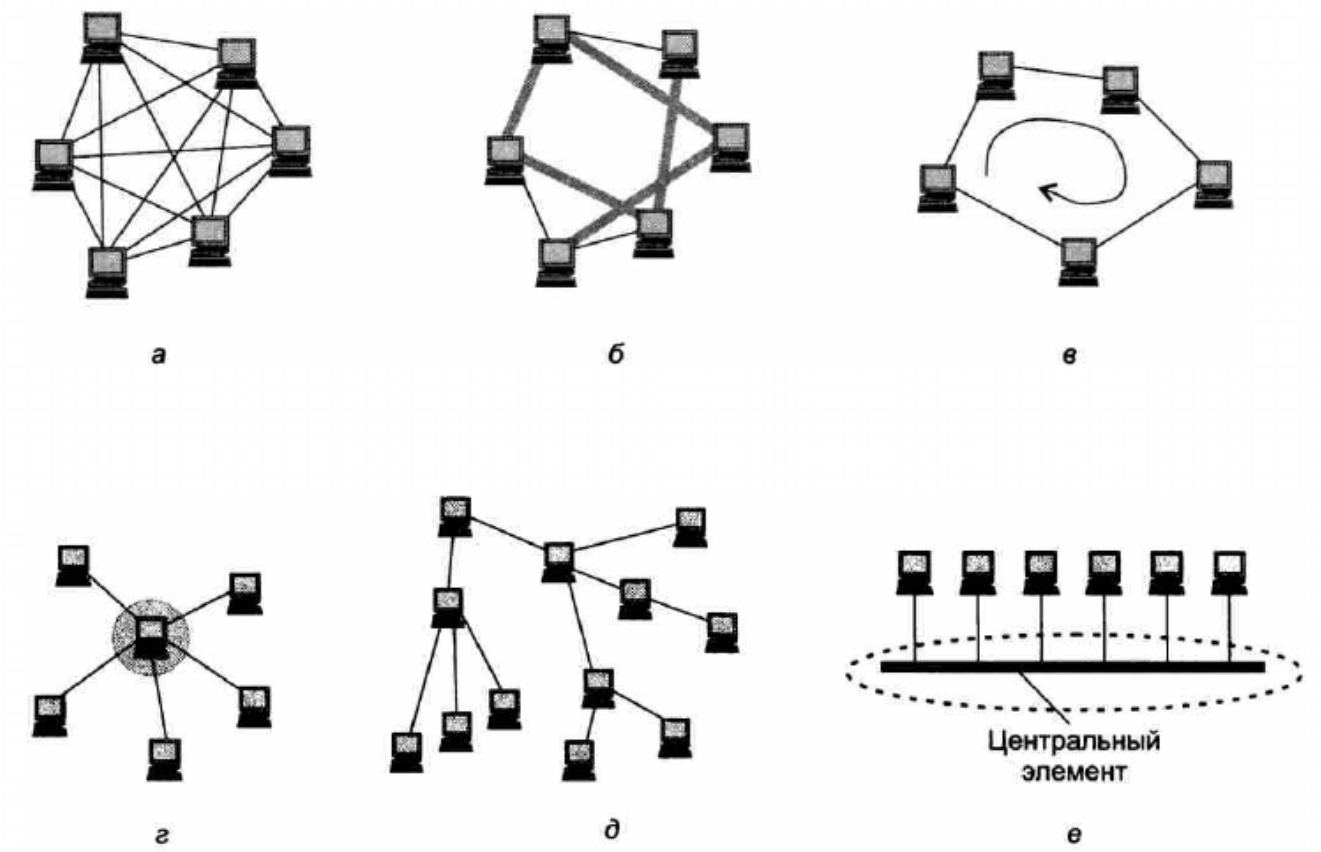


Рисунок 1.1 — Типовые топологии сетей

Ячеистая топология получается из полносвязной путем удаления некоторых связей (рис. 1.1, б). Ячеистая топология допускает соединение большого количества компьютеров и характерна, как правило, для крупных сетей.

В сетях с **кольцевой топологией** (рис. 1.1, в) данные передаются по кольцу от одного компьютера к другому. Главным достоинством кольца является то, что оно по своей природе обеспечивает резервирование связей. Действительно, любая пара узлов соединена здесь двумя путями — по часовой стрелке и против нее. Кроме того, кольцо представляет собой очень удобную конфигурацию для организации обратной связи — данные, сделав полный оборот, возвращаются к узлу-источнику. Поэтому источник может контролировать процесс доставки данных адресату. Часто это свойство кольца используется для тестирования связности сети и поиска узла, работающего некорректно. В то же время в сетях с кольцевой топологией необходимо принимать специальные меры, чтобы в случае выхода из строя или отключения какого-либо компьютера не прерывался канал связи между остальными узлами кольца.

Звездообразная топология (рис. 1.1, г) образуется в случае, когда каждый компьютер подключается непосредственно к общему центральному устройству, чаще всего коммутатору. В функции концентратора входит направление передаваемой компьютером информации одному или всем остальным компьютерам сети. В качестве концентратора может выступать как универсальный компьютер, так и специализированное устройство. К недостаткам звездообразной топологии относится более высокая стоимость сетевого оборудования из-за необходимости приобретения специализированного центрального устройства. Кроме того, возможности по наращиванию количества узлов в сети ограничиваются количеством портов концентратора.

Иногда имеет смысл строить сеть с использованием нескольких концентраторов, иерархически соединенных между собой звездообразными связями (рис. 1.1, д). Получаемую в результате структуру называют **иерархической звездой**, или **деревом**. В настоящее время дерево является самой распространенной топологией связей как в локальных, так и глобальных сетях.

Особым частным случаем звезды является **общая шина** (рис. 1.1, ё). Здесь в качестве центрального элемента выступает пассивный кабель, к которому по схеме «монтажного ИЛИ» подключается несколько компьютеров (такую же топологию имеют многие сети, использующие беспроводную связь, — роль общей шины здесь играет общая радиосреда). Передаваемая информация распространяется по кабелю и доступна одновременно всем компьютерам, присоединенным к этому кабелю. Основными преимуществами такой схемы являются ее дешевизна и простота присоединения новых узлов к сети, а недостатками — низкая надежность (любой дефект кабеля полностью парализует всю сеть) и невысокая производительность (в каждый момент времени только один компьютер может передавать данные по сети, поэтому пропускная способность делится здесь между всеми узлами сети).

В то время как небольшие сети, как правило, имеют типовую топологию —звезда, кольцо или общая шина, для крупных сетей характерно наличие произвольных связей между компьютерами. В таких сетях можно выделить отдельные произвольно связанные фрагменты (подсети), имеющие типовую топологию, поэтому их называют сетями со **смешанной топологией** (рис. 1.2).

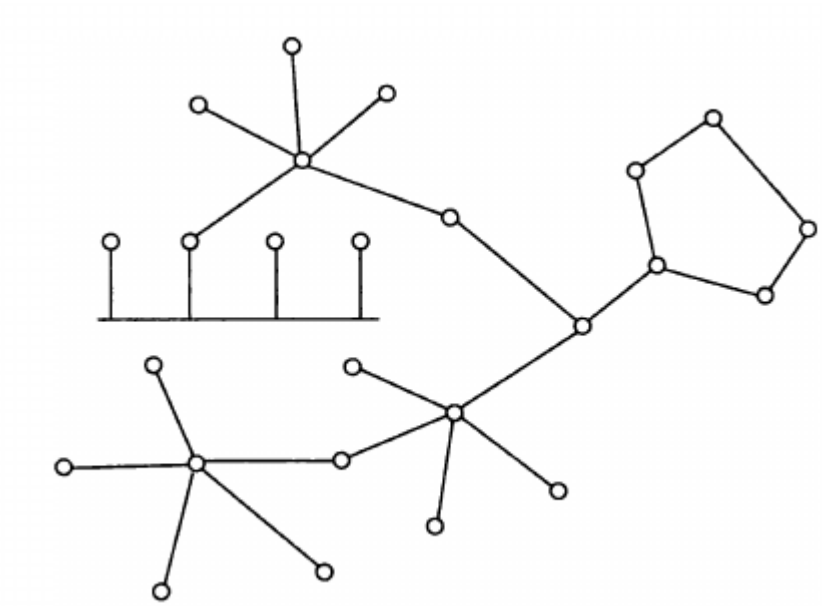


Рисунок 1.2 — Смешанная топология

1.2. ФИЗИЧЕСКАЯ СРЕДА ПЕРЕДАЧИ ДАННЫХ

1.2.1. Параметры линий связи

Вопросы для изучения:

- Линии связи и характеристики линий связи. Полоса пропускания и пропускная способность;
- Методы кодирования информации в линиях связи;
- Теорема Шенона-Хартли.

Линия связи включает передающую среду, аппаратуру передачи данных и промежуточную аппаратуру.

В зависимости от среды передачи данных линии связи разделяют на:

- Проводные (воздушные);
- Кабельные (медные и оптоволоконные);
- Радиоканалы наземной и спутниковой связи.

Проводные линии представляют собой провода, проложенные между столбами и висящие в воздухе. Эти линии традиционно используются для телефонной связи.

Кабельные линии состоят из проводника, заключенного в несколько слоев изоляции. Используются три основных типа кабеля: скрученная пара медных проводов, коаксиальный кабель с медной жилой и оптоволоконные кабели. Скрученная пара проводов называется витой парой. Витая пара бывает экранированной (Shielded Twisted Pair - STP), когда пара медных проводов обертывается в изоляционный материал и неэкранированная (Unshielded

Twisted Pair - UTP) – когда оплетка отсутствует. Коаксиальный кабель состоит из внутренней медной жилы и оплетки, которая отделена от жилы слоем изоляции. Существует несколько типов медного кабеля, которые отличаются характеристиками и применяются в локальных, глобальных сетях, в кабельном телевидении. Оптоволоконный кабель состоит из тонких волокон, по которым распространяется сигнал. Он обеспечивает наиболее высокую скорость распространения сигнала и лучшую защиту данных от помех.

Радиоканал образуются при помощи приемника и передатчика радиоволн. Радиоканалы отличаются как используемыми частотными диапазонами, так и дальностью канала.

Аппаратура линий связи включает аппаратуру передачи данных (DCE – Data Circuit terminating Equipment) и аппаратуру пользователя линии данных (DTE – Data Terminal Equipment).

Аппаратура передачи данных (DCE) непосредственно связывает компьютеры или локальные сети с линией связи. Как правило аппаратура передачи данных включается в состав линии связи. Примерами DCE являются модемы, устройства подключения к цифровым каналам.

Аппаратура пользователя линии связи (DTE) называется оконечным оборудованием данных и подключается непосредственно к аппаратуре передачи данных. Примером DTE могут служить компьютеры или маршрутизаторы локальных сетей. Эта аппаратура не включается в состав линии связи.

Промежуточная аппаратура линий связи выполняет две функции:

- улучшение качества сигналов;
- создание постоянного составного канала между двумя абонентами сети.

В качестве промежуточной аппаратуры в глобальных сетях выступают, как правило, мультиплексоры, демультиплексоры, коммутаторы. Эта аппаратура позволяет с высокой скоростью передавать данные нескольких низкоскоростных абонентских линий. Такой канал называют уплотненным каналом.

Промежуточная аппаратура образует **первичную сеть** и служит основой для построения компьютерных, телефонных или иных сетей.

В зависимости от типа промежуточной аппаратуры все сети делятся на аналоговые и цифровые. В аналоговых сетях сигнал имеет непрерывный диапазон значений. Для уплотнения нескольких абонентских аналоговых линий используется техника **частотного мультиплексирования** (FDM – Frequency Division Multiplexing).

В цифровых линиях связи сигнал имеет конечное число состояний и передается импульсом прямоугольной формы. Промежуточная аппаратура в цифровых каналах связи улучшает форму импульса и обеспечивает ресинхронизацию, т.е. восстанавливает период следования импульсов. В цифровых каналах связи используется способ **временного разделения канала** (TDM – Time Division Multiplexing).

Характеристики линий связи. К основным характеристикам линий связи относят следующие:

- амплитудно-частотная характеристика;
- полоса пропускания;
- Затухание;
- помехоустойчивость;
- перекрестные наводки на ближнем конце линии;
- пропускная способность;
- достоверность передачи данных;
- удельная стоимость.

Амплитудно-частотная характеристика показывает, как затухает амплитуда синусоиды на выходе линии связи по сравнению с амплитудой на входе для всех возможных частот передаваемого сигнала.

Полоса пропускания (bandwidth) – это непрерывный диапазон частот, для которого отношение амплитуды выходного сигнала к амплитуде входного сигнала превышает некоторое значение, как правило 0.5. Таким образом определяется диапазон частот, которые передаются без значительного искажения.

Затухание (attenuation) – относительное уменьшение амплитуды при передаче сигнала определенной частоты.

Пропускная способность (throughput) – характеризует максимально возможную скорость передачи данных по линии связи. Измеряется в битах в секунду, а также в килобитах в секунду, мегабитах в секунду и гигабитах в секунду.

Помехоустойчивость – способность линии уменьшать уровень внешних помех, и на внутренних проводниках.

Перекрестные наводки на ближнем конце (NEXT – near end crosstalk) - определяют помехоустойчивость к внутренним источникам помех (электромагнитное поле одной пары проводников наводит на другую пару сигнал помехи). **Достоверность передачи данных** характеризует вероятность искажения, для каждого передаваемого бита данных. Синоним – интенсивность битовых ошибок (Bit Error Rate -BER).

Кодирование

В вычислительной технике для представления данных используется двоичный код.

Внутри компьютера единицам и нулям данных соответствуют дискретные электрические сигналы.

Представление данных в виде электрических или оптических сигналов называется **кодированием**.

Существуют различные способы кодирования двоичных цифр, например потенциальный способ, при котором единице соответствует один уровень напряжения, а нулю —другой, или импульсный способ, когда для представления цифр используются импульсы различной полярности.

Аналогичные подходы применимы для кодирования данных и при передаче их между двумя компьютерами по линиям связи. Однако эти линии связи отличаются по своим характеристикам от линий внутри компьютера.

Главное отличие внешних линий связи от внутренних состоит в их гораздо большей протяженности, а также в том, что они проходят вне экранированного корпуса по пространствам, зачастую подверженным воздействию сильных электромагнитных помех. Все это приводит к существенно большим искажениям прямоугольных импульсов (например, «заваливанию» фронтов), чем внутри компьютера. Поэтому для надежного распознавания импульсов на приемном конце линии связи при передаче данных внутри и вне компьютера не всегда можно использовать одни и те же скорости и способы кодирования. Например, медленное нарастание фронта импульса из-за высокой емкостной нагрузки линии требует, чтобы импульсы передавались с меньшей скоростью (чтобы передний и задний фронты соседних импульсов не перекрывались и импульс успел «дорасти» до требуемого уровня).

В вычислительных сетях применяют как потенциальное, так и импульсное кодирование дискретных данных, а также специфический способ представления данных, который никогда не используется внутри компьютера, — модуляцию (рис. 2.1). При модуляции дискретная информация представляется синусоидальным сигналом той частоты, которую хорошо передает имеющаяся линия связи.

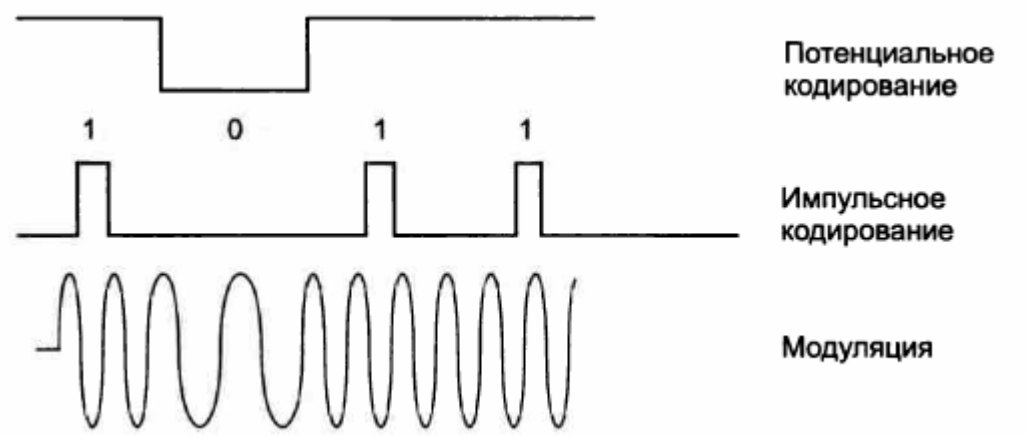


Рисунок 2.1 - Примеры представления дискретной информации

Теорема Шенона-Хартли

Рассматривая все возможные многоуровневые и многофазные методы кодирования, теорема Шеннона — Хартли утверждает, что пропускная способность канала C, означающая теоретическую верхнюю границу скорости передачи данных, которые можно передать с данной средней мощностью сигнала S через аналоговый канал связи, подверженный аддитивному белому гауссовскому шуму мощности N равна:

$$C = B \log_2 \left(1 + \frac{S}{N}\right),$$

где

C - пропускная способность канала, бит/с;

B - полоса пропускания канала, Гц;

S - полная мощность сигнала над полосой пропускания, Вт или В²;

N - полная шумовая мощность над полосой пропускания, Вт или В²;

S/N - отношение мощности сигнала к шуму (SNR).

В данной теореме определено, что достичь максимальной скорости (бит/с) можно путём увеличения полосы пропускания и мощности сигнала и, в то же время, уменьшения шума.

Теорема Шеннона — Хартли ограничивает информационную скорость (бит/с) для заданной полосы пропускания и отношения «сигнал/шум». Для увеличения скорости необходимо увеличить уровень полезного сигнала, по отношению к уровню шума.

Если бы существовала бесконечная полоса пропускания, бесшумовой аналоговый канал, то можно было бы передать неограниченное количество безошибочных данных по ней за единицу времени. Реальные каналы имеют ограниченные размеры и в них всегда присутствует шум.

Удивительно, но не только ограничения полосы пропускания влияют на количество передаваемой информации. Если мы комбинируем шум и ограничения полосы пропускания, мы действительно видим, что есть предел количества информации, которую можно было передать, даже используя многоуровневые методы кодирования. В канале, который рассматривает теорема Шеннона — Хартли, шум и сигнал дополняют друг друга. Таким образом, приёмник воспринимает сигнал, который равен сумме сигналов, кодирующего нужную информацию и непрерывную случайную, которая представляет шум.

Это дополнение создает неуверенность относительно ценности оригинального сигнала. Если приёмник обладает информацией о вероятности ненужного сигнала, который создает шум, то можно восстановить информацию в оригинальном виде, рассматривая все возможные влияния шумового процесса. В случае теоремы Шеннона — Хартли шум, как таковой, произведен гауссовским процессом с некоторыми отклонениями в канале передачи. Такой канал называют *совокупным белым гауссовским шумовым каналом*, так как гауссовский шум является частью полезного сигнала. «Белый» подразумевает равное количество шума во всех частотах в пределах полосы пропускания канала. Такой шум может возникнуть при воздействии случайных источников энергии, а также быть связан с ошибками, возникшими при кодировании. Знание о вероятности возникновения гауссовского шума значительно упрощает определение полезного сигнала.

1.2.2. Методы уплотнения линий связи

Вопросы для изучения:

- Передача дискретных (немодулированных) и гармонических (модулированных) сигналов.

Среды передачи данных предоставляют только потенциальную возможность передачи дискретной информации. Для того чтобы передатчик и приемник, соединенные некоторой средой, могли обмениваться информацией, им необходимо договориться о том, какие сигналы будут соответствовать

двоичным единицам и нулям дискретной информации. Для представления дискретной информации в среде передачи данных применяются сигналы двух типов: прямоугольные импульсы и синусоидальные волны. В первом случае чаще используют термин «кодирование», а во втором — «модуляция», но также можно встретить употребление этих терминов как синонимов.

Модуляция при передаче аналоговых сигналов:

Исторически модуляция начала применяться для аналоговой информации и только потом — для дискретной.

Необходимость в модуляции аналоговой информации возникает, когда нужно передать низкочастотный аналоговый сигнал через канал, находящийся в высокочастотной области спектра. Примером такой ситуации является передача голоса по радио или телевидению. Голос имеет спектр шириной примерно в 10 кГц, а радиодиапазоны включают гораздо более высокие частоты, от 30 кГц до 300 мГц. Еще более высокие частоты используются в телевидении. Очевидно, что непосредственно голос через такую среду передать нельзя.

Для решения проблемы амплитуду высокочастотного несущего сигнала изменяют (модулируют) в соответствии с изменением низкочастотного голосового сигнала (рис. 2.2). При этом спектр результирующего сигнала попадает в нужный высокочастотный диапазон. Такой тип модуляции называется амплитудной модуляцией (Amplitude Modulation, AM).

В качестве информационного параметра используют не только амплитуду несущего синусоидального сигнала, но и частоту. В этих случаях мы имеем дело с частотной модуляцией (Frequency Modulation, FM).

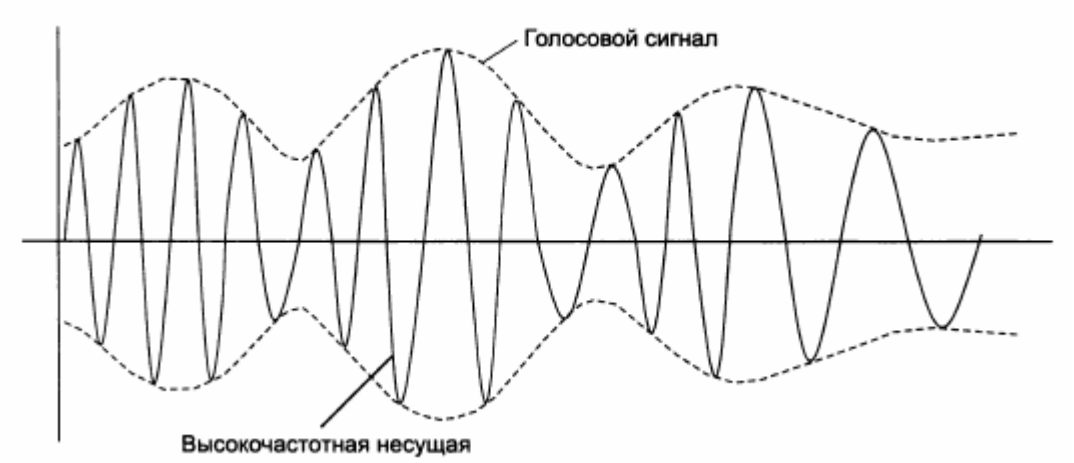


Рисунок 2.2 — Модуляция голосовым сигналом

Модуляция при передаче дискретных сигналов

При передаче дискретной информации посредством модуляции единицы и нули кодируются изменением амплитуды, частоты или фазы несущего синусоидального сигнала. В случае, когда модулированные сигналы передают дискретную информацию, вместо термина «модуляция» иногда используется термин «манипуляция»: амплитудная манипуляция (Amplitude Shift Keying,

ASK), частотная манипуляция (Frequency Shift Keying, FSK), фазовая манипуляция (Phase Shift Keying, PSK).

Пожалуй, самый известный пример применения модуляции при передаче дискретной информации — это передача компьютерных данных по телефонным каналам. Этот составной канал проходит через коммутаторы телефонной сети и соединяет телефоны абонентов. Канал тональной частоты передает частоты в диапазоне от 300 до 3400 Гц, таким образом, его полоса пропускания равна 3100 Гц. Такая узкая полоса пропускания вполне достаточна для качественной передачи голоса, однако она недостаточно широка для передачи компьютерных данных в виде прямоугольных импульсов. Решение проблемы было найдено благодаря аналоговой модуляции. Устройство, которое выполняет функцию модуляции несущей синусоиды на передающей стороне и обратную функцию демодуляции на приемной стороне, носит название модем (модулятор-демодулятор).

На рис. 2.3 показаны различные типы модуляции, применяемые при передаче дискретной информации. Исходная последовательность битов передаваемой информации приведена на диаграмме, представленной на рис. 2.3, а.

При амплитудной модуляции для логической единицы выбирается один уровень амплитуды синусоиды несущей частоты, а для логического нуля — другой (рис. 2.3, б). Этот способ редко используется в чистом виде на практике из-за низкой помехоустойчивости, но часто применяется в сочетании с другим видом модуляции — фазовой модуляцией.

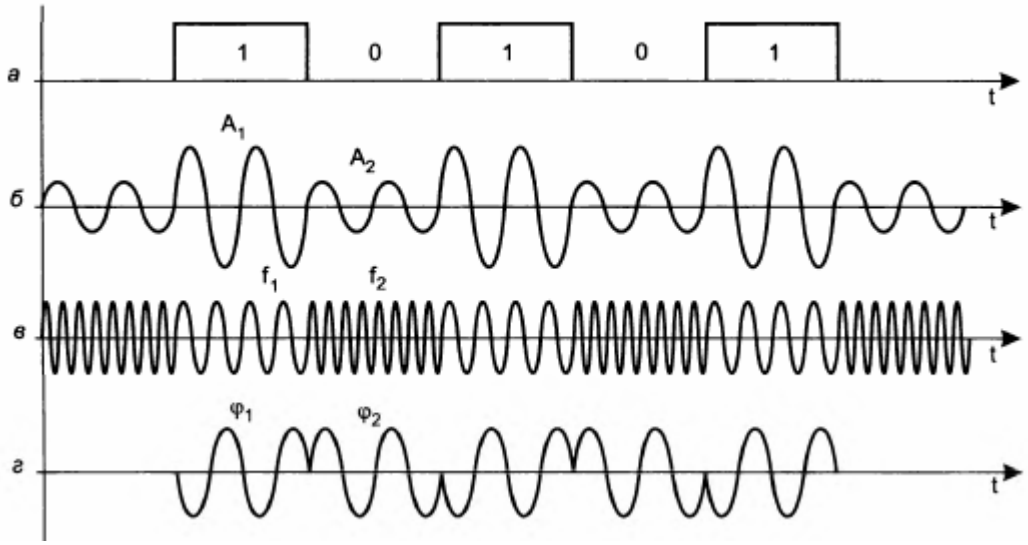


Рисунок 2.3 — Различные типы модуляции

При частотной модуляции значения нуля и единицы исходных данных передаются синусоидами с различной частотой — f_o и f_i (рис. 2.3, в). Этот способ модуляции не требует сложных схем и обычно применяется в низкоскоростных модемах, работающих на скоростях 300 и 1200 бит/с. При использовании только двух частот за один такт передается один бит информации, поэтому такой способ называется **двоичной частотной манипуляцией** (Binary FSK, BFSK). Могут также использоваться четыре различные частоты для кодирования двух битов информации в одном такте,

такой способ носит название **четырехуровневой частотной манипуляции** (four-level FSK). Применяется также название **многоуровневая частотная манипуляция** (Multilevel FSK, MFSK).

При фазовой модуляции значениям данных 0 и 1 соответствуют сигналы одинаковой частоты, но различной фазы, например 0 и 180° или 0, 90,180 и 270° (рис. 2.3, г). В первом случае такая модуляция носит название **двоичной фазовой манипуляции** (Binary PSK, BPSK), а во втором — **квадратурной фазовой манипуляции** (Quadrature PSK, QPSK).

Комбинированные методы модуляции

Для повышения скорости передачи данных прибегают к комбинированным методам модуляции. Наиболее распространенными являются методы квадратурной амплитудной модуляции (Quadrature Amplitude Modulation, QAM). Эти методы основаны на сочетании фазовой и амплитудной модуляции.

На рис. 2.4 показан вариант модуляции, в котором используется 8 различных значений фазы и 4 значения амплитуды. Однако из 32 возможных комбинаций сигнала задействовано только 16, так как разрешенные значения амплитуд у соседних фаз отличаются. Это повышает помехоустойчивость кода, но вдвое снижает скорость передачи данных. Другим решением, повышающим надежность кода за счет введения избыточности, являются так называемые решетчатые коды. В этих кодах к каждым четырем битам информации добавляется пятый бит, который даже при наличии ошибок позволяет с большой степенью вероятности определить правильный набор четырех информационных битов.

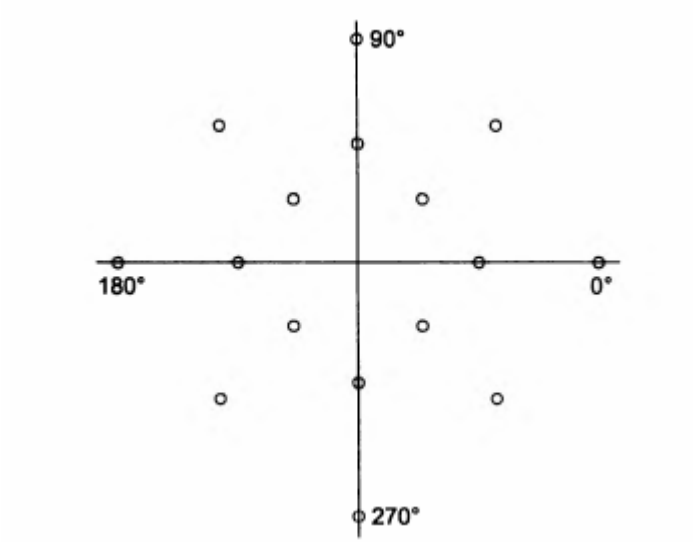


Рисунок 2.4 - Квадратурная амплитудная модуляция
с 16 состояниями сигнала.

1.2.3. Передающая среда

Вопросы для изучения:

- Стандарты кабелей;
- Радио и инфракрасный канал передачи данных;

- Структурированные кабельные системы. Стандарты СКС. Подсистемы СКС.

Кабель – изделие, которое состоит из проводников, слоев экрана и изоляции.

В компьютерных сетях применяются кабели, которые удовлетворяют определенным стандартам. Эти стандарты распространяются также на разъемы и дополнительное оборудование, которое используется для быстрой перекоммутации кабелей.

Общепризнанными являются три стандарта:

- американский стандарт EIA/TIA – 568A;
- международный стандарт ISO/IEC 11801;
- европейский EN50173.

Медный неэкранированный кабель UTP в зависимости от своих электрических и механических характеристик разделяется на 5 категорий. В стандарт EIA/TIA – 568A вошли кабели 3-5 категории. В сетях со скоростями 100 Мбит/с –1Гбит/с используются кабели пятой категории.

Экранированная витая пара описывается стандартами IBM и делится на типы: Type1, … , Type9. Type1 по своим характеристикам примерно соответствует UTP Cat5. Используется в качестве среды передачи данных в сетях Token Ring.

Коаксиальный кабель включает следующие типы коаксиального кабеля:

- RG-8, RG-11 – «толстый коаксиал», используется в сетях Ethernet 10Base-5;
- RG-58/U, RG-58 A/U, RG-58 C/U – «тонкий коаксиал», используется в сетях Ethernet 10Base-2;
- RG-59 – телевизионный кабель, применяется в кабельном телевидении.

Волоконно-оптический кабель состоит из проводника света, который окружен слоем стекла с меньшим показателем преломления. Луч света не выходит за пределы сердцевины кабеля, отражаясь от внешней оболочки. Различают три типа оптоволоконного кабеля:

- многомодовое волокно со ступенчатым преломлением показателя преломления;
- многомодовое волокно с плавным изменением показателя преломления;
- одномодовое волокно.

Понятие «мода» описывает режим распространения световых лучей во внутреннем сердечнике кабеля. В одномодовом кабеле (Single Mode Fiber, SMF) диаметр центрального проводника соизмерим по размеру с длиной волны света. При этом практически все лучи света распространяются вдоль оптической оси световода и не отражаются от внешнего проводника.

В многомодовых кабелях (Multi-Mode Fiber, MMF) используют более широкие внутренние сердечники. Во внутреннем проводнике одновременно существует несколько световых лучей, которые отражаются от внешнего проводника под разными углами. Угол отражения луча называют модой.

Радиоканал характеризуется частотой радиоволн передаваемого сигнала. Существует множество систем передачи данных через радиоканал, использование которых зависит от требований к каналу. Существует несколько нелицензируемых диапазонов. В таких диапазонах регистрация устройств не требуется.

Наиболее популярные диапазоны: 433МГц, 868 МГц, 2.4 ГГц, 5 ГГц.

В диапазонах 433МГц, 868 МГц чаще всего работают различные технологические сети, с помощью которых констролируется работа различного удаленного технологического оборудования, сбора информации с датчиков и т. п.

Диапазон 2.4 ГГц, 5 ГГц используют различные стандарты, но наиболее популярными являются Wi-Fi, Bluetooth, ZigBee.

На более высоких частотах работают линии радиорелейной связи, которая используется в удаленных районах, в которых затруднена прокладка кабелей, в сетях мобильных операторов и т. п. РРЛ работает в условиях прямой видимости и на больших расстояниях.

Инфракрасный канал — канал передачи данных, не требующий для своего функционирования проводных соединений и использующий инфракрасное излучение.

В отличие от радиоканала, инфракрасный канал нечувствителен к электромагнитным помехам, и это позволяет использовать его в производственных условиях. К недостаткам инфракрасного канала относятся высокая стоимость приемников и передатчиков, где требуется преобразование электрического сигнала в инфракрасный и обратно, а также низкие скорости передачи (обычно не превышает 5-10 Мбит/с, но при использовании инфракрасных лазеров возможны существенно более высокие скорости). В условиях прямой видимости инфракрасный канал может обеспечить связь на расстояниях в несколько километров, но наиболее удобен он для связи компьютеров, находящихся в одном помещении, где отражения от стен комнаты дает устойчивую и надежную связь. Наиболее естественный тип топологии здесь — «шина» (то есть переданный сигнал одновременно получают все абоненты).

Структурированные кабельные системы. Стандарты СКС. Подсистемы СКС

Структурированная кабельная система (СКС) представляет собой иерархическую кабельную систему здания или группы зданий, разделенную на структурные подсистемы.

СКС состоит из набора медных и оптических кабелей, кросс-панелей, соединительных шнуров, кабельных разъемов, модульных гнезд, информационных розеток и вспомогательного оборудования. Все перечисленные элементы объединяются в единую систему и эксплуатируются согласно определенным правилам. Структурированная кабельная система способна поддерживать широкий диапазон приложений и создается без предварительного знания тех приложений, которые будут использоваться

впоследствии. Все стандарты СКС можно разделить на три группы - проектирование, монтаж и администрирование.

Стандарты проектирования определяют среду передачи, параметры разъемов, линии и канала, в том числе предельно допустимые длины, способы подключения проводников (последовательность), топологию и функциональные элементы СКС. Приложения дополняют стандарты в смежных областях и подразделяются на нормативные (часть стандарта) и информационные (для сведения). К этой группе можно отнести также документы, определяющие параметры заземления, особенности СКС малых офисов и жилых зданий, централизованных систем и рекомендации по построению открытых офисов. **Стандарты монтажа** определяют телекоммуникационные аспекты проектирования и строительства (комплекса) зданий. **Стандарты администрирования** определяют правила документирования телекоммуникационной инфраструктуры и создаются на базе стандартов проектирования и монтажа.

Основными стандартами по СКС являются:

- **международный стандарт** ISO/IEC 11801 Generic Cabling for Customer Premises;
- **европейский стандарт** EN 50173 Information technology– Generic cabling systems;
- **американский стандарт** ANSI/TIA/EIA 568-B Commercial Building Telecommunication Cabling Standard.

По назначению структурированную сеть принято разделять на подсистемы. Согласно международным стандартам, выделяют три подсистемы: магистраль комплекса, магистраль здания и горизонтальную подсистему.

Магистраль комплекса служит для соединения различных зданий. Как правило, она реализуется на оптоволоконном (реже медном) кабеле и позволяет соединять между собой здания, находящиеся на расстоянии до нескольких километров.

Магистраль здания соединяет этажи здания, обеспечивает связь между распределительной панелью здания и панелями этажей. Она должна включать кабель, установленный вертикально между этажными панелями, главную или промежуточную панель в многоэтажном здании, а также кабель, установленный горизонтально между панелями в длинном одноэтажном здании.

Горизонтальная подсистема прокладывается между телекоммуникационной розеткой на рабочем месте и этажной распределительной панелью. Каждый этаж здания рекомендуется обслуживать собственной горизонтальной подсистемой. На каждое рабочее место должно быть проложено как минимум два горизонтальных кабеля.

По стандарту ANSI/TIA/EIA-568-A выделяют 6 подсистем.

- **Entrance facility (устройства ввода).** К этой подсистеме относятся все кабели, соединительное оборудование, защитные и другие устройства,

используемые для подключения к другим зданиям и/или внешним сетям. К примеру, эта подсистема служит точкой входа для кабеля внешней телефонной сети.

•**Equipment room (аппаратная).** Аппаратная определяется как место расположения основного (main cross-connect) или промежуточного (intermediate cross-connect) кросса, к которым подключаются кабели вертикальной подсистемы. Здесь также может располагаться различное телекоммуникационное оборудование (учрежденческие АТС, центральное компьютерное и сетевое оборудование). **Backbone cabling (вертикальная подсистема).**

Вертикальная подсистема обеспечивает связь между отсеками связи (telecommunications closet), аппаратными комнатами и входными узлами. К ней относятся вертикальные кабели, главный и промежуточный кроссы. Эта подсистема может соединять отсеки связи как внутри здания, так и между ними.

•**Telecommunications closet (отсек связи).** Отсек связи - это место подключения кабеля горизонтальной подсистемы, идущего от подсистемы рабочего места. В отсеке связи также выполняется подключение и кроссирование вертикального кабеля. Кроссирование выполняется на панелях переключения (patch panel) с помощью шнуров переключения (patch cord).

•**Horizontal cabling (горизонтальная подсистема).** Горизонтальная подсистема включает кабели, соединяющие отсеки связи с информационными розетками на одном этаже.

•**Work-area components (подсистема рабочего места).** К этой подсистеме относятся компоненты, соединяющие оконечное оборудование с информационными розетками.

1.3. МЕТОДЫ ДОСТУПА К РЕСУРСАМ ЛВС

1.3.1. Методы доступа в типовых архитектурах ЛВС

Вопросы для изучения:

- Технология Ethernet;
- Технология Token Ring;
- Технология ARCNET.

Метод доступа определяет алгоритм, согласно которому узлы сети получают доступ к среде передачи данных, и осуществляют мультиплексирование/ демультиплексирование данных.

Технология Ethernet

Термин **Ethernet** относится к семейству протоколов локальных сетей, которые описываются стандартом IEEE 802.3 и используют метод доступа к среде CSMA/CD.

CSMA/CD (Carrier Sense Multiple Access with Collision Detection — множественный доступ с прослушиванием несущей и обнаружением коллизий) — технология (IEEE 802.3) множественного доступа к общей передающей среде в локальной компьютерной сети с контролем коллизий. Он используется как в обычных сетях типа Ethernet, так и в высокоскоростных сетях (Fast Ethernet, Gigabit Ethernet).

Если во время передачи кадра рабочая станция обнаруживает другой сигнал, занимающий передающую среду, она останавливает передачу, посылает сигнал преднамеренной помехи и ждёт в течение случайного промежутка времени (известного как «backoff delay» и находимого с помощью экспоненциального двоичного алгоритма выдержки), перед тем как снова отправить кадр.

Обнаружение коллизий используется для улучшения производительности CSMA с помощью прерывания передачи сразу после обнаружения коллизии и снижения вероятности второй коллизии во время повторной передачи.

В настоящий момент существует три основные разновидности технологии, которые функционируют на базе оптоволоконных кабелей или неэкранированной витой пары:

1. 10 Mbps — 10Base-T Ethernet;
2. 100 Mbps — Fast Ethernet;
3. 1000 Mbps — Gigabit Ethernet.

В серверных системах применяется также 10G Ethernet.

10 – мегабитный Ethernet включает три стандарта физического уровня.

10Base – 5 («Толстый» коаксиал) – использует в качестве передающей среды коаксиальный кабель диаметром 0.5 дюйма, волновое сопротивление 50 Ом. Максимальная длина сегмента без повторителей – 500м. На один сегмент может подключаться не более 100 трансиверов. При построении сети используется правило «3-4- 5» (3 «нагруженных» сегмента, 4 повторителя, не более 5 сегментов). Повторитель подключается при помощи трансивера, т.о. в сети может быть не более 297 узлов. Для того чтобы предотвратить появление отраженных сигналов, используются терминаторы сопротивлением 50 Ом.

10 Base – 2 («Тонкий» коаксиал) – использует в качестве передающей среды коаксиальный кабель диаметром 0.25 дюйма, волновое сопротивление 50 Ом. Максимальная длина сегмента без повторителей – 185м. На один сегмент может подключаться не более 30 узлов. При построении сети используется правило «3-4-5» (3 «нагруженных» сегмента, 4 повторителя, не более 5 сегментов). Для того чтобы предотвратить появление отраженных сигналов, используются терминаторы сопротивлением 50 Ом.

10 Base – T (Неэкранированная витая пара) – в качестве передающей среды используются две неэкранированные витые пары, узлы подключаются к

концентратору и образуют топологию «звезда». Расстояние от повторителя до станции не более 100 метров для категории кабеля не ниже 3. Концентраторы могут соединяться между собой, увеличивая протяженность логического сегмента сети (домена коллизий). При построении сети используется правило 4-х хабов (между любыми двумя узлами сети должно быть не более 4-х повторителей), количество узлов в сети не должно превышать 1024.

100 – мегабитный Ethernet (Fast Ethernet) включает следующие спецификации.

100 Base – TX. Среда передачи данных - неэкранированная витая пара категории не ниже 5. Поддерживается функция автоопределения скорости. Возможна работа в полнодуплексном режиме.

100 Base – FX использует многомодовое оптоволокно.

100 Base – T4 использует 4 витые пары для передачи данных по кабелю 3 категории. Не поддерживает полнодуплексной передачи данных.

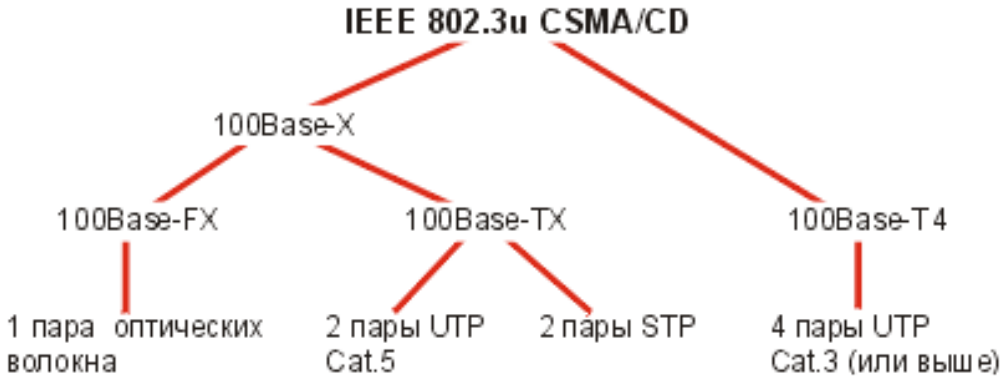


Рисунок 3.1 – Стандарт Ethernet

В сетях 100-мегабитного Ethernet используются повторители двух классов (**I** и **II**). Повторители класса **I** могут соединять каналы, отвечающие разным требованиям, например, 100Base-TX и 100Base-T4 или 100Base-FX. В пределах одного логического сегмента может быть применен только один повторитель класса **I**. Такие повторители часто имеют встроенные возможности управления с использованием протокола SNMP.

Повторители класса **II** не выполняют преобразования сигналов, и могут объединять только однотипные сегменты. Логический сегмент может содержать не более двух повторителей класса **II**.

При построении сети необходимо учитывать следующие ограничения:

Все сегменты на витой паре не должны превышать 100 м. Оптоволоконные сегменты не должны превышать 412 м. Расстояние между концентраторами класса **II** не должно превышать 5м.

1000 – мегабитный (Gigabit) Ethernet описан следующими стандартами:

1. IEEE 802.3z (1000Base-TX, 1000Base-LX, 1000Base-SX);
2. IEEE 802.3ab (1000Base-T).

•**1000Base-TX***:* передающая среда – экранированный медный кабель длиной до 25м;

•**1000Base-LX***:* передающая среда – одномодовое оптоволокно, длина до 5000м;

•**1000Base-CX***:* передающая среда – многомодовое оптоволокно, длина до 550м;

•**1000Base-T***:* передающая среда – UTP CAT5/CAT5e, длина сегмента до 100м.

При проектировании сетей Ethernet должно всегда выполняться требование корректного определения коллизий. Для этого время передачи кадра минимальной длины должно превышать или быть равным размеру интервала времени, за который кадр дважды пройдет расстояние между двумя самыми удаленными узлами сети.

Технология Token Ring

Сеть Token-Ring (маркерное кольцо) была предложена компанией IBM в 1985 году (первый вариант появился в 1980 году). Она предназначалась для объединения в сеть всех типов компьютеров, выпускаемых IBM. Token-Ring является в настоящее время международным стандартом IEEE 802.5 (хотя между Token-Ring и IEEE 802.5 есть незначительные отличия). Разрабатывалась Token-Ring как надежная альтернатива Ethernet, но сейчас Ethernet вытеснил все остальные сети, и Token-Ring можно считать безнадежно устаревшей. Уже в 1999 году большинством производителей оборудования было прекращено производство новых устройств для сетей Token-Ring.

Была разработана фирмой IBM в 1984 году. Топология сети Token Ring представляет собой кольцо, где все станции соединены отрезками кабеля. Способ доступа к сети – маркерный. Право передавать данные получает та станция, которая завладела маркером – кадром специального формата. Период времени, в течение которого станция может вести передачу определяется временем удержания маркера.

Данные передаются с двумя скоростями – **4 и 16 Мбит/с**. Работа на разных скоростях в одном кольце не допускается. Для контроля состояния сети одна из станций при инициализации кольца выбирается на роль активного монитора.

В сети Token Ring со скоростью передачи 4 Мбит станция передает кадр данных, который по кругу передается всеми станциями, пока его не получит станция – адресат. Станция – получатель копирует кадр в свой буфер, устанавливает признак того, что кадр был успешно принят, и передает его по кольцу дальше. Станция – отправитель кадра изымает кадр из сети, и, если время удержания маркера не истекло, то передает следующий кадр данных. В один момент времени в сети присутствует либо маркер, либо кадр данных.

В сети Token Ring со скоростью передачи 16 Мбит используется алгоритм раннего высвобождения маркера. Его суть заключается в том, что станция, передавшая кадр своих данных, передает следом кадр маркера, не дожидаясь возвращения кадра данных по кольцу. В этом случае по кольцу одновременно

циркулируют кадры данных и маркера, но данные может передавать только станция, захватившая маркер.

Для разных типов сообщений, кадрам могут присваиваться различные приоритеты- от 0 до 7. Кадр маркера имеет два поля в которых записываются текущее и резервируемое значения приоритета. Станция может захватить маркер только в том случае, если значение приоритета для ее данных выше или равно значению приоритета маркера. В противном случае она может записать значение приоритета своих данных в резервное поле приоритета маркера, зарезервировав его для себя во время следующего прохода (если это поле еще не зарезервировано для данных с более высоким уровнем приоритета). Станция, которая сумела захватить маркер, после завершения передачи своих данных переписывает биты поля резервного приоритета в поле приоритета маркера и обнуляет поле резервного приоритета. Механизм приоритетов используется только по требованию приложений. На физическом уровне узлы в сети Token Ring подключаются при помощи устройств многостанционного доступа (MSAU – Multistation Access Unit), которые объединяются кусками кабеля и образуют кольцо. Все станции в кольце работают на одной скорости. Максимальная длина кольца равна 4000м.

Технология ARCNET

Сеть Arcnet (или ARCnet от английского Attached Resource Computer Net, компьютерная сеть соединенных ресурсов) – это одна из старейших сетей. Она была разработана компанией Datapoint Corporation еще в 1977 году. Международные стандарты на эту сеть отсутствуют, хотя именно она считается родоначальницей метода маркерного доступа. Несмотря на отсутствие стандартов, сеть Arcnet до недавнего времени (в 1980 – 1990 г.г.) пользовалась популярностью, даже серьезно конкурировала с Ethernet. Большое количество компаний (например, Datapoint, Standard Microsystems, Xircom и др.) производили аппаратуру для сети этого типа. Но сейчас производство аппаратуры Arcnet практически прекращено. После распространения Ethernet в качестве технологии для создания ЛВС, ARCNET нашла применение во встраиваемых системах. Среди основных достоинств сети Arcnet по сравнению с Ethernet можно назвать ограниченную величину времени доступа, высокую надежность связи, простоту диагностики, а также сравнительно низкую стоимость адаптеров. К наиболее существенным недостаткам сети относятся низкая скорость передачи информации (2,5 Мбит/с), система адресации и формат пакета. В качестве среды передачи в сети используется коаксиальный кабель с волновым сопротивлением 93 Ом, к примеру, марки RG-62A/U. Варианты с витой парой (экранированной и неэкранированной) не получили широкого распространения. Были предложены и варианты на оптоволоконном кабеле, но и они также не спасли Arcnet. В качестве топологии сеть Arcnet использует классическую шину (Arcnet-BUS), а также пассивную звезду (Arcnet-STAR). В звезде применяются концентраторы (хабы). Возможно объединение с помощью концентраторов шинных и звездных сегментов в древовидную топологию (как и в Ethernet).

Главное ограничение – в топологии не должно быть замкнутых путей (петель). Еще одно ограничение: количество сегментов, соединенных последовательной цепочкой с помощью концентраторов, не должно превышать трех. Концентраторы бывают двух видов:

- Активные концентраторы (восстанавливают форму приходящих сигналов и усиливают их). Количество портов – от 4 до 64. Активные концентраторы могут соединяться между собой (каскадироваться). Шинные сегменты могут подключаться только к активным концентраторам.
- Пассивные концентраторы (просто смешивают приходящие сигналы без усиления). Количество портов – 4. Пассивные концентраторы не могут соединяться между собой. Они могут связывать только активные концентраторы и/или сетевые адаптеры.

Сетевые адаптеры также бывают двух видов:

- Высокоимпедансные (Bus), предназначенные для использования в шинных сегментах.
- Низкоимпедансные (Star), предназначенные для использования в пассивной звезде.

Низкоимпедансные адаптеры отличаются от высокоимпедансных тем, что они содержат в своем составе согласующие 93-омные терминаторы. При их применении внешнее согласование не требуется. В шинных сегментах низкоимпедансные адаптеры могут использоваться как оконечные для согласования шины. Высоимпедансные адаптеры требуют применения внешних 93-омных терминаторов. Некоторые сетевые адаптеры имеют возможность переключения из высокоимпедансного состояния в низкоимпедансное, они могут работать и в шине, и в звезде. Основные технические характеристики сети Arcnet следующие.

- Среда передачи – коаксиальный кабель, витая пара.
- Максимальная длина сети – 6 километров.
- Максимальная длина кабеля от абонента до пассивного концентратора – 30 метров.
- Максимальная длина кабеля от абонента до активного концентратора – 600 метров.
- Максимальная длина кабеля между активным и пассивным концентраторами – 30 метров.
- Максимальная длина кабеля между активными концентраторами – 600 метров.
- Максимальное количество абонентов в сети – 255.
- Максимальное количество абонентов на шинном сегменте – 8.
- Минимальное расстояние между абонентами в шине – 1 метр.
- Максимальная длина шинного сегмента – 300 метров.

- Скорость передачи данных – 2,5 Мбит/с.

В сети Arcnet используется маркерный метод доступа (метод передачи права), но он несколько отличается от аналогичного в сети Token-Ring. Ближе всего этот метод к тому, который предусмотрен в стандарте IEEE 802.4. Последовательность действий абонентов при данном методе:

1. Абонент, желающий передавать, ждет прихода маркера.
2. Получив маркер, он посылает запрос на передачу абоненту-приемнику информации (спрашивает, готов ли приемник принять его пакет).
3. Приемник, получив запрос, посылает ответ (подтверждает свою готовность).
4. Получив подтверждение готовности, абонент-передатчик посылает свой пакет.
5. Получив пакет, приемник посылает подтверждение приема пакета.
6. Передатчик, получив подтверждение приема пакета, заканчивает свой сеанс связи. После этого маркер передается следующему абоненту по порядку убывания сетевых адресов.

Таким образом, в данном случае пакет передается только тогда, когда есть уверенность в готовности приемника принять его. Это существенно увеличивает надежность передачи. Так же, как и в случае Token-Ring, конфликты в Arcnet полностью исключены. Как и любая маркерная сеть, Arcnet хорошо держит нагрузку и гарантирует величину времени доступа к сети (в отличие от Ethernet). Размер пакета сети Arcnet составляет 0,5 Кбайта. Помимо поля данных в него входят также 8-битные адреса приемника и передатчика и 16-битная циклическая контрольная сумма (CRC). Такой небольшой размер пакета оказывается не слишком удобным при высокой интенсивности обмена по сети. Адаптеры сети Arcnet отличаются от адаптеров других сетей тем, что в них необходимо с помощью переключателей или перемычек установить собственный сетевой адрес (всего их может быть 255, так как последний, 256-ой адрес применяется в сети для режима широкого вещания). Контроль уникальности каждого адреса сети полностью возлагается на пользователей сети. Подключение новых абонентов становится при этом довольно сложным, так как необходимо задавать тот адрес, который еще не использовался. Выбор 8-битного формата адреса ограничивает допустимое количество абонентов в сети – 255, что недостаточно для крупных компаний. В результате все это привело к практически полному отказу от сети Arcnet. Существовали варианты сети Arcnet, рассчитанные на скорость передачи 20 Мбит/с, но они не получили широкого распространения.

1.3.2. Логическое и физическое структурирование сетей.

Вопросы для изучения:

- Сетевое оборудование: интерфейсы, повторитель, концентратор, коммутатор, маршрутизатор.

Сетевое оборудование — устройства, необходимые для работы компьютерной сети.

Сегмент сети — логически или физически обособленная часть сети. Разбиение сети на сегменты осуществляется с целью оптимизации сетевого трафика и/или повышения безопасности сети в целом.

Сетевая плата (**сетевая карта**, **сетевой адаптер)** — устройство, позволяющее компьютеру взаимодействовать с другими устройствами сети. В настоящее время в персональных компьютерах и ноутбуках контроллер и компоненты, выполняющие функции сетевой платы, чаще всего интегрированы в материнские платы для удобства, в том числе унификации драйвера и удешевления всего компьютера в целом.

Сетевой адаптер (Network Interface Card (или Controller), NIC) вместе со своим драйвером реализует второй, канальный уровень модели открытых систем (OSI) в конечном узле сети — компьютере. Более точно, в сетевой операционной системе пара адаптер и драйвер выполняет только функции физического и MAC-уровней, в то время как LLC-уровень обычно реализуется модулем операционной системы, единым для всех драйверов и сетевых адаптеров. Собственно так оно и должно быть в соответствии с моделью стека протоколов IEEE 802.

Сетевой адаптер совместно с драйвером выполняют две операции: передачу и прием кадра. Передача кадра из компьютера в кабель состоит из перечисленных ниже этапов (некоторые могут отсутствовать, в зависимости от принятых методов кодирования):

- Прием кадра данных LLC через межуровневый интерфейс вместе с адресной информацией MAC-уровня. Обычно взаимодействие между протоколами внутри компьютера происходит через буферы, расположенные в оперативной памяти. Данные для передачи в сеть помещаются в эти буферы протоколами верхних уровней, которые извлекают их из дисковой памяти либо из файлового кэша с помощью подсистемы ввода-вывода операционной системы.
- Оформление кадра данных MAC-уровня, в который инкапсулируется кадр LLC (с отброшенными флагами 01111110). Заполнение адресов назначения и источника, вычисление контрольной суммы.
- Формирование символов кодов при использовании избыточных кодов типа 4B/5B. Скремблирование кодов для получения более равномерного спектра сигналов. Этот этап используется не во всех протоколах — например, технология Ethernet 10 Мбит/с обходится без него.

- Выдача сигналов в кабель в соответствии с принятым линейным кодом — манчестерским, NRZI, MLT-3 и т. п.

Прием кадра из кабеля в компьютер включает следующие действия:

- Прием из кабеля сигналов, кодирующих битовый поток.
- Выделение сигналов на фоне шума. Эту операцию могут выполнять различные специализированные микросхемы или сигнальные процессоры DSP. В результате в приемнике адаптера образуется некоторая битовая последовательность, с большой степенью вероятности совпадающая с той, которая была послана передатчиком.
- Если данные перед отправкой в кабель подвергались скремблированию, то они пропускаются через дескремблер, после чего в адаптере восстанавливаются символы кода, посланные передатчиком.
- Проверка контрольной суммы кадра. Если она неверна, то кадр отбрасывается, а через межуровневый интерфейс наверх, протоколу LLC передается соответствующий код ошибки. Если контрольная сумма верна, то из MAC-кадра извлекается кадр LLC и передается через межуровневый интерфейс наверх, протоколу LLC. Кадр LLC помещается в буфер оперативной памяти.

Распределение обязанностей между сетевым адаптером и его драйвером стандартами не определяется, поэтому каждый производитель решает этот вопрос самостоятельно. Обычно сетевые адаптеры делятся на адаптеры для клиентских компьютеров и адаптеры для серверов.

В адаптерах для клиентских компьютеров значительная часть работы перекладывается на драйвер, тем самым адаптер оказывается проще и дешевле. Недостатком такого подхода является высокая степень загрузки центрального процессора компьютера рутинными работами по передаче кадров из оперативной памяти компьютера в сеть. Центральный процессор вынужден заниматься этой работой вместо выполнения прикладных задач пользователя.

Поэтому адаптеры, предназначенные для серверов, обычно снабжаются собственными процессорами, которые самостоятельно выполняют большую часть работы по передаче кадров из оперативной памяти в сеть и в обратном направлении.

В зависимости от того, какой протокол реализует адаптер, адаптеры делятся на Ethernet-адаптеры, Token Ring-адаптеры, FDDI-адаптеры и т. д. Так как протокол Fast Ethernet позволяет за счет процедуры автопереговоров автоматически выбрать скорость работы сетевого адаптера в зависимости от возможностей концентратора, то многие адаптеры Ethernet сегодня поддерживают две скорости работы и имеют в своем названии приставку 10/100/1000. Это свойство некоторые производители называют авточувствительностью.

По конструктивной реализации сетевые платы делятся на:

- внутренние — отдельные платы, вставляющиеся в PCI или PCI-E слот;
- внешние, подключающиеся через USB или PCMCIA интерфейс, преимущественно использующиеся в ноутбуках;
- встроенные в материнскую плату.

Повторитель (репитер, от англ. *Repeater*) — сетевое оборудование, предназначенное для увеличения расстояния сетевого соединения и его расширения за пределы одного сегмента или для организации двух ветвей, путём повторения электрического сигнала «один в один». В терминах модели OSI работает на физическом уровне.

Одной из первых задач, которая стоит перед любой технологией транспортировки данных, является возможность их передачи на максимально большое расстояние. Физическая среда накладывает на этот процесс своё ограничение — рано или поздно мощность сигнала падает, и приём становится невозможным. Но ещё большее значение имеет то, что искажается «форма сигнала» — закономерность, в соответствии с которой мгновенное значение уровня сигнала изменяется во времени. Это происходит в результате того, что провода, по которым передаётся сигнал, имеют собственную ёмкость и индуктивность. Электрические и магнитные поля одного проводника наводят ЭДС в других проводниках (длинная линия).

Привычное для аналоговых систем усиление не годится для высокочастотных цифровых сигналов. Разумеется, при его использовании какой-то небольшой эффект может быть достигнут, но с увеличением расстояния искажения быстро нарушат целостность данных.

Проблема не нова, и в таких ситуациях применяют не усиление, а повторение сигнала. При этом устройство на входе должно принимать сигнал, далее распознавать его первоначальный вид, и генерировать на выходе его точную копию. Такая схема в теории может передавать данные на сколь угодно большие расстояния (если не учитывать особенности разделения физической среды в Ethernet).

Сетевой концентратор, также **хаб** — устройство для объединения компьютеров в сеть Ethernet с применением кабельной инфраструктуры типа *витая пара*. В настоящее время вытеснены сетевыми коммутаторами.

Сетевые концентраторы также могли иметь разъёмы для подключения к существующим сетям на базе толстого или тонкого коаксиального кабеля.

Концентратор работает на первом (физическом) уровне сетевой модели OSI, ретранслируя входящий сигнал с одного из портов в сигнал на все остальные (подключённые) порты, реализуя, таким образом, свойственную Ethernet топологию *общая шина*, с разделением пропускной способности сети между всеми устройствами и работой в режиме полудуплекса. Коллизии (то есть попытка двух и более устройств начать передачу одновременно) обрабатываются аналогично сети Ethernet на других носителях — устройства

самостоятельно прекращают передачу и возобновляют попытку через случайный промежуток времени, говоря современным языком, концентратор объединяет устройства в одном домене коллизий.

Сетевой концентратор также обеспечивает бесперебойную работу сети при отключении устройства от одного из портов или повреждении кабеля, в отличие, например, от сети на коаксиальном кабеле, которая в таком случае прекращает работу целиком.

Единственное преимущество концентратора — низкая стоимость — было актуально лишь в первые годы развития сетей Ethernet. По мере совершенствования и удешевления электронных микропроцессорных компонентов данное преимущество концентратора полностью сошло на нет, так как стоимость вычислительной части коммутаторов и маршрутизаторов составляет лишь малую долю на фоне стоимости разъёмов, разделительных трансформаторов, корпуса и блока питания, общих для концентратора и коммутатора.

Недостатки концентратора являются логическим продолжением недостатков топологии общая шина, а именно — снижение пропускной способности сети по мере увеличения числа узлов. Кроме того, поскольку на физическом уровне узлы не изолированы друг от друга, все они будут работать со скоростью передачи данных самого худшего узла. Например, если в сети присутствуют узлы со скоростью 100 Мбит/с и всего один узел со скоростью 10 Мбит/с, то все узлы будут работать на скорости 10 Мбит/с, даже если узел 10 Мбит/с вообще не проявляет никакой информационной активности.

Ещё одним важным недостатком является то, что вещание сетевого трафика во все порты, создает существенную проблему с точки зрения информациооной безопастности, т. к. любой участник может прослушивать всю информацию в сети, что создает угрозу утечек важных данных (логины и пароли, cookie и т.п.).

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

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

от необходимости (и возможности) обрабатывать данные, которые им не предназначались.

Принцип работы коммутатора

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

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

Режимы коммутации:

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

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

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

Симметричная и асимметричная коммутация

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

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

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

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

Буфер памяти

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

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

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

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

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

Возможности и разновидности коммутаторов

Коммутаторы подразделяются на управляемые и неуправляемые (наиболее простые).

Более сложные коммутаторы позволяют управлять коммутацией на сетевом (третьем) уровне модели OSI. Обычно их именуют соответственно, например «Layer 3 Switch» или сокращенно «L3 Switch». Управление коммутатором может осуществляться посредством Web-интерфейса, интерфейса командной строки (CLI), протокола SNMP, RMON и т. п.

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

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

Маршрутизатор (**роутер**) — устройство, которое пересылает пакеты между различными сегментами сети на основе правил и таблиц маршрутизации. Маршрутизатор может связывать разнородные сети различных архитектур. Для принятия решений о пересылке пакетов используется информация о топологии сети и определённые правила, заданные администратором.

Маршрутизаторы работают на «сетевом» (третьем) уровне сетевой модели OSI, нежели свитч и концентратор (хаб), которые работают соответственно на втором и первом уровнях модели OSI.

Принцип работы:

Обычно маршрутизатор использует адрес получателя, указанный в заголовке пакета, и определяет по таблице маршрутизации путь, по которому следует передать данные. Если в таблице маршрутизации для адреса нет описанного маршрута, пакет отбрасывается.

Существуют и другие способы определения маршрута пересылки пакетов, когда, например, используется адрес отправителя, используемые протоколы верхних уровней и другая информация, содержащаяся в

заголовках пакетов сетевого уровня. Нередко маршрутизаторы могут осуществлять трансляцию адресов отправителя и получателя, фильтрацию транзитного потока данных на основе определённых правил с целью ограничения доступа, шифрование/расшифрование передаваемых данных и т. д.

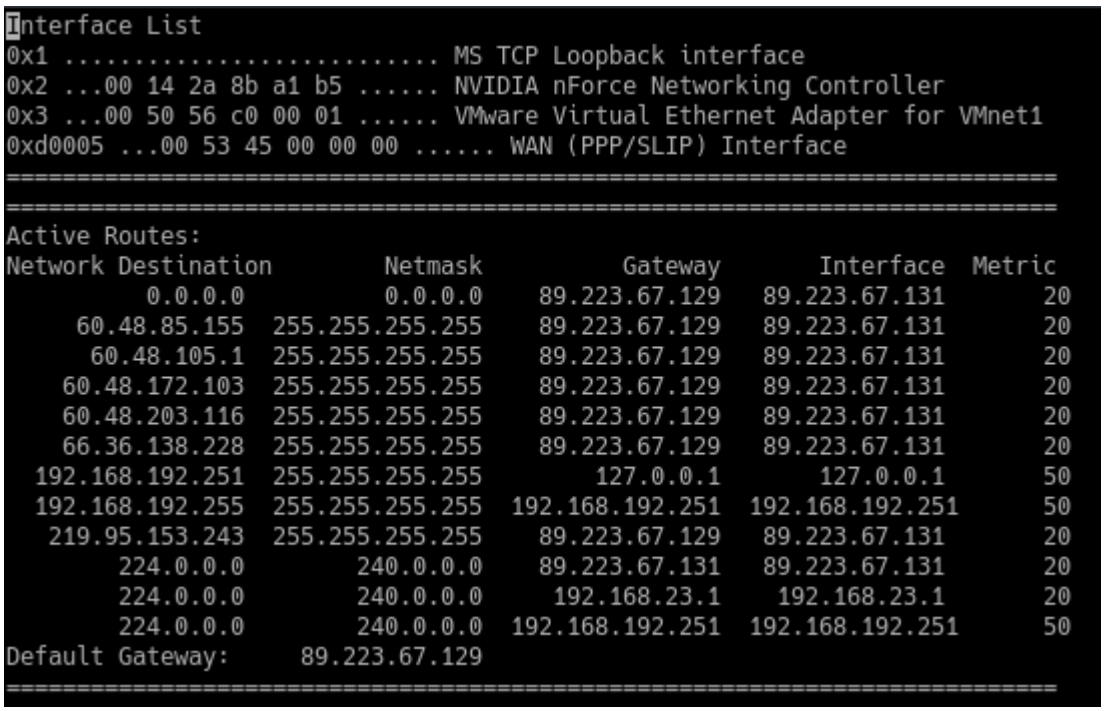
Таблица маршрутизации

Таблица маршрутизации содержит информацию, на основе которой маршрутизатор принимает решение о дальнейшей пересылке пакетов. Таблица состоит из некоторого числа записей — маршрутов, в каждой из которых содержится идентификатор сети получателя (состоящий из адреса и маски сети), адрес следующего узла, которому следует передавать пакеты, административное расстояние — степень доверия к источнику маршрута и некоторый вес записи — метрика. Метрики записей в таблице играют роль в вычислении кратчайших маршрутов к различным получателям. В зависимости от модели маршрутизатора и используемых протоколов маршрутизации, в таблице может содержаться некоторая дополнительная служебная информация. Пример таблицы маршрутизации показан на рис. 3.2.

Таблица маршрутизации может составляться двумя способами:

- с**татическая маршрутизация** — когда записи в таблице вводятся и изменяются вручную. Такой способ требует вмешательства администратора каждый раз, когда происходят изменения в топологии сети. С другой стороны, он является наиболее стабильным и требующим минимума аппаратных ресурсов маршрутизатора для обслуживания таблицы.

- д**инамическая маршрутизация** — когда записи в таблице обновляются автоматически при помощи одного или нескольких протоколов маршрутизации — RIP, OSPF, BGP, и др. Кроме того, маршрутизатор строит таблицу оптимальных путей к сетям назначения на основе различных критериев — количества промежуточных узлов, пропускной способности каналов, задержки передачи данных и т. п. Критерии вычисления оптимальных маршрутов чаще всего зависят от протокола маршрутизации, а также задаются конфигурацией маршрутизатора. Такой способ построения таблицы позволяет автоматически держать таблицу маршрутизации в актуальном состоянии и вычислять оптимальные маршруты на основе текущей топологии сети. Однако динамическая маршрутизация оказывает дополнительную нагрузку на устройства, а высокая нестабильность сети может приводить к ситуациям, когда маршрутизаторы не успевают синхронизировать свои таблицы, что приводит к противоречивым сведениям о топологии сети в различных её частях и потере передаваемых данных.



```
Interface List
0x1 ........................... MS TCP Loopback interface
0x2 ...00 14 2a 8b a1 b5 ...... NVIDIA nForce Networking Controller
0x3 ...00 50 56 c0 00 01 ...... VMware Virtual Ethernet Adapter for VMnet1
0xd0005 ...00 53 45 00 00 00 ...... WAN (PPP/SLIP) Interface
===========================================================================
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0    89.223.67.129   89.223.67.131       20
     60.48.85.155  255.255.255.255    89.223.67.129   89.223.67.131       20
      60.48.105.1  255.255.255.255    89.223.67.129   89.223.67.131       20
    60.48.172.103  255.255.255.255    89.223.67.129   89.223.67.131       20
    60.48.203.116  255.255.255.255    89.223.67.129   89.223.67.131       20
    66.36.138.228  255.255.255.255    89.223.67.129   89.223.67.131       20
  192.168.192.251  255.255.255.255        127.0.0.1       127.0.0.1       50
  192.168.192.255  255.255.255.255  192.168.192.251  192.168.192.251      50
   219.95.153.243  255.255.255.255    89.223.67.129   89.223.67.131       20
        224.0.0.0        240.0.0.0    89.223.67.131   89.223.67.131       20
        224.0.0.0        240.0.0.0     192.168.23.1    192.168.23.1       20
        224.0.0.0        240.0.0.0  192.168.192.251  192.168.192.251      50
Default Gateway:     89.223.67.129
===========================================================================
```

Рисунок 3.2 - Пример таблицы маршрутизации

Применение

Маршрутизаторы помогают уменьшить загрузку сети, благодаря её разделению на домены коллизий или широковещательные домены, а также благодаря фильтрации пакетов. В основном их применяют для объединения сетей разных типов, зачастую несовместимых по архитектуре и протоколам, например для объединения локальных сетей Ethernet и WAN-соединений, использующих протоколы xDSL, PPP, ATM, Frame relay и т. д. Нередко маршрутизатор используется для обеспечения доступа из локальной сети в глобальную сеть Интернет, осуществляя функции трансляции адресов и межсетевого экрана.

В качестве маршрутизатора может выступать как специализированное (аппаратное) устройство, так и обычный компьютер, выполняющий функции маршрутизатора.

1.3.3. Система адресации в ЛВС

Вопросы для изучения:

- Стек протоколов TCP/IP;
- Адресация в сетях TCP/IP;
- Протокол ARP;
- Система DNS;
- Протокол IP;
- Протокол DHCP;
- Промышленные компьютерные сети.

В этой главе рассмотрены протоколы самой популярной сетевой технологии TCP/IP, которая появилась уже почти 40 лет назад как результат создания Интернета, а сегодня используется практически во всех существующих и вновь создаваемых локальных и глобальных сетях.

Будут рассмотрены средства адресации узлов в сетях **TCP/IP**, включая протокол **ARP** и систему **DNS**, технологию **DHCP**, как на основе протокола IP происходит объединение нескольких сетей в единую сеть, как осуществляется маршрутизация и какую роль играет в работе в такой составной сети протокол **ICMP**.

Стек протоколов TCP/IP

Сегодня стек TCP/IP широко используется как в глобальных, так и локальных сетях. Этот стек имеет иерархическую структуру, в которой определено 4 уровня (рис. 3.3).

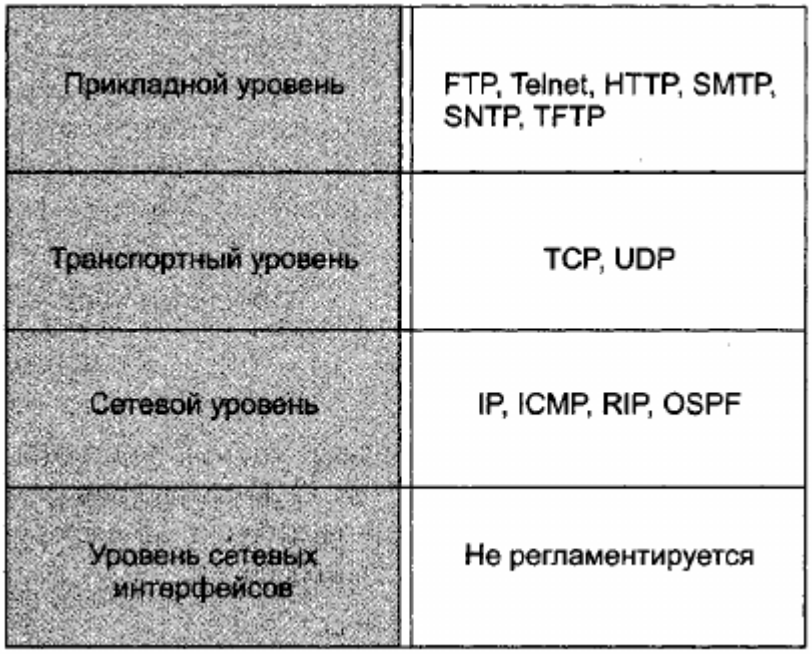


Рисунок 3.3 - Структура стека TCP/IP

Прикладной уровень стека TCP/IP соответствует трем верхним уровням модели OSI: прикладному, представления и сеансовому. Он объединяет службы, предоставляемые системой пользовательским приложениям. За долгие годы применения в сетях различных стран и организаций стек TCP/IP накопил большое количество протоколов и служб прикладного уровня. К ним относятся такие распространенные протоколы, как протокол передачи файлов (File Transfer Protocol, FTP), протокол эмуляции терминала telnet, простой протокол передачи электронной почты (Simple Mail Transfer Protocol, SMTP), протокол передачи гипертекста (Hypertext Transfer Protocol, HTTP) и многие другие. Протоколы прикладного уровня развертываются на хостах.

Транспортный уровень стека TCP/IP может предоставлять вышележащему уровню два типа сервиса:

- гарантированную доставку обеспечивает протокол управления передачей (Transmission Control Protocol, TCP);

- доставку по возможности, или с максимальными усилиями, обеспечивает протокол пользовательских дейтаграмм (User Datagram Protocol, UDP).

Для того чтобы обеспечить надежную доставку данных, протокол TCP предусматривает установление логического соединения, что позволяет ему нумеровать пакеты, подтверждать их прием квитанциями, в случае потери организовывать повторные передачи, распознавать и уничтожать дубликаты, доставлять прикладному уровню пакеты в том порядке, в котором они были отправлены. Благодаря этому протоколу, объекты на хосте-отправителе и хосте-получателе могут поддерживать обмен данными в дуплексном режиме. TCP дает возможность без ошибок доставить сформированный на одном из компьютеров поток байтов на любой другой компьютер, входящий в составную сеть.

Второй протокол этого уровня, UDP, является простейшим дейтаграммным протоколом, который используется в том случае, когда задача надежного обмена данными либо вообще не ставится, либо решается средствами более высокого уровня —прикладного уровня или пользовательскими приложениями.

В функции протоколов TCP и UDP входит также исполнение роли связующего звена между прилегающими к транспортному уровню прикладным и сетевым уровнями. От прикладного протокола транспортный уровень принимает задание на передачу данных с тем или иным качеством прикладному уровню-получателю. Нижележащий сетевой уровень протоколы TCP и UDP рассматривают как своего рода инструмент, не очень надежный, но способный перемещать пакет в свободном и рискованном путешествии по составной сети.

Программные модули, реализующие протоколы TCP и UDP, подобно модулям протоколов прикладного уровня, устанавливаются на хостах.

Сетевой уровень, называемый также уровнем Интернета, является стержнем всей архитектуры TCP/IP. Именно этот уровень, функции которого соответствуют сетевому уровню модели OSI, обеспечивает перемещение пакетов в пределах составной сети, образованной объединением нескольких подсетей. Протоколы сетевого уровня поддерживают интерфейс с вышележащим транспортным уровнем, получая от него запросы на передачу данных по составной сети, а также с нижележащим уровнем сетевых интерфейсов, о функциях которого мы расскажем далее.

Основным протоколом сетевого уровня является межсетевой протокол (Internet Protocol, IP). В его задачу входит продвижение пакета между сетями — от одного маршрутизатора к другому до тех пор, пока пакет не попадет в сеть назначения. В отличие от протоколов прикладного и транспортного уровней, протокол IP развертывается не только на хостах, но и на всех маршрутизаторах (шлюзах). Протокол IP — это дейтаграммный протокол,

работающий без установления соединений по принципу доставки с максимальными усилиями (best effort). Такой тип сетевого сервиса называют также «ненадежным».

К сетевому уровню TCP/IP часто относят протоколы, выполняющие вспомогательные функции по отношению к IP. Это, прежде всего, протоколы маршрутизации RIP и OSPF, предназначенные для изучения топологии сети, определения маршрутов и составления таблиц маршрутизации, на основании которых протокол IP перемещает пакеты в нужном направлении. По этой же причине к сетевому уровню могут быть отнесены протокол межсетевых управляющих сообщений (Internet Control Message Protocol, ICMP), предназначенный для передачи маршрутизатором источнику сведений об ошибках, возникших при передаче пакета, и некоторые другие протоколы.

Идеологическим отличием архитектуры стека TCP/IP от многоуровневой архитектуры других стеков является интерпретация функций самого нижнего уровня — **уровня сетевых интерфейсов.**

Напомним, что нижние уровни модели OSI (канальный и физический) реализуют множество функций доступа к среде передачи, формированию кадров, согласованию величин электрических сигналов, кодированию и синхронизации, а также некоторые другие. Все эти весьма конкретные функции составляют суть таких протоколов обмена данными, как Ethernet, PPP и многих других.

У нижнего уровня стека TCP/IP задача существенно проще — он отвечает только за организацию взаимодействия с технологиями сетей, входящих в составную сеть. TCP/IP рассматривает любую подсеть, входящую в составную сеть, как средство транспортировки пакетов между двумя соседними маршрутизаторами.

Задачу обеспечения интерфейса между технологией TCP/IP и любой другой технологией промежуточной сети упрощенно можно свести к определению способа:

- упаковки (инкапсуляции) IP -пакета в единицу передаваемых данных промежуточной сети;
- преобразования сетевых адресов в адреса технологии данной промежуточной сети.

Такой гибкий подход упрощает задачу расширения набора поддерживаемых технологий. При появлении новой популярной технологии она быстро включается в стек TCP/IP путем разработки соответствующего стандарта, определяющего метод инкапсуляции IP -пакетов в ее кадры (например, спецификация RFC 1577, определяющая работу протокола IP через сети ATM, появилась в 1994 году вскоре после принятия основных стандартов ATM). Так как для каждой вновь появляющейся технологии разрабатываются собственные интерфейсные средства, функции этого уровня нельзя определить раз и навсегда, и именно поэтому нижний уровень стека TCP/IP не регламентируется.

Адресация в сетях TCP/IP

Одним из важных достоинств технологии TCP/IP является гибкость и масштабируемость системы адресации, что позволяет ей достаточно просто включать в составную сеть сети разных технологий и разного масштаба.

Типы адресов стека TCP/IP

Для идентификации сетевых интерфейсов используются три типа адресов:

- локальные (аппаратные) адреса;
- сетевые адреса (IP-адреса);
- символьные (доменные) имена.

В разных сетевых технологиях в общем случае используются собственные системы адресации, которые предназначены исключительно для обеспечения связи собственных узлов. Однако как только некоторая сеть объединяется с другими сетями в составную сеть, функциональность этих адресов расширяется, они становятся необходимым элементом вышележащей объединяющей технологии — в данном случае технологии TCP/IP. Роль, которую играют эти адреса в TCP/IP, не зависит от того, какая именно технология применяется в подсети, поэтому они имеют общее название — **локальные (аппаратные) адреса**. Например, для сети, построенной по технологии Ethernet, локальными адресами сетевых интерфейсов этой сети для технологии TCP/IP будут, соответственно, **MAC-адреса.**

Чтобы технология TCP/IP могла решать свою задачу объединения сетей, ей необходима собственная глобальная система адресации, позволяющая универсальным и однозначным способом идентифицировать любой интерфейс составной сети. Очевидным решением является нумерация всех подсетей составной сети, а затем нумерация сетевых интерфейсов в пределах каждой из этих подсетей. Пара, состоящая из номера сети и номера узла, отвечает поставленным условиям и может служить в качестве **сетевого адреса**, или **IP-адреса**. Сетевой адрес представляет собой набор чисел, например, 192.45.66.17.

Числовое представление сетевого адреса достаточно эффективно для программных и аппаратных средств. Однако пользователи обычно предпочитают работать с более удобными **символьными (доменными) именами** компьютеров. Символьные имена в пределах составной сети строятся по иерархическому признаку. Примером доменного имени может служить имя base2.sales.zil.ru. Символьные имена называют также **DNS-именами**.

Между локальным адресом, доменным именем и IP -адресом, относящимся к одному и тому же сетевому интерфейсу, нет никакой функциональной зависимости. В общем случае сетевой интерфейс может иметь несколько локальных адресов, сетевых адресов, доменных имен.

Протокол ARP

Как уже было сказано, никакой функциональной зависимости между локальным адресом и его IP -адресом не существует, следовательно,

единственный способ установления соответствия — ведение таблиц. В результате конфигурирования сети каждый интерфейс «знает» свои IP -адрес и локальный адрес, что можно представить как таблицу, состоящую из одной строки. Проблема состоит в том, как организовать обмен имеющейся информацией между узлами сети.

Для определения локального адреса по IP-адресу используется протокол разрешения адресов (Address Resolution Protocol, ARP). Протокол разрешения адресов реализуется различным образом в зависимости от того, работает ли в данной сети протокол локальной сети с возможностью широковещания (Ethernet) или же какой-либо из протоколов глобальной сети (ATM, Frame Relay), которые не поддерживают широковещательный доступ.

Рассмотрим работу протокола ARP в локальных сетях с широковещанием.

Протокол ARP поддерживает на каждом интерфейсе сетевого адаптера или маршрутизатора ARP-таблицу, в которой в ходе функционирования сети накапливается информация о соответствии между IP -адресами и MAC-адресами других интерфейсов данной сети. Первоначально при включении компьютера или маршрутизатора в сеть все его ARP-таблицы пусты.

На рис. 3.4 показан фрагмент IP -сети, включающий две сети — Ethernet 1 и Ethernet 2, подключенные к интерфейсам 1 и 2 маршрутизатора соответственно.

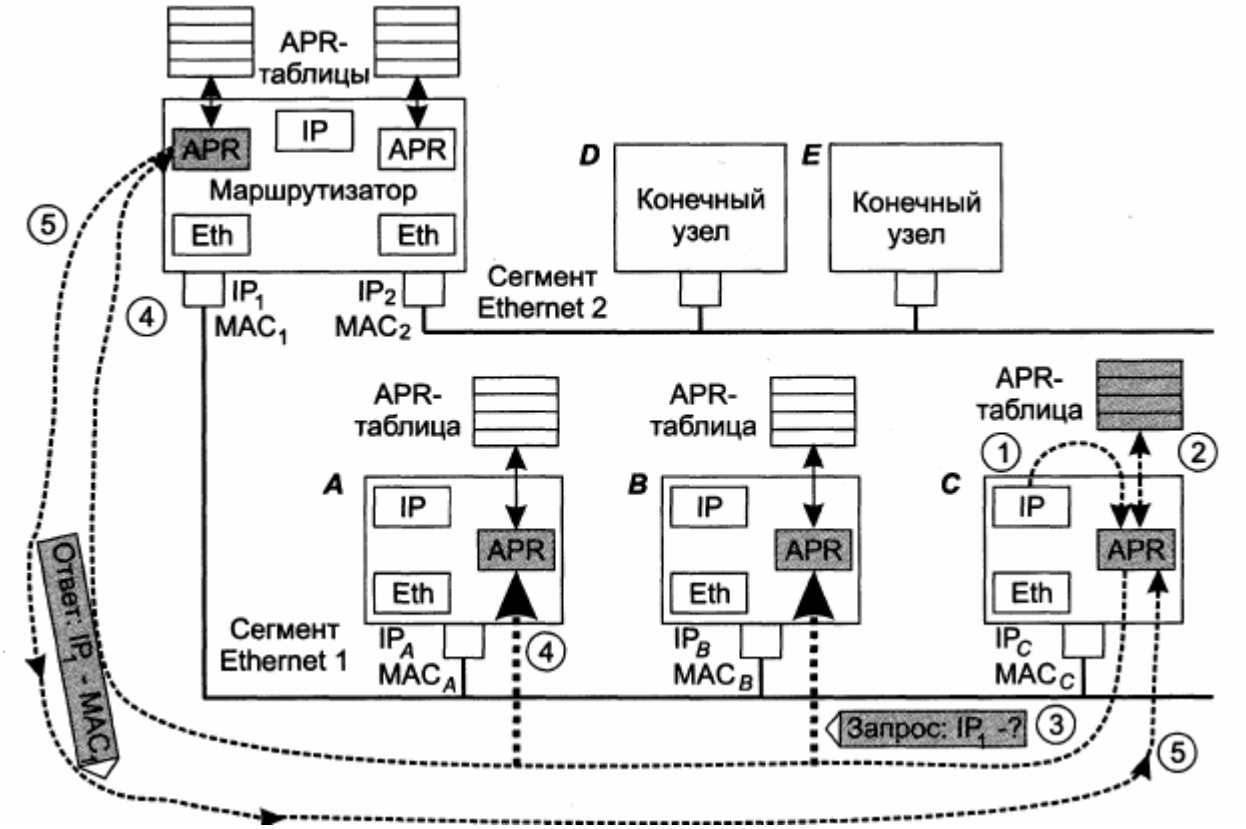


Рисунок 3.4 — Схема работы протокола ARP

Пусть в какой-то момент IP-модуль узла С направляет пакет узлу D. Протоколу IP узла С в результате конфигурирования стал известен IP-адрес интерфейса следующего маршрутизатора — это IP_1. Однако для того, чтобы

направить пакет маршрутизатору, необходимо определить его локальный адрес (MAC-адрес). Для решения этой задачи предпринимаются следующие шаги:

1. На первом шаге происходит передача от протокола IP протоколу ARP примерно такого сообщения: «Какой MAC-адрес имеет интерфейс с адресом IP_1?»

2. Работа протокола ARP начинается с просмотра собственной ARP-таблицы. Предположим, что среди содержащихся в ней записей отсутствует запрашиваемый IP-адрес.

3. В этом случае протокол ARP формирует ARP-запрос, вкладывает его в кадр протокола Ethernet и широковещательно рассылает. Заметим, что зона распространения ARP-запроса ограничивается сетью Ethernet 1, так как на пути широковещательных кадров барьером стоит маршрутизатор.

4. Все интерфейсы сети Ethernet 1 получают ARP-запрос и направляют его «своему» протоколу ARP. ARP сравнивает указанный в запросе адрес IP_1 с IP -адресом собственного интерфейса.

5. Протокол ARP, который констатировал совпадение (в данном случае это ARP интерфейса 1 маршрутизатора), формирует ARP-ответ. В ARP-ответе маршрутизатор указывает локальный адрес MAC_1 соответствующий адресу IP_1 своего интерфейса, и отправляет его запрашивающему узлу (в данном примере узлу C).

Чтобы уменьшить число ARP-обращений в сети, найденное соответствие между IP -адресом и MAC-адресом запоминается в ARP-таблице компьютера C (в данном случае это запись: IP_1 — MAC_1). Теперь, если вдруг вновь возникнет необходимость послать пакет по адресу IP_1 соответствующий локальный адрес будет быстро извлечен из ARP-таблицы.

ARP-таблица пополняется не только за счет поступающих на данный интерфейс ARP-ответов, но и в результате извлечения полезной информации из широковещательных ARP-запросов. Поскольку в каждом запросе содержатся IP- и MAC-адреса отправителя, все интерфейсы, получившие этот запрос, могут поместить информацию о соответствии локального и сетевого адресов отправителя в собственную ARP-таблицу. В нашем примере все узлы, получившие ARP-запрос от узла C, могут пополнить свои ARP-таблицы записью: IP_C— MAC_C.

В ARP-таблицах существует два типа записей: динамические и статические. Статические записи создаются вручную с помощью утилиты arp и не имеют срока устаревания, точнее, они существуют до тех пор, пока компьютер или маршрутизатор остается включенным. Динамические записи должны периодически обновляться. Если запись не обновлялась в течение определенного времени (порядка нескольких минут), то она исключается из таблицы. Таким образом, в ARP-таблице содержатся записи не обо всех узлах сети, а только о тех, которые активно участвуют в сетевых операциях. Поскольку такой способ хранения информации называют кэшированием, ARP-таблицы иногда называют **ARP-кэшем**.

Система DNS

Широковещательный механизм установления соответствия между символьными именами и локальными адресами, подобный протоколу ARP, хорошоработает только в небольшой локальной сети, не разделенной на подсети. В крупных сетях, где возможность всеобщей широковещательной рассылки не поддерживается, нужен другой инструмент разрешения символьных имен.

На раннем этапе развития Интернета на каждом хосте вручную создавался текстовый файл с известным именем hosts.txt. Этот файл состоял из некоторого количества строк, каждая из которых содержала одну пару «доменное имя — IP -адрес», например:

rhino.acme.com — 102.54.94.97.

По мере роста Интернета файлы hosts.txt также увеличивались в объеме, поддерживать их становилось все сложнее, и создание масштабируемого решения для разрешения имен стало необходимостью.

Таким решением стала **система доменных имен** (Domain Name System, DNS). Эта служба состоит из некоторого количества серверов, рассеянных по всей составной сети, и множества клиентов, работающих практически на каждом конечном узле. DNS-серверы поддерживают распределенную базу отображений «доменное имя — IP -адрес», а DNS-клиенты обращаются к серверам с запросами о разрешении доменного имени в IP-адрес.

Служба DNS использует текстовые файлы почти такого формата, как и файлы hosts.txt, и эти файлы администратор также подготавливает вручную.

Однако в базе данных DNS предусмотрено несколько типов записей. Помимо основного типа записей — A, в которых устанавливается соответствие между DNS-именами и IP -адресами хостов, имеются и другие типы записей, расширяющие функциональность системы DNS. Так, например, запись типа MX указывает DNS-имя почтового сервера, относящегося к тому или иному домену имен, а запись типа SOA содержит электронный адрес и другую идентифицирующую информацию об администраторе, который создавал записи для этой базы данных.

Система DNS является распределенной, она опирается на иерархию доменов, и каждый сервер службы DNS хранит только часть имен сети, а не все имена, как это происходит при использовании файлов hosts.txt. При росте количества узлов в сети проблема масштабирования решается созданием новых доменов и поддоменов имен и добавлением в службу DNS новых серверов.

Для каждого домена имен создается свой DNS-сервер. Имеется два механизма распределения имен на серверах. В первом случае сервер может хранить отображения «доменное имя — IP -адрес» для всего домена, включая все его поддомены. Однако такое решение оказывается плохо масштабируемым, так какпри добавлении новых поддоменов нагрузка на этот сервер может превысить его возможности. Чаще используется другой подход, когда сервер домена хранит только имена, которые заканчиваются на следующем ниже уровне иерархии по сравнению с именем домена

(аналогично каталогу файловой системы, который содержит записи о файлах и подкаталогах, непосредственно в него «входящих»). Именно при такой организации службы DNS нагрузка по разрешению имен распределяется более-менее равномерно между всеми DNS-серверами сети. Например, в первом случае DNS-сервер домена mmt. ru будет хранить отображения для всех имен, заканчивающихся суффиксом mmt.ru (www.zil.mmt.ru, ftp.zil.mmt.ru, mail.mmt.ru и т. д.). Во втором случае этот сервер хранит отображения только имен типа mail.mmt.ru, www.mmt.ru, а все остальные отображения должны храниться на DNS-сервере поддомена zil.

Каждый DNS-сервер помимо таблицы отображений имен содержит ссылки на DNS-серверы своих поддоменов. Эти ссылки связывают отдельные DNS-серверы в единую службу DNS. Ссылки представляют собой IP -адреса соответствующих серверов. Для обслуживания корневого домена выделено несколько дублирующих друг друга DNS-серверов, IP -адреса которых являются общедоступными.

Существует две основные схемы разрешения DNS-имен. В первом варианте, называемом итеративным, DNS-клиент сам координирует работу по поиску адреса, циклически обращаясь с запросами к разным серверам имен:

1.DNS-клиент обращается к корневому DNS-серверу с указанием полного доменного имени.

2.DNS-сервер сообщает клиенту адрес следующего DNS-сервера, обслуживающего домен верхнего уровня, заданный в следующей старшей части запрошенного имени.

3.DNS-клиент обращается к следующему DNS-серверу, который отсылает его к DNS-серверу нужного поддомена, и т. д., пока не будет найден DNS-сервер, в котором хранится соответствие запрошенного имени IP -адресу. Этот сервер дает окончательный ответ клиенту.

Во втором варианте реализуется рекурсивная процедура:

1. DNS-клиент обращается с запросом к локальному DNS-серверу, то есть серверу, находящемуся с ним в одном домене (либо к DNS-серверу, адрес которого указан в его конфигурационных параметрах). Запрос представляет собой сообщение определенного формата, и смысл его сводится к простому вопросу, например: «Какой IP -адрес имеет хост mail.mmt.ru?»

2. Далее возможны два варианта действий:

а) Если локальный DNS-сервер знает ответ, то он сразу же возвращает его клиенту (это может произойти, когда запрошенное имя входит в тот же поддомен, к которому относится DNS-клиент, или когда сервер уже узнавал данное соответствие для другого клиента и сохранил его в своем кэше).

б) Если локальный DNS-сервер ответа не знает, то он обращается к корневому DNS-серверу с указанием полного доменного имени. Корневой DNS-сервер сообщает локальному DNS-серверу адрес следующего DNS-сервера, обслуживающего домен верхнего уровня, заданный в следующей старшей части запрошенного имени. Локальный DNS-сервер делает запрос к следующему DNS-серверу, который отсылает его к DNS-серверу нужного поддомена и т. д., пока не будет найден DNS-сервер, в котором хранится

соответствие запрошенного имени IP -адресу. Получив искомый IP -адрес, локальный DNS-сервер передает его DNS-клиенту.

Для ускорения поиска IP -адресов DNS-серверы широко применяют кэширование проходящих через них ответов. Чтобы служба DNS могла оперативно отрабатывать изменения, происходящие в сети, ответы кэшируются на относительно короткое время — обычно от нескольких часов до нескольких дней.

Протокол межсетевого взаимодействия IP

Этот раздел посвящен протоколу IP (Intrenet Protocol — межсетевой протокол), описанному в документе RFC 751.

Протокол IP относится к дейтаграммным протоколам, то есть протоколам, работающим *без установления соединений*, он поддерживает обработку каждого IP -пакета как независимой единицы обмена, не связанной с другими IP-пакетами. Основной задачей протокола IP является доставка пакета через составную сеть. Эта задача маршрутизации, а также поддержания интерфейсов с протоколами выше- и нижележащего уровней. Протокол IP реализует политику доставки с максимальными усилиями, то есть в протоколе IP нет механизмов, обычно применяемых для обеспечения достоверности конечных данных. Если во время продвижения пакета происходит какая-либо ошибка, то протокол IP по своей инициативе ничего не предпринимает для исправления этой ошибки. Например, если на промежуточном маршрутизаторе пакет был отброшен из-за ошибки по контрольной сумме, то модуль IP не пытается заново послать потерянный пакет.

В настоящее время как основа используется протокол IP версии 4 (Ipv4). IPv4 использует 32-битные (четырёхбайтные) адреса, ограничивающие адресное пространство 4 294 967 296 (2^{32}) возможными уникальными адресами.

Традиционной формой записи IPv4 адреса является запись в виде четырёх десятичных чисел (от 0 до 255), разделённых точками. Через дробь указывается длина маски подсети.

Но IPv4 не удовлетворяет текущим темпам роста сети Интернет и адресов четвертой версии не хватает всем устройствам в сети. Исчерпание адресов стало причиной, из-за которой был создан и принят ряд новых технологий, включая бесклассовую адресацию (CIDR) в 1993 году, NAT и новую версию Internet Protocol, IPv6, в 1998 году.

Переход Интернета на Internet Protocol версии 6 является единственным доступным долговременным решением проблемы исчерпания IPv4-адресов. Несмотря на то, что предсказанное исчерпание адресного пространства IPv4 вступило в заключительную стадию в 2008 году, большинство провайдеров интернет-услуг и разработчиков программного обеспечения только начинают внедрение IPv6.

Протокол DHCP

Для нормальной работы сети каждому сетевому интерфейсу компьютера и маршрутизатора должен быть назначен IP-адрес. Процедура присвоения

адресов происходит в ходе конфигурирования компьютеров и маршрутизаторов. Назначение IP -адресов может происходить вручную при выполнении процедуры конфигурирования интерфейса, для компьютера сводящейся, например, к заполнению системы экранных форм. При этом администратор должен помнить, какие адреса из имеющегося множества он уже использовал для других интерфейсов, а какие еще свободны. При конфигурировании помимо IP-адресов сетевых интерфейсов (и соответствующих масок) устройству сообщается ряд других конфигурационных параметров, необходимых для его эффективной работы, например, маска и IP -адрес маршрутизатора по умолчанию, IP-адрес DNS-сервера, доменное имя компьютера и т. п. Даже при не очень большом размере сети эта работа представляет для администратора утомительную процедуру.

DHCP (Dynamic Host Configuration Protocol) — протокол динамической настройки узла. сетевой протокол, позволяющий компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP.

Протокол DHCP работает в соответствии с моделью клиент-сервер. Когда компьютер включают, установленный на нем DHCP-клиент посылает ограниченное широковещательное сообщение, предназначенное для поиска DHCP-сервера. DHCP-сервер, который должен находиться в одной подсети с клиентами, откликается и посылает сообщение-ответ, содержащее IP-адрес и некоторые другие конфигурационные параметры.

DHCP-сервер может работать в разных режимах, включая:

- ручное назначение статических адресов;
- автоматическое назначение статических адресов;
- автоматическое распределение динамических адресов.

Во всех режимах работы администратор должен предварительно выполнить конфигурирование DHCP-сервера, сообщив ему один или несколько диапазонов доступных для распределения IP -адресов. Все эти адреса должны относиться к одной и той же подсети.

В ручном режиме администратор, помимо пула доступных адресов, снабжает DHCP-сервер информацией о жестком соответствии IP-адресов физическим адресам или другим идентификаторам клиентских узлов. DHCP-сервер, пользуясь этой информацией, всегда выдает определенному DHCP-клиенту один и тот же назначенный ему администратором IP-адрес (а также набор других конфигурационных параметров).

В режиме автоматического назначения статических адресов DHCP-сервер самостоятельно, без вмешательства администратора, произвольным образом выбирает клиенту IP -адрес из пула наличных IP -адресов. Адрес дается клиенту из пула в постоянное пользование, то есть между идентифицирующей информацией клиента и его IP -адресом по-прежнему, как и при ручном назначении, существует постоянное соответствие. Оно устанавливается в момент первого назначения DHCP-сервером IP -адреса клиенту. При всех последующих запросах сервер возвращает клиенту тот же самый IP-адрес.

При динамическом распределении адресов DHCP-сервер выдает адрес клиенту на ограниченное время, называемое сроком аренды. Срок аренды

диктует, как долго компьютер может применять назначенный IP -адрес, перед тем как снова запросить его от DHCP-сервера. Срок аренды зависит от режима работы пользователей сети. Если это небольшая сеть учебного заведения, куда со своими компьютерами приходят многочисленные студенты для выполнения лабораторных работ, то срок аренды может быть равен длительности лабораторной работы. Если же это корпоративная сеть, в которой сотрудники предприятия работают на регулярной основе, то срок аренды может быть достаточно длительным —до нескольких дней или даже недель.

Когда компьютер, являющийся DHCP-клиентом, удаляется из подсети, назначенный ему IP -адрес автоматически освобождается и может быть назначен другому DHCP-клиенту. В общем случае при каждом следующем подключении к сети компьютеру автоматически назначается новый адрес. Ни пользователь, ни сетевой администратор не вмешиваются в этот процесс. Такое свойство DHCP дает возможность динамически разделять адреса между несколькими компьютерами.

Давайте рассмотрим преимущества динамического разделения пула адресов
на примере организации, в которой сотрудники значительную часть рабочего времени проводят вне офиса — дома или в командировках. Каждый из них имеет портативный компьютер, который во время пребывания в офисе подключается к корпоративной IP-сети. Возникает вопрос, сколько IP-адресов необходимо этой организации?

Первый ответ — столько, скольким сотрудникам необходим доступ в сеть. Если их 500 человек, то каждому из них должен быть назначен IP -адрес и выделено рабочее место. Однако вспомним, что сотрудники в этой организации редко появляются в офисе, значит, большая часть ресурсов при таком решении будет простаивать.

Второй ответ — столько, сколько сотрудников обычно присутствует в офисе (с некоторым запасом). Если обычно в офисе работает не более 50 сотрудников, то достаточно получить у поставщика услуг пул из 64 адресов и установить в рабочем помещении сеть с 64 коннекторами для подключения компьютеров. Но тут возникает другая проблема: кто и как будет конфигурировать компьютеры, состав которых постоянно меняется?

И эта проблема имеет два решения. Во-первых, администратор (или сам мобильный пользователь) может конфигурировать компьютер вручную каждый раз, когда возникает необходимость подключения к офисной сети. Такой подход требует от администратора (или пользователей) большого объема рутинной работы, следовательно, это — плохое решение. Гораздо привлекательнее выглядят возможности автоматического динамического назначения адресов DHCP. Действительно, администратору достаточно один раз при настройке DHCP сервера указать диапазон из 64 адресов, а каждый вновь прибывающий мобильный пользователь будет просто физически подключать в сеть свой компьютер, на котором запускается DHCP-клиент. Он запросит конфигурационные параметры и автоматически получит их от

DHCP-сервера. Таким образом, для работы 500 мобильных сотрудников достаточно иметь в офисной сети 64 IP-адреса и 64 рабочих места.

Промышленные сети

Промышленная сеть — сеть передачи данных, связывающая различные датчики, исполнительные механизмы, промышленные контроллеры и используемая в промышленной автоматизации. Термин употребляется преимущественно в автоматизированных системах управления технологическими процессами (АСУТП).

Примеры промышленных сетей:

Modbus — самая простая, дешёвая и широко распространённая промышленная сеть;

HART — сеть для аналоговых датчиков и их настройки;

CAN — промышленная сеть для автоматизации транспорта и машиностроения;

ProfiBus — промышленная сеть, международный стандарт, созданный с активным участием фирмы Siemens AG;

Промышленный Ethernet — версия Ethernet для применения в промышленности.

Modbus

Modbus — открытый коммуникационный протокол, основанный на архитектуре ведущий-ведомый (master-slave). Широко применяется в промышленности для организации связи между электронными устройствами. Может использоваться для передачи данных через последовательные линии связи RS-485, RS-422, RS-232, и сети TCP/IP (Modbus TCP).

Достоинства стандарта

Основные достоинства стандарта — открытость и массовость. Промышленностью сейчас (2014 г.) выпускается очень много типов и моделей датчиков, исполнительных устройств, модулей обработки и нормализации сигналов и др. Практически все промышленные системы контроля и управления имеют программные драйверы для работы с MODBUS-сетями.

Недостатки стандарта

Стандарт в своей основе был разработан в 1979 году с учётом потребностей и вычислительных возможностей того времени, и многие актуальные для современных промышленных сетей вопросы не были учтены. Необходимо отметить, что отсутствие перечисленных возможностей является следствием простоты протокола, которая облегчает его изучение и ускоряет внедрение.

- Стандарт специфицирует метод передачи только двух типов данных.
- Стандарт не регламентирует начальную инициализацию системы. Назначение сетевых адресов и прописывание в системе параметров каждого конкретного устройства выполняются вручную.

- Не предусмотрена передача сообщений по инициативе подчинённого устройства (прерываний). Ведущее устройство должно периодически опрашивать ведомые.
- Не предусмотрен способ, с помощью которого подчинённое устройство могло бы обнаружить потерю связи с ведущим.

Profibus

Profibus (Process Field Bus — шина полевого уровня) — открытая промышленная сеть, прототип которой был разработан компанией Siemens AG. На основе этого прототипа Организация пользователей Profibus разработала международные стандарты, принятые затем некоторыми национальными комитетами по стандартизации.

Сеть Profibus — это комплексное понятие, она основывается на нескольких стандартах и протоколах.

Profibus определяет следующие уровни:

- Физический уровень — отвечает за характеристики физической передачи;
- Канальный уровень — определяет протокол доступа к шине;
- Уровень приложений — отвечает за прикладные функции.

Физический уровень Profibus

Физически Profibus может представлять собой:

- электрическую сеть с шинной топологией, использующую экранированную витую пару, соответствующую стандарту RS-485;
- оптическую сеть на основе волоконно-оптического кабеля;
- инфракрасную сеть.

Скорость передачи по ней может варьироваться от 9,6 Кбит/сек до 12 Мбит/сек.

Протоколы сети Profibus

Одни и те же каналы связи сети Profibus допускают одновременное использование нескольких протоколов передачи данных:

- Profibus DP (Decentralized Peripheral — распределённая периферия) — протокол, ориентированный на обеспечение скоростного обмена данными между:
  - системами автоматизации (ведущими DP-устройствами),
  - устройствами распределённого ввода-вывода (ведомыми DP-устройствами).

  Протокол характеризуется минимальным временем реакции и высокой стойкостью к воздействию внешних электромагнитных полей. Оптимизирован для высокоскоростных и недорогих систем.
- Profibus PA (англ. Process Automation — автоматизация процесса) — протокол обмена данными с оборудованием полевого уровня, расположенным в обычных или Ex-зонах (взрывоопасных зонах).

Позволяет подключать датчики и приводы на одну линейную шину или кольцевую шину.

- Profibus FMS (англ. Fieldbus Message Specification — спецификация сообщений полевого уровня) — универсальный протокол для решения задач по обмену данными между интеллектуальными сетевыми устройствами (контроллерами, компьютерами/программаторами, системами человеко-машинного интерфейса) на полевом уровне. Некоторый аналог промышленного Ethernet, обычно используется для высокоскоростной связи между контроллерами и компьютерами верхнего уровня и используемыми диспетчерами. Скорость до 12 Мбит/с.

Все протоколы используют одинаковые технологии передачи данных и общий метод доступа к шине, поэтому они могут функционировать на одной шине.

1.4. ОСНОВЫ АДМИНИСТРИРОВАНИЯ И УПРАВЛЕНИЯ В ЛВС

1.4.1. Методы обеспечения безопасности и сохранения данных.

Вопросы для изучения:

- Некоторые виды технических угроз: программные ошибки, DoS и DDoS-атаки, вредоносное ПО;
- Некоторые способы зашиты информации: шифрование, электронная цифровая подпись, частные виртуальные сети.

Некоторые виды технических угроз:

Программная ошибка (*жарг. «Баг»*) — означает ошибку в программе или в системе, из-за которой программа выдает неожиданное поведение и, как следствие, результат. Большинство программных ошибок возникают из-за ошибок, допущенных разработчиками программы в её исходном коде, либо в её дизайне. Также некоторые ошибки возникают из-за некорректной работы инструментов разработчика, например из-за компилятора, вырабатывающего некорректный код.

Обычно уязвимость позволяет атакующему «обмануть» приложение — заставить его совершить действие, на которое у того не должно быть прав. Это делается путём внедрения каким-либо образом в программу данных или кода в такие места, что программа воспримет их как «свои». Некоторые уязвимости появляются из-за недостаточной проверки данных, вводимых пользователем, и позволяют вставить в интерпретируемый код произвольные команды (SQL-инъекция, XSS, SiXSS). Другие уязвимости появляются из-за более сложных проблем, таких как запись данных в буфер без проверки его границ (переполнение буфера).

DoS (аббр. англ. *Denial of Service* «отказ в обслуживании») — хакерская атака на вычислительную систему с целью довести её до отказа, то есть создание таких условий, при которых добросовестные пользователи системы не могут получить доступ к предоставляемым системным ресурсам (серверам), либо этот доступ затруднён. Отказ «вражеской» системы может быть и шагом к овладению системой (если в нештатной ситуации ПО выдаёт какую-либо критическую информацию — например, версию, часть программного кода и т. д.). Но чаще это мера экономического давления: потеря простой службы, приносящей доход, счета от провайдера и меры по уходу от атаки ощутимо бьют «цель» по карману. В настоящее время DoS и DDoS-атаки наиболее популярны, так как позволяют довести до отказа практически любую систему, не оставляя юридически значимых улик.

Если атака выполняется одновременно с большого числа компьютеров, говорят о **DDoS-атаке** (от англ. *Distributed Denial of Service*, *распределённая атака типа «отказ в обслуживании»*). Такая атака проводится в том случае, если требуется вызвать отказ в обслуживании хорошо защищённой крупной компании или правительственной организации.

Первым делом злоумышленник сканирует крупную сеть с помощью специально подготовленных сценариев, которые выявляют потенциально слабые узлы. Выбранные узлы подвергаются нападению, и злоумышленник получает на них права администратора. На захваченные узлы устанавливаются троянские программы, которые работают в фоновом режиме.[3] Теперь эти компьютеры называются компьютерами-зомби, их пользователи даже не подозревают, что являются потенциальными участниками DDoS-атаки. Далее злоумышленник отправляет определенные команды захваченным компьютерам и те, в свою очередь осуществляют мощную DoS-атаку на целевой компьютер.

Существуют также программы для добровольного участия в DDoS-атаках.

В некоторых случаях к фактической DDoS-атаке приводит непреднамеренное действие, например, размещение на популярном интернет-ресурсе ссылки на сайт, размещённый на не очень производительном сервере (слэшдот-эффект). Большой наплыв пользователей приводит к превышению допустимой нагрузки на сервер и, следовательно, отказу в обслуживании части из них.

Защита

Для защиты от сетевых атак применяется ряд фильтров, подключенных к интернет-каналу с большой пропускной способностью. Фильтры действуют таким образом, что последовательно анализируют проходящий трафик, выявляя нестандартную сетевую активность и ошибки. В число анализируемых шаблонов нестандартного трафика входят все известные на сегодняшний день методы атак, в том числе реализуемые и при помощи распределённых бот-сетей.

Вредоносное ПО

Вредоносное ПО - любое программное обеспечение, предназначенное для получения несанкционированного доступа к вычислительным ресурсам самой ЭВМ или к информации, хранимой на ЭВМ, с целью несанкционированного использования ресурсов ЭВМ или причинения вреда (нанесения ущерба) владельцу информации, и/или владельцу ЭВМ, и/или владельцу сети ЭВМ, путём копирования, искажения, удаления или подмены информации. Многие антивирусы считают крэки (кряки), кейгены и прочие программы для взлома приложений вредоносными программами, или потенциально опасными.

Классификация:

По вредоносной нагрузке:

1. Инсталляция другого вредоносного ПО.
    1. Загрузка из сети (*downloader*).
    2. Распаковка другой вредоносной программы, уже содержащейся внутри файла (*dropper*).
2. Кража, мошенничество, вымогательство и шпионаж за пользователем. Для кражи может применяться сканирование жёсткого диска, регистрация нажатий клавиш (Keylogger) и перенаправление пользователя на поддельные сайты, в точности повторяющие исходные ресурсы.
    1. Кража аккаунтов различных служб (электронной почты, мессенджеров, игровых серверов…).
    2. Кража аккаунтов платёжных систем.
    3. Блокировка компьютера, шифрование файлов пользователя с целью шантажа и вымогательства денежных средств (см. Ransomware). В большинстве случаев после оплаты компьютер или не разблокируется, или вскоре блокируется второй раз.
    4. Платное ПО, имитирующее, например, антивирус, но ничего полезного не делающее (fraudware или scareware (англ.)русск.; см. тж лжеантивирус).
3. Получение несанкционированного (и/или дарового) доступа к ресурсам самого компьютера или третьим ресурсам, доступным через него, в том числе прямое управление компьютером (так называемый backdoor).
4. Организация на компьютере открытых релеев и общедоступных прокси-серверов.
5. Заражённый компьютер (в составе ботнета) может быть использован для проведения DDoS-атак.
6. Сбор адресов электронной почты и распространение спама, в том числе в составе ботнета.
7. Накрутка электронных голосований, щелчков по рекламным баннерам.
8. Генерация криптовалют.

Файлы, не являющиеся истинно вредоносными, но в большинстве случаев нежелательные:

1. Шуточное ПО, делающее какие-либо беспокоящие пользователя вещи.
2. Adware — программное обеспечение, показывающее рекламу.

3. Spyware — программное обеспечение, занимающееся массовым сбором малоценной информации — например, конфигурации компьютера, каталогов диска, активности пользователя…
4. «Отравленные» документы, дестабилизирующие ПО, открывающее их (например, архив размером меньше мегабайта может содержать гигабайты данных и надолго «завесить» архиватор).
5. Программы удалённого администрирования могут применяться как для того, чтобы дистанционно решать проблемы с компьютером, так и для неблаговидных целей.
6. Руткит нужен, чтобы скрывать другое вредоносное ПО от посторонних глаз.
7. Иногда вредоносное ПО для собственного «жизнеобеспечения» устанавливает дополнительные утилиты: IRC-клиенты, программные маршрутизаторы, открытые библиотеки перехвата клавиатуры… Такое ПО вредоносным не является, но из-за того, что за ним часто стоит истинно вредоносная программа, детектируется антивирусами. Бывает даже, что вредоносным является только скрипт из одной строчки, а остальные программы вполне легитимны.

По методу размножения:

- Эксплойт — теоретически безобидный набор данных (например, графический файл или сетевой пакет), некорректно воспринимаемый программой, работающей с такими данными. Здесь вред наносит не сам файл, а неадекватное поведение ПО с ошибкой. Также эксплойтом называют программу для генерации подобных «отравленных» данных.
- Логическая бомба в программе срабатывает при определённом условии, и неотделима от полезной программы-носителя.
- Троянская программа не имеет собственного механизма размножения.
- Компьютерный вирус размножается в пределах компьютера и через сменные диски. Размножение через сеть возможно, если пользователь сам выложит заражённый файл в сеть. Вирусы, в свою очередь, делятся по типу заражаемых файлов (файловые, загрузочные, макро-, автозапускающиеся); по способу прикрепления к файлам (паразитирующие, «спутники» и перезаписывающие) и т. д.
- Сетевой червь способен самостоятельно размножаться по сети. Делятся на IRC-, почтовые, размножающиеся с помощью эксплойтов и т. д.

Вредоносное ПО может образовывать цепочки: например, с помощью эксплойта на компьютере жертвы развёртывается загрузчик, устанавливающий из интернета червя.

Некоторые способы зашиты информации

Шифрование — базовая технология безопасности

Шифрование является краеугольным камнем всех служб информационной безопасности, будь то система аутентификации или авторизации, защищенный канал или средства безопасного хранения данных.

Шифрование — это обратимое преобразование информации в целях обеспечения конфиденциальности данных. Дешифрование — процедура, которая, будучи примененной к зашифрованному тексту, снова приводит его в исходное состояние.

Пара процедур — шифрование и дешифрование — называется **криптосистемой**. Обычно криптосистема предусматривает наличие специального элемента — секретного ключа.

Криптосистема считается раскрытой , если найдена процедура, позволяющая подобрать ключ за реальное время. Методы раскрытия криптосистемы, процедуры выявления уязвимости криптографических алгоритмов, выяснение секретного ключа называют **криптоанализом**, или взломом шифра. Попытку раскрытия конкретного шифра с применением методов криптоанализа называют **криптографической атакой**.

Существует два класса криптосистем — **симметричные** и **асимметричные**. В симметричных схемах шифрования (классическая криптография) секретный ключ шифрования совпадает с секретным ключом дешифрования.

В асимметричных схемах шифрования (криптография с открытым ключом) ключ шифрования (открытый) не совпадает с ключом дешифрования (закрытый). В таких системах шифрования и/или электронной подписи (ЭП) *открытый ключ* передаётся по открытому (то есть незащищённому, доступному для наблюдения) каналу и используется для проверки ЭП и для шифрования сообщения. Для генерации ЭП и для расшифровки сообщения используется закрытый ключ. Криптографические системы с открытым ключом в настоящее время широко применяются в различных сетевых протоколах, в частности, в протоколах TLS и его предшественнике SSL (лежащих в основе HTTPS), в SSH, в PGP.

Электронная подпись (ЭП), **Электро́нная цифровая по́дпись** (ЭЦП), **Цифровая подпись** (ЦП) — реквизит электронного документа, полученный в результате криптографического преобразования информации с использованием закрытого ключа подписи и позволяющий проверить отсутствие искажения информации в электронном документе с момента формирования подписи (целостность), принадлежность подписи владельцу сертификата ключа подписи (авторство), а в случае успешной проверки подтвердить факт подписания электронного документа (неотказуемость).

Виртуальные частные сети

VPN (*Virtual Private Network*) — обобщённое название технологий, позволяющих обеспечить одно или несколько сетевых соединений (логическую сеть) поверх другой сети (например, Интернет). Несмотря на то, что коммуникации осуществляются по сетям с меньшим или неизвестным уровнем доверия (например, по публичным сетям), уровень доверия к построенной логической сети не зависит от уровня доверия к базовым сетям благодаря использованию средств криптографии (шифрования, аутентификации, инфраструктуры открытых ключей, средств для защиты от повторов и изменений передаваемых по логической сети сообщений).

В зависимости от применяемых протоколов и назначения, VPN может обеспечивать соединения трёх видов: узел-узел, узел-сеть и сеть-сеть.

1.4.2. Защита ЛВС от компьютерных вирусов

Вопросы для изучения:

- Основные принципы построения антивирусной защиты сети.

Основные принципы построения антивирусной защиты сети.

В основу построения системы антивирусной защиты сети могут быть положены следующие принципы:

- принцип реализации единой технической политики при обосновании выбора антивирусных продуктов для различных сегментов локальной сети;
- принцип полноты охвата системой антивирусной защиты всей локальной сети организации;
- принцип непрерывности контроля локальной сети предприятия, для своевременного обнаружения компьютерной инфекции;
- принцип централизованного управления антивирусной защитой;

Принцип реализации единой технической политики предусматривает использование во всех сегментах локальной сети только антивирусного ПО, рекомендуемого подразделением антивирусной защиты предприятия. Эта политика носит долгосрочный характер, утверждается руководством предприятия и является основой для целевого и долговременного планирования затрат на приобретение антивирусных программных продуктов и их дальнейшее обновление.

Принцип полноты охвата системой антивирусной защиты локальной сети предусматривает постепенной внедрение в сеть программных средств антивирусной защиты до полного насыщения в сочетании с организационно-режимными мерами защиты информации.

Принцип непрерывности контроля за антивирусным состоянием локальной сети подразумевает такую организацию ее защиты, при которой обеспечивается постоянная возможность отслеживания состояния сети для выявления вирусов.

Принцип централизованного управления антивирусной защитой предусматривает управление системой из одной органа с использованием технических и программных средств. Именно этот орган организует централизованный контроль в сети, получает данные контроля или доклады пользователей со своих рабочих мест об обнаружении вирусов и обеспечивает внедрение принятых решений по управлению системой антивирусной защиты.

Для решения этих задач в комплексной системе информационной безопасности кроме администраторов информационной безопасности создаются администраторы антивирусной защиты. Если ЛВС небольшая или достаточно хорошо оснащена антивирусным ПО, то назначение специального

администратора антивирусной защиты чаще всего нецелесообразно, так как его функции может выполнять администратор безопасности сети.

Для организации функционирования антивирусной защиты необходима разработка внутренних организационно-распорядительных документов. Кроме того, должны быть определены порядки передачи сообщений о вирусах от пользователей и оповещений администраторов о фактах и возможностях вирусных заражений локальной сети.

Эффективность создаваемой подсистемы антивирусной защиты зависит также от выполнения следующих дополнительных условий:

- подключение ПК пользователей в корпоративную сеть должно производиться только по заявке с отметкой администратора антивирусной защиты об установке лицензионного антивирусного ПО (заявка заносится в базу данных с фиксацией сроков действия лицензии);
- передачу ПК от одного пользователя другому необходимо производить с переоформлением подключения к сети;
- обнаруженные вирусы целесообразно исследовать на стенде подразделения защиты информации с целью выработки рекомендаций по их корректному обезвреживанию;
- в удаленных структурных подразделениях следует назначить внештатных сотрудников, ответственных за антивирусную защиту.

Практическая реализация антивирусной защиты информации на серверах и ПК корпоративной сети осуществляется с использованием ряда программно-технических методов, являющихся стандартными, но имеющих свою специфику, определяемую особенностями корпоративной сети. К ним относятся:

- использование антивирусных пакетов;
- архивирование информации;
- резервирование информации;
- ведение базы данных о вирусах и их характеристиках;

Рассмотрим эти методы более подробно.

Главным методом антивирусной защиты является установка антивирусных пакетов. Выбор антивирусного ПО является одной из важнейших задач антивирусной защиты, от правильности решения которой в дальнейшем будут зависеть антивирусная безопасность системы, а также затраты на ее поддержание. Используемые антивирусные средства должны удовлетворять следующим общим требованиям:

- система должны быть совместима с операционными системами серверов и ПК;
- система антивирусной защиты не должна нарушать логику работы остальных используемых приложений;
- наличие полного набора антивирусных функций, необходимых для обеспечения антивирусного контроля и обезвреживания всех известных вирусов;

- частота обновления антивирусного ПО и гарантии поставщиков (разработчиков) в отношении его своевременности.

В отличие от других подсистем информационной безопасности в рассматриваемой области отсутствуют четко сформулированные показатели защищенности и соответствующие критерии сравнения различных антивирусных средств. Как правило антивирусные комплексы сравниваются по следующим показателям: обнаружение, лечение, блокирование, восстановление, регистрация, обеспечение целостности, обновление базы данных компьютерных вирусов, защита антивирусных средств от доступа паролем, средства управления, гарантии проектирования, документация.

При комплексной защите локальной сети необходимо уделить внимание всем возможным точкам проникновения вирусов в сеть извне.

Применение антивирусов для межсетевых экранов на сегодняшний день сводится к осуществлению фильтрации доступа в Интернет при одновременной проверке на вирусы проходящего трафика. Осуществляемая такими продуктами антивирусная проверка сильно замедляет работу и имеет крайне не высокий уровень обнаружения, по этому в отсутствии необходимости фильтрации посещаемых пользователями веб-узлов применение таких продуктов является не целесообразным.

Антивирусной защите подлежат все компоненты информационной системы, участвующие в транспортировке информации и/или её хранении:

- файл-серверы;
- рабочие станции;
- рабочие станции мобильных пользователей;
- сервера резервного копирования;
- почтовые сервера.

Следующими по важности методами антивирусной защиты являются архивирование и резервное копирование информации, позволяющие исключить потерю информации в случае вирусного заражения. Архивирование заключается в периодическом копировании системных областей машинных носителей информации на внешние устройства. На серверах с наиболее важной информацией архивирование необходимо проводить с минимальной периодичностью. Резервное копирование информации проводится ежедневно в целях защиты ее от искажения и разрушения.

1.4.3. Модели администрирования и регистрации в сети

Вопросы для изучения:

- Доменная модель. Модель рабочей группы;
- Служба Active Directory.

Организация домена

Доменное имя — символьное имя, служащее для идентификации областей — единиц административной автономии в сети Интернет — в составе вышестоящей по иерархии такой области. Каждая из таких областей называется доменом. Общее пространство имён Интернета функционирует благодаря DNS — системе доменных имён. Доменные имена дают возможность адресации интернет-узлов и расположенных на них сетевых ресурсов (веб-сайтов, серверов электронной почты, других служб) в удобной для человека форме.

Полное доменное имя состоит из непосредственного имени домена и далее имён всех доменов, в которые он входит, разделённых точками. Например, полное имя «ru.wikipedia.org» обозначает домен третьего уровня «ru», который входит в домен второго уровня «wikipedia», который входит в домен верхнего уровня «org», который входит в безымянный корневой домен «.» (точка). В обыденной речи под доменным именем нередко понимают именно полное доменное имя.

FQDN (сокр. от англ. Fully Qualified Domain Name — «полностью определённое имя домена», иногда сокращается до «полное доменное имя» или «полное имя домена») — имя домена, не имеющее неоднозначностей в определении. Включает в себя имена всех родительских доменов иерархии DNS.

В DNS и, что особенно существенно, в файлах зоны (англ.), FQDN завершаются точкой (например, «example.com.»), то есть включают корневое доменное имя «.», которое является безымянным.

Различие между FQDN и доменным именем появляется при именовании доменов второго, третьего (и так далее) уровней. Для получения FQDN требуется обязательно указать в имени домены более высокого уровня. Например, «sample» является доменным именем, однако его полное доменное имя (FQDN) выглядит как доменное имя пятого уровня — «sample.gtw-02.office4.example.com.», где:

«sample» 5-й уровень;
«gtw-02» 4-й уровень;
«office4» 3-й уровень;
«example» 2-й уровень;
«com» 1-й (верхний) уровень;
«.» 0-й (корневой) уровень.

В DNS-записях доменов (для перенаправления, почтовых серверов и так далее) всегда используются FQDN. Обычно в практике сложилось написание полного доменного имени за исключением постановки последней точки перед корневым доменом, например, «sample.gtw-02.office4.example.com».

Доме́нная зона — совокупность доменных имён определённого уровня, входящих в конкретный домен. Например, зона wikipedia.org включает все доменные имена третьего уровня в этом домене. Термин «доменная зона» в основном применяется в технической сфере, при настройке DNS-серверов (поддержание зоны, делегирование зоны, трансфер зоны).

Active Directory

Active Directory («Активный каталог», AD) — службы каталогов корпорации Microsoft для операционных систем семейства Windows Server. Первоначально создавалась, как LDAP-совместимая реализация службы каталогов, однако, начиная с Windows Server 2008, включает возможности интеграции с другими службами авторизации, выполняя для них интегрирующую и объединяющую роль. Позволяет администраторам использовать групповые политики для обеспечения единообразия настройки пользовательской рабочей среды, разворачивать программное обеспечение на множестве компьютеров через групповые политики или посредством System Center Configuration Manager (ранее — Microsoft Systems Management Server), устанавливать обновления операционной системы, прикладного и серверного программного обеспечения на всех компьютерах в сети, используя Службу обновления Windows Server. Хранит данные и настройки среды в централизованной базе данных. Сети Active Directory могут быть различного размера: от нескольких десятков до нескольких миллионов объектов.

Active Directory имеет иерархическую структуру, состоящую из объектов. Объекты разделяются на три основные категории: ресурсы (например, принтеры), службы (например, электронная почта) и учётные записи пользователей и компьютеров. Служба предоставляет информацию об объектах, позволяет организовывать объекты, управлять доступом к ним, а также устанавливает правила безопасности.

Объекты.Объекты могут быть хранилищами для других объектов (группы безопасности и распространения). Объект уникально определяется своим именем и имеет набор атрибутов — характеристик и данных, которые он может содержать; последние, в свою очередь, зависят от типа объекта. Атрибуты являются составляющей базой структуры объекта и определяются в схеме. Схема определяет, какие типы объектов могут существовать.

Сама схема состоит из двух типов объектов: объекты классов схемы и объекты атрибутов схемы. Один объект класса схемы определяет один тип объекта Active Directory (например, объект «Пользователь»), а один объект атрибута схемы определяет атрибут, который объект может иметь.

Каждый объект атрибута может быть использован в нескольких разных объектах классов схемы. Эти объекты называются объектами схемы (или метаданными) и позволяют изменять и дополнять схему, когда это необходимо и возможно. Однако каждый объект схемы является частью определений объектов, поэтому отключение или изменение этих объектов могут иметь серьёзные последствия, так как в результате этих действий будет изменена структура каталогов. Изменение объекта схемы автоматически распространяется в службе каталогов. Будучи однажды созданным, объект схемы не может быть удалён, он может быть только отключён. Обычно все изменения схемы тщательно планируются.

Контейнер аналогичен объекту в том смысле, что он также имеет атрибуты и принадлежит пространству имён, но, в отличие от объекта,

контейнер не обозначает ничего конкретного: он может содержать группу объектов или другие контейнеры.

Структура. Верхним уровнем структуры является лес — совокупность всех объектов, атрибутов и правил (синтаксиса атрибутов) в Active Directory. Лес содержит одно или несколько деревьев, связанных транзитивными отношениями доверия. Дерево содержит один или несколько доменов, также связанных в иерархию транзитивными отношениями доверия. Домены идентифицируются своими структурами имён DNS — пространствами имён.

Объекты в домене могут быть сгруппированы в контейнеры — подразделения. Подразделения позволяют создавать иерархию внутри домена, упрощают его администрирование и позволяют моделировать, например, организационную или географическую структуру организации в службе каталогов. Подразделения могут содержать другие подразделения. Microsoft рекомендует использовать как можно меньше доменов в службе каталогов, а для структурирования и политик использовать подразделения. Часто групповые политики применяются именно к подразделениям. Групповые политики сами являются объектами. Подразделение является самым низким уровнем, на котором могут делегироваться административные полномочия.

Хозяева операций.Schema master — Хозяин схемы. По сути это компьютер или виртуальная машинка, которая управляет всем, что находится в схеме. Обновление схемы леса невозможно без доступа к этой роли fsmo. Очень важно, что на лес она одна.

Domain naming master - Хозяин именования доменов. Это контроллер домена служащий для управления, добавлением и удалением доменов в лесу. Так же обеспечивает уникальность имен. В лесу он также один.

PDC emulator - PDC emulator. Пожалуй, самая нужная роль fsmo, так как без двух верхних вы еще проживете если они сломались, а вот без этого товарища никак, вот почему: является основным обозревателем в сети Windows, если пользователя заблокировала политика неправильно введенных паролей, эта роль вам об этом сообщает, главный NTP сервер в вашем домене, объявляет себя главным контроллером домена для рабочих станций и серверов прошлых версий (XP, 2000), обновление групповых политик. На каждый домен свой PDC эмулятор.

Infrastructure Master - Хозяин инфраструктуры. В каждом домене он свой. Отвечает за обновление ссылок объектов домена на объекты других доменов, например, SID, GUID. Если бы его не было или был сломан, то вы не смогли бы выполнить команду adprep /domainprep. Средняя значимость из всех fsmo.

RID Master - Хозяин относительных идентификаторов (RID). Каждый объект active directory имеет свой уникальный sid, по сути эта роль их генерирует. Изначально контроллер домена заказывает у этой роли 500 sid, если они заканчиваются, просит еще.

Различные уровни взаимодействия с Active Directory могут быть реализованы в большинстве UNIX-подобных операционных систем посредством LDAP-клиентов, но такие системы, как правило, не воспринимают большую часть атрибутов, ассоциированных с компонентами

Windows, например, групповые политики и поддержку односторонних доверенностей. Большинстве современных сетей TCP/IP используется служба DNS, главное назначение которой — преобразовывать простые для запоминания имена типаcompany.com в IP-адреса. Для этого каждый компьютер-сервер DNS имеет набор записей с информацией о ресурсах. Каждая запись имеет некоторый тип, определяющий характер и назначение хранящейся информации. Например, запись типа A применяется для преобразования доменного имени компьютера в заданный IP-адрес, а запись типа MX — для поиска почтового сервера в определенном почтовом домене. Каждый DNS-сервер «знает» свое место в глобальном пространстве DNS-имен, что позволяет передавать неразрешенные запросы другим серверам. Поэтому пусть и не сразу, но почти каждый клиентский запрос находит нужный сервер, хранящий искомую информацию.

Интеграцию служб Active Directory и DNS можно рассматривать в трех аспектах:

•домены Active Directory и домены DNS имеют одинаковую иерархическую структуру и схожее пространство имен;

•зоны (zone) DNS могут храниться в Active Directory. Если используется сервер DNS, входящий в состав Windows .Server, то первичные зоны (primary zone), занесенные в каталог, реплицируются на все контроллеры домена, что обеспечивает лучшую защищенность службы DNS;

•использование клиентами службы DNS при поиске контроллеров домена.

Active Directory может использовать любую стандартную, законченную реализацию службы DNS: не обязательно задействовать DNS-сервер, входящий в Windows 2000 Server. Windows Server поддерживает также службу динамического именования хостов, Dynamic DNS. В соответствии с RFC 2136 служба Dynamic DNS расширяет протокол DNS, позволяя модифицировать базу данных DNS со стороны удаленных систем. Например, при подключении некоторый контроллер домена может сам добавлять SRV- запись для себя, освобождая администратора от такой необходимости.

1.4.4. Функции и архитектура систем управления сетями.

Вопросы для изучения:

- Управление конфигурацией сети.
- Обработка ошибок, анализ производительности и надежности, учет работы сети.

Как и любой сложный технический объект, компьютерная сеть требует выполнения различных действий для поддержания ее в рабочем состоянии, анализа и оптимизации ее производительности, защиты от внутренних и внешних угроз. Среди многообразия средств, привлекаемых для достижения этих целей, важное место занимают службы (системы) управления сетью.

Система управления сетью (Network Management System, NMS) — это сложный программно-аппаратный комплекс, который контролирует сетевой трафик и управляет коммуникационным оборудованием крупной компьютерной сети.

Системы управления сетью работают, как правило, в автоматизированном режиме, выполняя наиболее простые действия автоматически и оставляя человеку принятие сложных решений на основе подготовленной системой информации.

Система управления сетью предназначена для решения следующих групп задач:

- Управление конфигурацией сети и именованием заключается в конфигурировании параметров как отдельных элементов сети, так и сети в целом. Для элементов сети, таких как маршрутизаторы, мультиплексоры и т. п., конфигурирование состоит в назначении сетевых адресов, идентификаторов (имен), географического положения и пр. Для сети в целом управление конфигурацией обычно начинается с построения карты сети, то есть с отображения реальных связей между элементами сети и связей между ними.

- Обработка ошибок включает выявление, определение и устранение последствий сбоев и отказов.

- Анализ производительности и надежности связан с оценкой на основе накопленной статистической информации таких параметров, как время реакции системы, пропускная способность реального или виртуального канала связи между двумя конечными абонентами сети, интенсивность трафика в отдельных сегментах и каналах сети, а также вероятность искажения данных при их передаче через сеть. Результаты анализа производительности и надежности позволяют контролировать соглашение об уровне обслуживания (SLA), заключаемое между пользователем сети и ее администраторами (или компанией, продающей услуги). Без средств анализа производительности и надежности поставщик услуг публичной сети или отдел информационных технологий предприятия не сможет ни проконтролировать, ни тем более обеспечить нужный уровень обслуживания для конечных пользователей сети.

- Управление безопасностью подразумевает контроль доступа к ресурсам сети (данным и оборудованию) и сохранение целостности данных при их хранении и передаче через сеть. Базовыми элементами управления безопасностью являются процедуры аутентификации пользователей, назначение и проверка прав доступа к ресурсам сети, распределение и поддержка ключей шифрования, управления полномочиями и т. п. Часто функции этой группы не включаются в системы управления сетями, а либо реализуются в виде специальных продуктов обеспечения безопасности, например сетевых экранов или централизованных систем авторизации, либо входят в состав операционных систем и системных приложений.

- Учет работы сети включает регистрацию времени использования различных ресурсов сети (устройств, каналов и транспортных служб) и ведение биллинговых операций (плата за ресурсы).

В стандартах систем управления не делается различий между управляемыми объектами, представляющими коммуникационное оборудование (каналы, сегменты локальных сетей, коммутаторы и маршрутизаторы, модемы и мультиплексоры), и объектами, представляющими аппаратное и программное обеспечение компьютеров. Однако на практике деление систем управления по типам управляемых объектов широко распространено.

В тех случаях, когда управляемыми объектами являются компьютеры, а также их системное и прикладное программное обеспечение, то для системы управления часто используют особое название — система управления системой (System Management System, SMS).

SMS обычно автоматически собирает информацию об установленных в сети компьютерах и создает записи в специальной БД об аппаратных и программных ресурсах. SMS может централизованно устанавливать и администрировать приложения, которые запускаются с серверов, а также удаленно измерять наиболее важные параметры компьютера, операционной системы, СУБД (например, коэффициент использования процессора или физической памяти, интенсивность страничных прерываний и др.). SMS позволяет администратору брать на себя удаленное управление компьютером в режиме эмуляции графического интерфейса популярных операционных систем.

1.4.5. Мониторинг и анализ локальных сетей

Вопросы для изучения:

- Протоколы SNMP, SNMPv3;
- Классификация средств мониторинга и анализа сети.

Протокол SNMP

Протокол SNMP (Simple Management Network Protocol — простой протокол сетевого администрирования) используется в качестве стандартного протокола взаимодействия менеджера и агента.

Протокол SNMP относится к прикладному уровню стека TCP/IP. Для транспортировки своих сообщений он использует дейтаграммный транспортный протокол UDP, который, как известно, не обеспечивает надежной доставки. Протокол TCP, организующий надежную передачу сообщений на основе соединений, весьма значительно загружает управляемые устройства, которые на момент разработки протокола SNMP были не очень мощными, поэтому от услуг протокола TCP было решено отказаться.

SNMP —это протокол типа «запрос-ответ», то есть на каждый запрос, поступивший от менеджера, агент должен передать ответ. Особенностью протокола является его чрезвычайная простота — он включает в себя всего несколько команд.

SNMP-сообщения, в отличие от сообщений многих других коммуникационных протоколов, не имеют заголовков с фиксированными полями. Любое SNMP-сообщение состоит из трех основных частей: версии протокола, общей строки и области данных.

Общая строка (community string) используется для группирования устройств, управляемых определенным менеджером. Общая строка является своего рода паролем, так как для того, чтобы устройства могли взаимодействовать по протоколу SNMP, они должны иметь одно и то же значение этого идентификатора (по умолчанию часто употребляется строка «public»). Однако этот механизм служит скорее для «распознавания» партнеров, нежели для безопасности.

В области данных содержатся описанные команды протокола, а также имена объектов и их значения. Область данных состоит из одного или более блоков, каждый из которых может относиться к одному из перечисленных типов команд протокола SNMP. Для каждого типа команды определен свой формат.

SNMPv3

Хотя SNMPv3 не приносит никаких изменений в протокол помимо добавления криптографической защиты, он является улучшением за счёт новых текстовых соглашений, концепций и терминологии.

Безопасность была большой проблемой SNMP с самого появления. Аутентификация в SNMP версий 1 и 2 сводилась не более чем к паролю (строке сообщества), который пересылался в открытом виде между менеджером и агентом.

В отличие от SNMPv1 и v2, в SNMPv3 каждое сообщение содержит параметры безопасности, которые закодированы как строка октетов. Значение этих параметров зависит от используемой модели безопасности.

SNMPv3 предоставляет важные особенности безопасности:

- Аутентификация — определение источника сообщения.
- Конфиденциальность — шифрование пакетов для защиты от перехвата.
- Целостность — предотвращение изменений сообщений в пути, включая дополнительный механизм защиты от повторной трансляции перехваченного пакета.

С 2004 года IETF признаёт SNMPv3, определённый в RFC 3411, RFC 3418 (также известный как STD0062) в качестве текущей стандартной версии SNMP. IETF отметил SNMPv3 как полный Интернет-стандарт, что является самым высоким уровнем готовности для RFC. При этом более ранние версии считаются устаревшими (обозначаются как «исторические» — Historic).

На практике в реализациях SNMP часто поддерживаются несколько версий: v1, v2c и v3.

Классификация средств мониторинга и анализа

Все многообразие средств, применяемых для мониторинга и анализа вычислительных сетей, можно разделить на несколько крупных классов:

Системы управления сетью (NetworkManagementSystems) — централизованные программные системы, которые собирают данные о

состоянии узлов и коммуникационных устройств сети, а также данные о трафике, циркулирующем в сети. Эти системы не только осуществляют мониторинг и анализ сети, но и выполняют в автоматическом или полуавтоматическом режиме действия по управлению сетью — включение и отключение портов устройств, изменение параметров мостов адресных таблиц мостов, коммутаторов и маршрутизаторов и т. п. Примерами систем управления могут служить популярные системы HPOpenView, SunNetManager, IBMNetView.

Средства управления системой (SystemManagement). Средства управления системой часто выполняют функции, аналогичные функциям систем управления, но по отношению к другим объектам. В первом случае объектом управления является программное и аппаратное обеспечение компьютеров сети, а во втором — коммуникационное оборудование. Вместе с тем, некоторые функции этих двух видов систем управления могут дублироваться, например, средства управления системой могут выполнять простейший анализ сетевого трафика.

Встроенные системы диагностики и управления (Embeddedsystems). Эти системы выполняются в виде программно-аппаратных модулей, устанавливаемых в коммуникационное оборудование, а также в виде программных модулей, встроенных в операционные системы. Они выполняют функции диагностики и управления только одним устройством, и в этом их основное отличие от централизованных систем управления. Примером средств этого класса может служить модуль управления концентратором Distrebuted 5000, реализующий функции автосегментации портов при обнаружении неисправностей, приписывания портов внутренним сегментам концентратора и некоторые другие. Как правило, встроенные модули управления «по совместительству» выполняют роль SNMP-агентов, поставляющих данные о состоянии устройства для систем управления.

Анализаторы протоколов (Protocolanalyzers). Представляют собой программные или аппаратно-программные системы, которые ограничиваются в отличие от систем управления лишь функциями мониторинга и анализа трафика в сетях. Хороший анализатор протоколов может захватывать и декодировать пакеты большого количества протоколов, применяемых в сетях — обычно несколько десятков. Анализаторы протоколов позволяют установить некоторые логические условия для захвата отдельных пакетов и выполняют полное декодирование захваченных пакетов, то есть показывают в удобной для специалиста форме вложенность пакетов протоколов разных уровней друг в друга с расшифровкой содержания отдельных полей каждого пакета.

Анализатор протоколов представляет собой либо самостоятельное специализированное устройство, либо персональный компьютер, оснащенный специальной сетевой картой и соответствующим программным обеспечением. Применяемые сетевая карта и программное обеспечение должны соответствовать топологии сети (кольцо, шина, звезда). Анализатор подключается к сети точно также, как и обычный узел. Отличие состоит в том,

что анализатор может принимать все пакеты данных, передаваемые по сети, в то время как обычная станция — только адресованные ей. Программное обеспечение анализатора состоит из ядра, поддерживающего работу сетевого адаптера и декодирующего получаемые данные, и дополнительного программного кода, зависящего от типа топологии исследуемой сети. Кроме того, поставляется ряд процедур декодирования, ориентированных на определенный протокол, например, IPX. В состав некоторых анализаторов может входить также экспертная система, которая может выдавать пользователю рекомендации о том, какие эксперименты следует проводить в данной ситуации, что могут означать те или иные результаты измерений, как устранить некоторые виды неисправности сети.

Несмотря на относительное многообразие анализаторов протоколов, представленных на рынке, можно назвать некоторые черты, в той или иной мере присущие всем им:

- Пользовательский интерфейс. Большинство анализаторов имеют развитый дружественный интерфейс, базирующийся, как правило, на Windows или Motif. Этот интерфейс позволяет пользователю: выводить результаты анализа интенсивности трафика; получать мгновенную и усредненную статистическую оценку производительности сети; задавать определенные события и критические ситуации для отслеживания их возникновения; производить декодирование протоколов разного уровня и представлять в понятной форме содержимое пакетов.

- Буфер захвата. Буферы различных анализаторов отличаются по объему. Буфер может располагаться на устанавливаемой сетевой карте, либо для него может быть отведено место в оперативной памяти одного из компьютеров сети. Если буфер расположен на сетевой карте, то управление им осуществляется аппаратно, и за счет этого скорость ввода повышается. Однако это приводит к удорожанию анализатора. В случае недостаточной производительности процедуры захвата, часть информации будет теряться, и анализ будет невозможен. Размер буфера определяет возможности анализа по более или менее представительным выборкам захватываемых данных. Но каким бы большим ни был буфер захвата, рано или поздно он заполнится. В этом случае либо прекращается захват, либо заполнение начинается с начала буфера.

- Фильтры. Фильтры позволяют управлять процессом захвата данных, и, тем самым, позволяют экономить пространство буфера. В зависимости от значения определенных полей пакета, заданных в виде условия фильтрации, пакет либо игнорируется, либо записывается в буфер захвата. Использование фильтров значительно ускоряет и упрощает анализ, так как исключает просмотр ненужных в данный момент пакетов.

- Переключатели — это задаваемые оператором некоторые условия начала и прекращения процесса захвата данных из сети. Такими условиями могут быть выполнение ручных команд запуска и остановки процесса захвата, время суток, продолжительность процесса захвата, появление определенных значений в кадрах данных. Переключатели могут использоваться совместно с

фильтрами, позволяя более детально и тонко проводить анализ, а также продуктивнее использовать ограниченный объем буфера захвата.

- Поиск. Некоторые анализаторы протоколов позволяют автоматизировать просмотр информации, находящейся в буфере, и находить в ней данные по заданным критериям. В то время, как фильтры проверяют входной поток на предмет соответствия условиям фильтрации, функции поиска применяются к уже накопленным в буфере данным.

Методология проведения анализа может быть представлена в виде следующих шести этапов:

1. Захват данных.
2. Просмотр захваченных данных.
3. Анализ данных.
4. Поиск ошибок. (Большинство анализаторов облегчают эту работу, определяя типы ошибок и идентифицируя станцию, от которой пришел пакет с ошибкой.)
5. Исследование производительности. Рассчитывается коэффициент использования пропускной способности сети или среднее время реакции на запрос.
6. Подробное исследование отдельных участков сети. Содержание этого этапа конкретизируется по мере того, как проводится анализ.

Сетевые анализаторы

Сетевые анализаторы (не следует путать их с анализаторами протоколов) представляют собой эталонные измерительные инструменты для диагностики и сертификации кабелей и кабельных систем.

Сетевые анализаторы содержат высокоточный частотный генератор и узкополосный приемник. Передавая сигналы различных частот в передающую пару и измеряя сигнал в приемной паре, можно измерить затухание и др. Сетевые анализаторы — это прецизионные крупногабаритные и дорогие (стоимостью более $20’000) приборы, предназначенные для использования в лабораторных условиях специально обученным техническим персоналом.

Экспертные системы. Этот вид систем аккумулирует знания технических специалистов о выявлении причин аномальной работы сетй и возможных способах приведения сетей в работоспособное состояние. Экспертные системы часто реализуются в виде отдельных подсистем различныз средств мониторинга и анализа сетей: систем управления сетями, анализаторов протоколов, сетевых анализаторов. Простейшим вариантом экспертной системы является контекстно-зависимая help-система. Более сложные экспертные системы представляют собой так называемые базы знаний, обладающие элементами искусственного интеллекта. Примером такой системы является экспертная система, встроенная в систему управления Spectrum компании Cabletron.

2. ЛАБОРАТОРНЫЕ РАБОТЫ

2.1 Лабораторная работа 1

Изучение программных средств тестирования параметров соединения в компьютерных сетях и проверки настройки протокола TCP/IP.

Цель работы: Знакомство с программными средствами для тестирования параметров соединения в компьютерных сетях и проверки настройки протокола TCP/IP.

Ход работы:

Все команды и утлиты, которые будут приведены ниже используются в контексте Command Prompt ОС Windows (cmd).

•**Netstat.** Команда netstat отображает статистику активных подключений TCP, портов, прослушиваемых компьютером, статистики Ethernet, таблицы маршрутизации IP, статистики Ipv4 (для протоколов IP, ICMP, TCP и UDP) и Ipv6 (для протоколов Ipv6, ICMPv6, TCP через Ipv6 и UDP через Ipv6). Запущенная без параметров, команда netstat отображает подключения TCP.

Формат команды: **netstat [-a] [-e] [-n] [-o] [-p протокол] [-r] [-s] [интервал]**, где:

-a – вывод всех активных подключений TCP и прослушиваемых компьютером портов TCP и UDP.

-e – вывод статистики Ethernet, например количества отправленных и принятых байтов и пакетов. Этот параметр может комбинироваться с ключом –s.

-n – вывод активных подключений TCP с отображением адресов и номеров портов в числовом формате без попыток определения имен.

-o – вывод активных подключений TCP и включение кода процесса (PID) для каждого подключения. Код процесса позволяет найти приложение на вкладке Процессы диспетчера задач Windows. Этот параметр может комбинироваться с ключами –a, -n и –p.

-p протокол – вывод подключений для протокола, указанного параметром протокол. В этом случае параметр протокол может принимать значения tcp, udp, tcpv6 или udpv6. Если данный параметр используется с ключом –s для вывода статистики по протоколу, параметр протокол может иметь значение tcp, udp, icmp, ip, tcpv6, udpv6, icmpv6 или ipv6.

-s – вывод статистики по протоколу. По умолчанию выводится статистика для протоколов TCP, UDP, ICMP и IP. Если установлен протокол Ipv6 для Windows XP, отображается статистика для протоколов TCP через Ipv6, UDP через Ipv6, ICMPv6 и Ipv6. Параметр –p может использоваться для указания набора протоколов.

-r – вывод содержимого таблицы маршрутизации IP. Эта команда эквивалентна команде route print.

интервал – обновление выбранных данных с интервалом, определенным параметром интервал (в секундах). Нажатие клавиш CTRL+C останавливает обновление. Если этот параметр пропущен, netstat выводит выбранные данные только один раз.

/? – отображение справки в командной строке.

Задание:

•**Ping.** Ping — утилита командной строки для проверки соединений в сетях на основе TCP/IP. Команда PING с помощью отправки сообщений с эхо-запросом по протоколу ICMP проверяет соединение на уровне протокола IP с другим компьютером, поддерживающим TCP/IP. После каждой передачи выводится соответствующее сообщение с эхо-ответом.

Формат команды: **ping [-t] [-a] [-n счетчик] [-l размер] [-f] [-i TTL] [-v тип] [-r счетчик] [-s счетчик] [{-j список_узлов | -k список_узлов}] [-w интервал] [имя_конечного_компьютера]**

-t – Задает для команды ping отправку сообщений с эхо-запросом к точке назначения до тех пор, пока команда не будет прервана. Для прерывания команды и вывода статистики нажмите комбинацию CTRL-BREAK. Для прерывания команды ping и выйхода из нее нажмите клавиши CTRL-C.

-a – Задает разрешение обратного имени по IP-адресу назначения. В случае успешного выполнения выводится имя соответствующего узла.

-n счетчик – Задает число отправляемых сообщений с эхо-запросом. По умолчанию — 4.

-l размер – Задает длину (в байтах) поля данных в отправленных сообщениях с эхо-запросом. По умолчанию — 32 байта. Максимальный размер — 65527.

-f – Задает отправку сообщений с эхо-запросом с флагом «Don't Fragment» в IP-заголовке, установленном на 1. Сообщения с эхо-запросом не фрагментируются маршрутизаторами на пути к месту назначения. Этот параметр полезен для устранения проблем, возникающих с максимальным блоком данных для канала (Maximum Transmission Unit).

-i TTL – Задает значение поля TTL в IP-заголовке для отправляемых сообщений с эхо-запросом. По умолчанию берется значение TTL, заданное по умолчанию для узла. Для узлов Windows XP это значение обычно равно 128. Максимальное значение TTL — 255.

-v тип – Задает значение поля типа службы (TOS) в IP-заголовке для отправляемых сообщений с эхо-запросом. По умолчанию это значение равно 0. тип — это десятичное значение от 0 до 255.

-r счетчик – Задает параметр записи маршрута (Record Route) в IP-заголовке для записи пути, по которому проходит сообщение с эхо-запросом и соответствующее ему сообщение с эхо-ответом. Каждый переход в пути использует параметр записи маршрута. По возможности значение счетчика задается равным или большим, чем количество переходов между источником и местом назначения. Параметр счетчик имеет значение от 1 до 9.

-s счетчик – Указывает вариант штампа времени Интернета (Internet Timestamp) в заголовке IP для записи времени прибытия сообщения с эхо-

запросом и соответствующего ему сообщения с эхо-ответом для каждого перехода. Параметр счетчик имеет значение от 1 до 4.

-j список_узлов – Указывает для сообщений с эхо-запросом использование параметра свободной маршрутизации в IP-заголовке с набором промежуточных точек назначения, указанным в списке_узлов. При свободной маршрутизации последовательные промежуточные точки назначения могут быть разделены одним или несколькими маршрутизаторами. Максимальное число адресов или имен в списке узлов — 9. Список узлов — это набор IP-адресов (в точечно-десятичной нотации), разделенных пробелами.

-k список_узлов – Указывает для сообщений с эхо-запросом использование параметра строгой маршрутизации в IP-заголовке с набором промежуточных точек назначения, указанным в списке_узлов. При строгой маршрутизации следующая промежуточная точка назначения должна быть доступной напрямую (она должна быть соседней в интерфейсе маршрутизатора). Максимальное число адресов или имен в списке узлов равно 9. Список узлов — это набор IP-адресов (в точечно-десятичной нотации), разделенных пробелами.

-w интервал – Определяет в миллисекундах время ожидания получения сообщения с эхо-ответом, которое соответствует сообщению с эхо-запросом. Если сообщение с эхо-ответом не получено в пределах заданного интервала, то выдается сообщение об ошибке «Request timed out». Интервал по умолчанию равен 4000 (4 секунды).

имя_конечного_компьютера – Задает точку назначения, идентифицированную IP-адресом или именем узла.

Tracert. Команда TRACERT определяет путь до точки назначения с помощью посылки в точку назначения эхо-сообщений протокола Control Message Protocol (ICMP) с постоянным увеличением значений срока жизни (Time to Live, TTL). Выведенный путь — это список ближайших интерфейсов маршрутизаторов, находящихся на пути между узлом источника и точкой назначения. Ближний интерфейс представляют собой интерфейс маршрутизатора, который является ближайшим к узлу отправителя на пути. Запущенная без параметров, команда tracert выводит справку.

Формат команды: **tracert [-d] [-h максимальное_число_переходов] [-j список_узлов] [-w интервал [имя_конечного_компьютера].**

-d – Предотвращает попытки команды tracert разрешения IP-адресов промежуточных маршрутизаторов в имена. Увеличивает скорость вывода результатов команды tracert.

-h максимальное_число_переходов – Задает максимальное количество переходов на пути при поиске конечного объекта. Значение по умолчанию равно 30.

-j список_узов – Указывает для сообщений с эхо-запросом использование параметра свободной маршрутизации в заголовке IP с набором промежуточных мест назначения, указанных в списке_узлов. При свободной маршрутизации успешные промежуточные места назначения могут быть разделены одним или несколькими маршрутизаторами. Максимальное число

адресов или имен в списке — 9. Список_адресов представляет набор IP-адресов (в точечно-десятичной нотации), разделенных пробелами.

-w интервал – Определяет в миллисекундах время ожидания для получения эхо-ответов протокола ICMP или ICMP-сообщений об истечении времени, соответствующих данному сообщению эхо-запроса. Если сообщение не получено в течение заданного времени, выводится звездочка (*). Таймаут по умолчанию 4000 (4 секунды).

- имя_конечного_компьютера – задает точку назначения, указанную IP-адресом или именем узла.

-? – Отображает справку в командной строке по утилите tracert.

Содержание отчета:

1. Титульный лист.
2. Цель работы.
3. Результаты выполнения всех команд.
4. Выводы.

Контрольные вопросы:

1. Какие утилиты можно использовать для проверки правильности конфигурирования TCP/IP?
2. Каким образом команда ping проверяет соединение с узлом сети? Отметьте возможные причины, по которым ping не может связаться с удаленным хостом.
3. Что такое хост?
4. Что такое петля обратной связи?
5. Сколько промежуточных маршрутизаторов сможет пройти IP-пакет, если его время жизни равно 30?
6. Как работает утилита tracert?
7. Каково назначение протокола ARP?

2.2. Лабораторная работа 2

Ознакомление с интерфейсом программы NetEmul. Соединение ЭВМ в сеть.

Цель работы: Ознакомиться с основами работы с программным эмулятором ЛВС NetEmul, освоить основы логического моделирования компьютерной сети.

Ход работы:

Для запуска эмулятора NetEmul необходимо либо воспользоваться соответствующим пунктом главного меню операционной системы, либо выполнить в терминале команду netemul.

Соединение двух ЭВМ напрямую

Добавить на рабочее поле эмулятора два компьютера (см. рис. 2.1), использовав кнопку «Добавить компьютер» на панели инструментов.

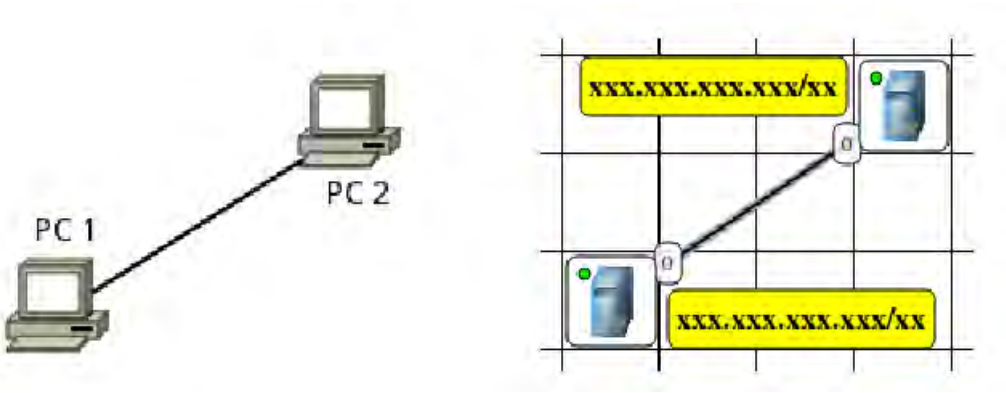


Рисунок 2.1. – Соединение двух ЭВМ напрямую.

Соединить добавленные компьютеры как показано на рис. 2.1. Для этого:

а) нажать кнопку «Создать соединение» на панели инструментов;

б) навести указатель на один из компьютеров;

в) зажав ЛКМ, перевести курсор на второй компьютер — за курсором от первого компьютера должна тянуться прямая линия;

г) отпустить ЛКМ — после этого должно появиться окно начальных настроек с выбором соединяемых интерфейсов;

д) подтвердить соединение между интерфейсами eth0 и eth0, нажав «Соединить»;

е) если все сделано правильно, то компьютеры теперь соединены, на каждом конце соединения показан номер используемого интерфейса (в данном случае — 0), а индикатор соединения на иконке компьютера сменил цвет с красного на желтый (соединение есть, но интерфейсы не настроены).

Настроить компьютеры, задав каждому IP-адрес и маску подсети в соответствии с вариантом. Для этого

а) выбрать инструмент «Перемещение объектов» на панели инструментов;

б) выделить первый компьютер щелчком ЛКМ;

в) вызвать контекстное меню щелчком ПКМ и выбрать пункт «Интерфейсы»;

г) в появившемся окне указать в соответствующих полях IP-адрес и маску подсети;

д) подтвердить ввод последовательным нажатием кнопок «Применить» и «ОК»;

е) если все сделано правильно, то индикатор соединения на иконке компьютера должен сменить цвет с желтого на зеленый (соединение есть, и интерфейсы настроены);

ж) добавить возле каждого компьютера надпись с его IP-адресом имаской подсети как показано на рис. 2.1.

Проверить работоспособность построенной модели ЛВС, передав пакеты от одного компьютера до другого. Для этого необходимо:

а) выбрать инструмент «Отправить данные» на панели инструментов;

б) под курсором (на рабочем поле программы) должен появиться красный круг;

в) навести курсор с красным кругом на передающий компьютер и нажать ЛКМ;

г) в появившемся окне «Отправка» указать: протокол TCP, размер данных 5 KB;

д) нажать «Далее» — окно пропадет, а кружок под курсором сменит цвет на зеленый;

е) навести курсор с зеленым кругом на принимающий компьютер нажать ЛКМ;

ж) в появившемся окне подтвердить интерфейс на принимающем компьютере eth0, нажав «Отправка»;

з) проследить за перемещением пакетов.

•Построение ЛВС на концентраторах

Добавить на рабочее поле эмулятора шесть компьютеров и три концентратора как показано на рис. 2.2. Соединить устройства как показано на рис. 2.2. Добавить возле каждого компьютера надпись с его IP-адресом и маской подсети. Проверить работоспособность построенной модели ЛВС, передав пакеты (TCP, 5 KB) от одного компьютера до другого. Проследить за перемещением пакетов и сделать выводы об особенностях работы ЛВС на основе концентраторов.

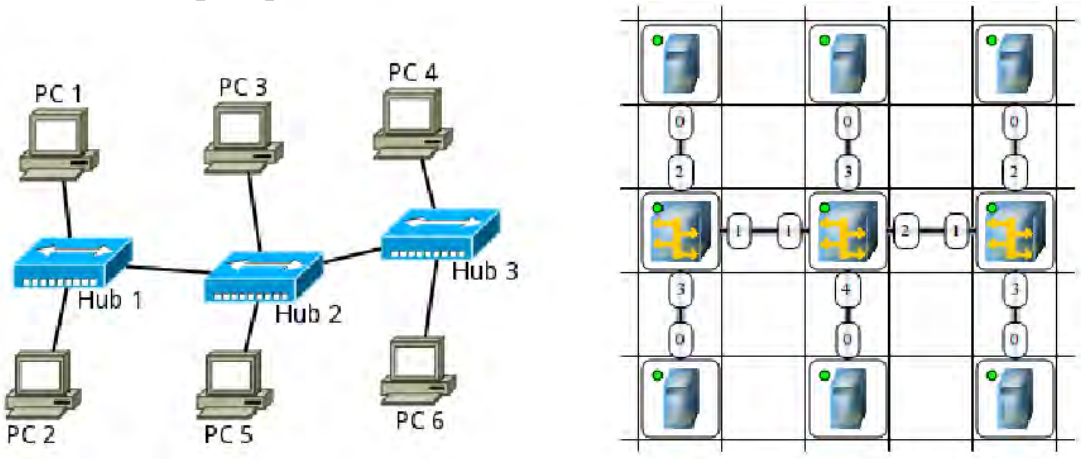


Рисунок 2.2. – Построение ЛВС на концентраторах

Построение ЛВС на коммутаторах

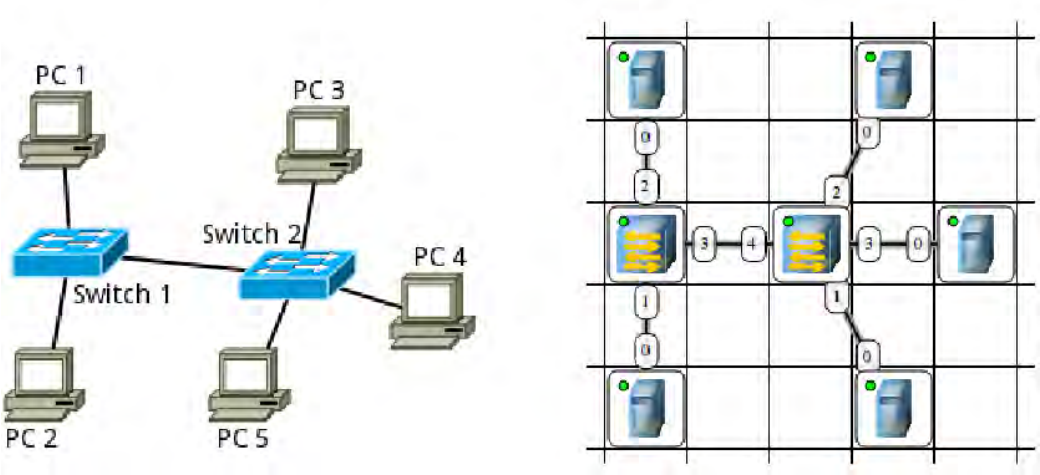


Рисунок 2.3. – Построение ЛВС на коммутаторах

Добавить на рабочее поле эмулятора пять компьютеров и два коммутатора как показано на рис. 2.3.

Соединить устройства как показано на рис. 2.3.

Настроить компьютеры, задав каждому IP-адрес и маску подсети в соответствии с вариантом.

Добавить возле каждого компьютера надпись с его IP-адресом и маской подсети. Проверить работоспособность построенной модели ЛВС, передав пакеты (TCP, 5 KB) от одного компьютера до другого. Проследить за перемещением пакетов и сделать выводы об особенностях работы ЛВС на основе коммутаторов.

Содержание отчета:

1. Титульный лист.
2. Цель работы.
3. Результаты выполнения всех команд.
4. Выводы.

Контрольные вопросы:

1. Какие сетевые устройства применяются для создания компьютерной сети?
2. Какие настройки необходимы для прямого соединения двух компьютеров по сети?
3. напишите операции при обмене пакетами между компьютерами.
4. Перечислите особенности передачи информации при организации сети на базе концентраторов.
5. Перечислите особенности передачи информации при организации сети на базе коммутаторов.

2.3. Лабораторная работа 3
Маршрутизация в NetEmul.

Цель работы: Ознакомиться с работой маршрутизаторов.

Задача: Научиться формировать статические маршруты и прописывать их в таблицы маршрутизации сетевых устройств.

Ход работы:

С помощью инструмента «Вставить текстовую надпись» добавить на рабочее поле эмулятора надпись, содержащую номер группы.

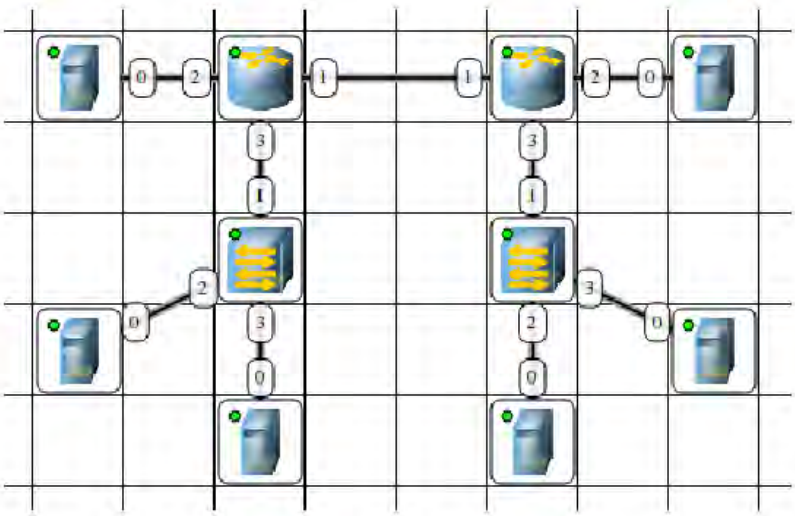


Рисунок 2.4. – Модель ЛВС

Используя соответствующие инструменты на панели эмулятора, построить сеть в соответствии с рис. 2.4. В свойствах каждого маршрутизатора необходимо указать количество интерфейсов, равное 4. Настроить интерфейсы компьютеров и маршрутизаторов, задав каждому IP-адрес и маску подсети в соответствии с вариантом. Добавить возле каждого компьютера и интерфейса роутера надписи с их IP-адресом и маской подсети. Проверить работоспособность построенной модели ЛВС, передав пакеты (TCP, 5 KB) от одного устройства до другого в пределах одной подсети.

Формирование таблицы статической маршрутизации.

В таблице маршрутизации роутера задать на каждом компьютере маршрут «по умолчанию» (IP сети = 0.0.0.0; маска подсети = 0.0.0.0). Задать на каждом маршрутизаторе статические маршруты до удалённых от него сетей. Для этого в полях сеть назначения и маска указать сеть назначения маршрута (Например, 192.168.1.0 и маска 255.255.255.0). В поле шлюз указать шлюз сети назначения, а в поле интерфейс выбирается интерфейс, через который пакеты будут уходить в сеть назначения.

Проверить работоспособность построенной модели ЛВС, передав пакеты (TCP и UDP, 5 KB) между удаленными друг от друга сетями. Проследить за

перемещением пакетов и сделать выводы об особенностях работы ЛВС на основе маршрутизаторов.

Содержание отчета:

1. Титульный лист.
2. Цель работы.
3. По каждому пункту лабораторной должна быть приведена схема модели с указанием IP-адресов устройств и номеров интерфейсов.
4. По каждому пункту лабораторной должны быть приведены выводы по работе.

Контрольные вопросы:

1. Что такое IP-адрес?
2. Что такое маска подсети?
3. Как работает маршрутизатор?
4. Принципы статической маршрутизации?

2.4. Лабораторная работа 4

Разрешение адресов по протоколу ARP.

Цель работы: Ознакомиться с механизмом работы протокола ARP.

Задачи: Научиться формировать и отправлять пользовательские пакеты. Ознакомиться с журналом работы сетевого устройства в эмуляторе. Научиться проводить сетевую атаку вида ARP-спуфинг.

Ход работы:

ARP (Address Resolution Protocol — протокол определения адреса) — протокол в компьютерных сетях, предназначенный для определения MAC-адреса сетевого устройства по известному IP-адресу.

Наибольшее распространение ARP получил благодаря повсеместности сетей IP, построенных поверх Ethernet, поскольку в подавляющем большинстве случаев при таком сочетании используется ARP. В семействе протоколов Ipv6 протокола ARP не существует, его функции возложены на ICMPv6. Описание протокола было опубликовано в ноябре 1982 г. в RFC 826.

ARP был спроектирован для случая передачи IP-пакетов через сегмент Ethernet. При этом общий принцип, предложенный для ARP, был использован и для сетей других типов.

Существуют следующие типы сообщений ARP: запрос ARP (ARP-request) и ответ ARP (ARP-reply).

Система-отправитель при помощи запроса ARP запрашивает физический адрес системы-получателя. Ответ (физический адрес узла-получателя) приходит в виде ответа ARP. Принцип работы протокола: узел (хост A), которому нужно выполнить отображение IP-адреса на MAC-адрес, формирует

ARP-запрос, вкладывает его в кадр протокола канального уровня, указывая в нем известный IP-адрес (хост B), и рассылает запрос широковещательно (в поле MAC-адрес назначения заголовка Ethernet указывается широковещательный MAC-адрес FF:FF:FF:FF:FF:FF).

Все узлы локальной сети получают ARP-запрос и сравнивают указанный там IP-адрес с собственным. В случае их совпадения узел (хост B) формирует ARP-ответ, в котором указывает свой IP-адрес и свой локальный адрес и отправляет его уже направленно, так как в ARP запросе отправитель (хост A) указывает свой локальный адрес.

Схема работы показана на рисунке 2.5.

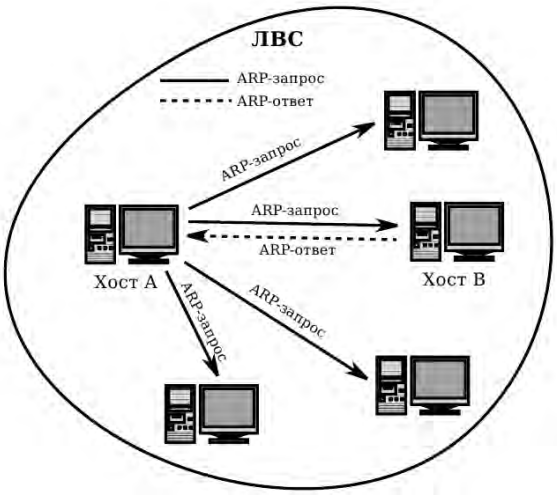


Рисунок 2.5 – Схема работы протокола ARP

При получении ARP-ответа хост A записывает в кэш ARP запись с соответствием IP-адреса хоста B и MAC-адреса хоста B, полученного из ARP-ответа. Время хранения такой записи ограничено. По истечении времени хранения хост A посылает повторный запрос, теперь уже адресно, на известный MAC-адрес хоста B. В случае, если ответ не получен, снова посылается широковещательный запрос.

Структура кадра ARP с учетом заголовка Ethernet показана на рисунке 2.6.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Destination MAC						Source MAC						ETH TYPE		HTYPE	
PTYPE		HLEN	PLEN	OP CODE		Sender MAC						Sender IP			
Target MAC						Target IP									

Рисунок 2.6 – Структура кадра ARP

Самопроизвольный ARP (gratuitous ARP) — такое поведение ARP, когда ARP-ответ присылается, когда в этом (с точки зрения получателя) нет особой необходимости. Самопроизвольный ARP-ответ — это пакет-ответ ARP,

присланный без запроса. Он применяется для определения конфликтов IP-адресов в сети: как только станция получает адрес по DHCP или адрес присваивается вручную, рассылается ARP-ответ gratuitous ARP.

Самопроизвольный ARP может быть полезен в следующих случаях:

• обновление ARP-таблиц, в частности, в кластерных системах;

• информирование коммутаторов;

• извещение о включении сетевого интерфейса.

Несмотря на эффективность самопроизвольного ARP, он является особенно небезопасным, поскольку с его помощью можно уверить удаленный узел в том, что MAC-адрес какой-либо системы, находящейся с ней в одной сети, изменился, и указать, какой адрес используется теперь.

Сетевая атака ARP-спуфинг (ARP-spoofing) основана на использовании самопроизвольного ARP. Чтобы перехватить сетевые пакеты, которые атакуемый хост (A) оправляет на хост B, атакующий хост (C) формирует ARP-ответ, в котором ставит в соответствие IP-адресу хоста B свой MAC-адрес. Далее этот пакет отправляется на хост A. В том случае, если хост A поддерживает самопроизвольный ARP, он модифицирует собственную ARP-таблицу и помещает туда запись, где вместо настоящего MAC-адреса хоста B стоит MAC-адрес атакующего хоста C.

Теперь пакеты, отправляемые хостом A на хост B, будут передаваться хосту C.

Построение сети.

1.Постройте сеть, отображенную на рисунке 2.7.

2. Используя соответствующие инструменты на панели эмулятора, построить сеть в соответствии с рис. 2.7. В свойствах маршрутизатора необходимо указать количество интерфейсов, равное 2.

3. Настроить интерфейсы компьютеров и маршрутизаторов, задав каждому IP-адрес и маску подсети (слева — первая подсеть в заданной сети, справа — вторая подсеть). Добавить возле каждого компьютера и интерфейса роутера надписи с их IP-адресом и маской подсети.

4. Настроить на компьютерах маршруты «по умолчанию» (IP сети = 0.0.0.0; маска подсети = 0.0.0.0). Можно воспользоваться «Таблицей маршрутизации» либо вызвать свойства компьютера двойным щелчком, указать шлюз по умолчанию и включить маршрутизацию.

5. Включить маршрутизацию на маршрутизаторе.

6. Проверить работоспособность построенной модели ЛВС, передав пакеты (TCP, 5 KB) от компьютера в левой подсети до компьютера в правой подсети. Если пакеты не проходят, в меню “Интерфейсы” маршрутизатора нажать кнопку “сбросить статистику” для автоматического формирования ARP запроса.

7. Задать каждому компьютеру имя-описание, воспользовавшись пунктом контекстного меню «Задать описание».

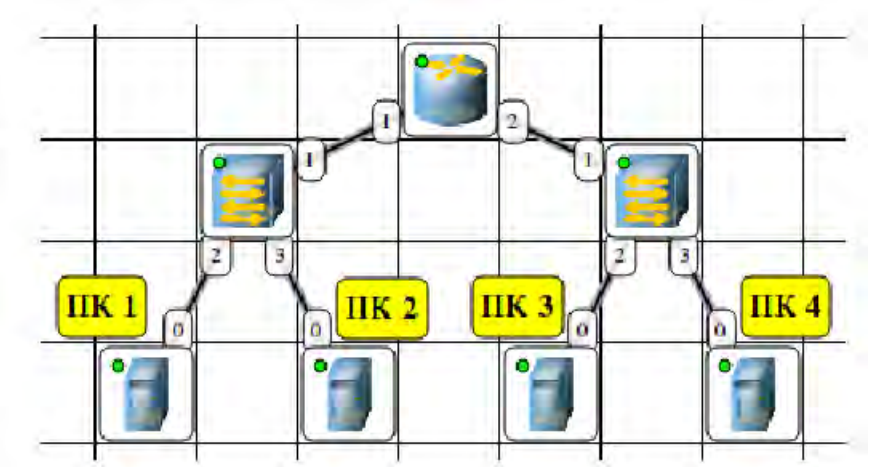


Рисунок 2.7 – Структура ЛВС для ознакомления с ARP протоколом

Определение MAC-адреса с помощью ARP-запроса.

1.Запустить для компьютеров 1 и 2 журналы пакетов (пункт меню «Показать журнал»).

2. Очистить ARP-таблицу компьютера 1.

3. Выделить компьютер 1 и с помощью инструмента «Конструктор пакетов» сформировать пакет ARP-запроса для определения MAC-адреса компьютера 2. Помните, что ARP-запрос рассылается широковещательно (MAC-адрес получателя в заголовке Ethernet — FF:FF:FF:FF:FF:FF), а MAC-адрес искомого узла в заголовке ARP приравнивается к нулевому 00:00:00:00:00:00. MAC-адрес компьютера 1 указан в окне «Интерфейсы» для компьютера 1.

4. Запустить ARP-запрос, проследить за ним и за сгенерированным для него ARP-ответом по схеме сети и журналам компьютеров 1 и 2.

5. Открыть ARP-таблицу компьютера 1 и убедиться, что запись добавилась в таблицу.

6. Сохранить скриншот экрана (с открытыми журналами) для отчета.

Реализация атаки ARP-спуфинг.

1. Запустить для компьютеров 1 и 2 журналы пакетов (пункт меню «Пказать журнал»). При необходимости очистить их.

2. Очистить ARP-таблицу компьютера 1.

3. Выделить компьютер 2 и с помощью инструмента «Конструктор пакетов» сформировать пакет ARP-ответа, в котором будут указаны:

• MAC отправителя — MAC компьютера 2;

• IP отправителя — IP интерфейса роутера в левой подсети;

• MAC получателя — MAC компьютера 1;

• IP получателя — IP компьютера 1.

4. Запустить ARP-ответ, проследить за ним. Может возникнуть окно о дублировании IP-адресов в сети — это происходит в том случае, если из-за действий коммутатора пакет-атаку получает и роутер. Окно быстро закрыть.

5. Сразу же запустить передачу пакетов (UDP, 5 KB) от компьютера 1 на компьютер 3. Убедиться, что пакеты вначале приходят на компьютер 2 и лишь потом (если на компьютере 2 включена маршрутизация) отправляются на компьютер 3 (через маршрутизатор).

Содержание отчета:

1. Титульный лист.
2. Цель работы.
3. Разбиение заданной сети /27 на две подсети /28.
4. Схема модели с указанием IP-адресов устройств и номеров интерфейсов.
5. Скриншоты с результатами разрешения адреса и сетевой атаки.
6. По каждому пункту лабораторной должны быть приведены выводы по работе.

Контрольные вопросы:

1. Протокол ARP.
2. Формат пакета ARP.
3. Самопроизвольный ARP.
4. IP-адрес.
5. MAC-адрес.
6. ARP-спуфинг.

2.5. Лабораторная работа 5

Динамическая маршрутизация по протоколу RIP. Получение сетевых настроек по DHCP.

Цель работы: Ознакомиться с механизмом динамической маршрутизации по протоколу RIP. Научиться настраивать компьютеры и серверы для автоматизации получения компьютерами сетевых настроек.

Ход работы:

Используя соответствующие инструменты на панели эмулятора, построить сеть в соответствии с рис. 2.8.

Распределить полученные ранее адреса сетей между сетями SR1–SR5 и SH11–SH13. Добавить возле каждой сети надпись с ее IP-адресом. Настроить интерфейсы маршрутизаторов, задав каждому IP-адрес и маску подсети в соответствии с выбранным распределением.

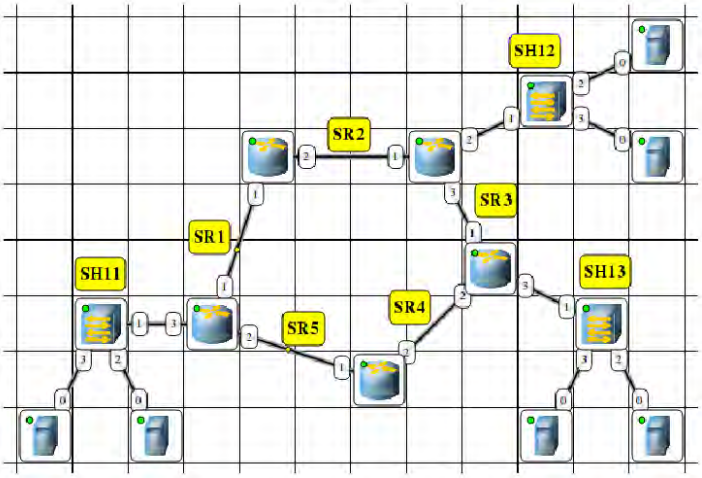


Рисунок 2.8. Структура сети для знакомства с протоколом RIP

Настройка динамической маршрутизации по протоколу RIP.

1. На каждом маршрутизаторе добавить и запустить программу RIP. Пункт контекстного меню «Программы». Кнопка «Добавить». Не забудьте поставить флаг для активации программы.
2. Включить маршрутизацию на маршрутизаторе.
3. Открыть журнал одного из маршрутизаторов. Проследить за перемещением пакетов протокола RIP по сети.
4. Поочередно открыть таблицы маршрутизации каждого маршрутизатора и убедиться, что таблица заполнилась.

Настройка автоматического получения сетевых настроек по протоколу DHCP.

1. На маршрутизаторах, которые отвечают за сети SH11–SH13 добавить и запустить программу DHCP-сервер. Не забудьте поставить флаг для активации программы.
2. В настройках каждого DHCP-сервера указать интерфейс, «смотрящий» в сторону сети SH, тип адресов — динамические, диапазон адресов, выделяемых для динамической адресации, маску подсети и IP-адрес шлюза.
3. На каждом компьютере добавить и запустить программу DHCP-клиент. Не забудьте поставить флаг для активации программы.
4. В настройках каждого DHCP-клиента укажите интерфейс, который должен автоматически получать сетевые настройки.
5. Открыть диалог настройки интерфейсов каждого компьютера и убедиться, что стоит флаг «Получать настройки автоматически».
6. Дождаться, пока все компьютеры не получат сетевые настройки.
7. Проверить работоспособность построенной модели ЛВС, передав пакеты (TCP, 5 KB) между компьютерами в разных подсетях.

Содержание отчета:

1. Титульный лист.
2. Цель работы.
3. Схема модели с указанием IP-адресов устройств и номеров интерфейсов.

4. По каждому пункту лабораторной должны быть приведены выводы по работе.

Контрольные вопросы:

1. Протокол RIP.
2. Протокол DHCP.

2.6. Лабораторная работа 6

Преобразование десятичных чисел в двоичные и двоичных в десятичные.

Цель работы: Освоить методики преобразования десятичных чисел в двоичные и наоборот.

Ход работы:

На рисунке 2.9 наглядно изображен способ преобразования чисел из двоичной в десятичную систему счисления и из десятичной в двоичную. Заполните таблицу 2.1, чтобы попрактиковаться в преобразовании десятичных чисел в двоичные.

Преобразование десятичных в двоичные

Основание 2	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
Десятичное	128	64	32	16	8	4	2	1	Двоичное
48	0	0	1	1	0	0	0	0	48 = 32 + 16 = 00110000

Преобразование двоичных в десятичные

Основание 2	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
Десятичное	128	64	32	16	8	4	2	1	Десятичное
11001100	1	1	0	0	1	1	0	0	128 + 64 + 8 + 4 = 204

Рисунок 2.9. Методика преобразования

Таблица 2.1

Основание 2	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
Десятичное число	128	64	32	16	8	4	2	1	Двоичное число
48	0	0	1	1	0	0	0	0	48 = 32+16 = 00110000
146	1	0	0	1					
222									
119									

135									
60									

Заполните таблицу 2.2, чтобы попрактиковаться в преобразовании двоичные чисел в десятичные.

Таблица2.2

Основание 2	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
Двоичное число	128	64	32	16	8	4	2	1	Десятичное число
11001100	1	1	0	0	1	1	0	0	128 + 64 + 8 + 4 = 204
10101010	1	0	1	0					
11100011									
10110011									
00110101									
10010111									

Содержание отчета:

1. Титульный лист.
2. Цель работы.
3. Заполненные таблицы.

Контрольные вопросы:

Методика преобразования чисел: из десятичной системы счисления в двоичную, из двоичной системы счисления в десятичную.

2.7. Лабораторная работа 7

Классификация способов сетевой адресации.

Цель работы: Освоить навыки сетевой адресации.

Ход работы:

Дополните таблицу 2.3.

Таблица 2.3

	Десятичный IP-адрес	Класс адреса	Количество бит в идентификаторе сети	Максимальное количество узлов (2П-2)
10010001.00100000.00111011.00011000	145.32.59.24	Класс B	16	
11001000.00101010.10000001.00010000	200.42.129.16			

Преобразование IP-адреса в десятичном формате в двоичный формат

Заполните таблицу 2.4, чтобы представить адрес 200.42.129.16 в двоичном формате.

Таблица 2.4

Основание 2	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
Десятичное число	128	64	32	16	8	4	2	1	Двоичное число
200	1	1	0	0	1	1	0	0	
42	1	0	1	0					
129									
16									

IP-адрес в двоичном формате.

Заполните таблицу 2.5, чтобы представить адрес 14.82.19.54 в двоичном формате.

Таблица 2.5

Основание 2	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
Десятичное число	128	64	32	16	8	4	2	1	Двоичное число
14									
82									
19									
54									

Преобразование IP-адреса в двоичном формате в десятичный формат

Заполните следующую таблицу 2.6, чтобы представить IP-адрес 11011000.00011011.00111101.10001001 в десятичном формате.

Таблица 2.6

Основание 2	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
Двоичное число	128	64	32	16	8	4	2	1	Десятичное число
11011000									
00011011									
00111101									
10001001									

Содержание отчета:

1. Титульный лист.
2. Цель работы.
3. Заполненные таблицы.

Контрольные вопросы:

1. Преобразование IP адресов из двоичного формата в десятичный.

2.8. Лабораторная работа 8

Вычисление масок подсети.

Цель работы: Приобрести навыки вычисления подсетей.

Ход работы:

Определение количества доступных сетевых адресов

Для сети класса А на основе указанного числа бит сети заполните таблицу 2.7, чтобы определить маску подсети к количество возможных адресов хостов для каждой маски.

Таблица 2.7

Классовый адрес	Десятичная маска подсети	Двоичная маска подсети	Количество хостов для подсети ($2^n - 2$)
/20			
/21			
/22			
/23			
/24			
/25			
/26			
/27			
/28			
/29			
/30			

Определение подсетей для сетевого адреса

Предположим, что вам выделена сеть 172.25.0.0.16. Необходимо создать двенадцать подсетей. Ответьте на следующие вопросы (Таблица 2.8).

Таблица 2.8

Действие	Описание
1.	Укажите разделяемый октет в двоичном формате.
2.	Укажите маску или длину классового префикса в двоичном формате.
3.	Отделите линией значимые биты в назначенном IP-адресе. Разделите линией маску, чтобы выделить значимые биты IP-адреса.
4.	Скопируйте значимые биты четыре раза.
5.	В первой строке укажите сетевой адрес, поставив 0 в оставшиеся биты хоста.
6.	В последней строке укажите широковещательный адрес, поставив 1 в битах хоста.
7.	В средних строках укажите идентификатор первого и последнего хостов подсети.
8.	Чтобы определить следующий адрес подсети, увеличивайте биты подсети на единицу. Повторите шаги с 4 по 8 для всех подсетей.

1. Сколько бит потребуется позаимствовать для задания 12 подсетей?

2. Укажите классовый адрес и маску подсети в двоичном и десятичном формате, которые позволят создать 12 подсетей.

3. Используйте метод, включающий восемь действий, чтобы задать 12 подсетей.

Заполните таблицу 2.9.

Таблица 2.9

Номер подсети	Адрес подсети	Диапазон адресов хостов	Широковещательный адрес
0			
1			
2			
3			
4			
5			
6			
7			

Определение подсетей на основе другого сетевого адреса

Предположим, что вам выделена сеть 192.168.1.0.24.

1. Сколько бит потребуется позаимствовать для задания 6 подсетей?

2. Укажите классовый адрес и маску подсети в двоичном и десятичном формате, которые позволят создать 6 подсетей.

3. Используйте метод, включающий восемь действий, чтобы задать 6 подсетей (Таблица 2.10).

Таблица 2.10

Действие	Описание
1.	Укажите разделяемый октет в двоичном формате.
2.	Укажите маску или длину классового префикса в двоичном формате.
3.	Отделите линией значимые биты в назначенном IP-адресе. Разделите линией маску, чтобы выделить значимые биты IP-адреса.
4.	Скопируйте значимые биты четыре раза.
5.	В первой строке укажите сетевой адрес, поставив 0 в оставшиеся биты хоста.
6.	В последней строке укажите широковещательный адрес, поставив 1 в битах хоста.
7.	В средних строках укажите идентификатор первого и последнего хостов подсети.
8.	Чтобы определить следующий адрес подсети, увеличивайте биты подсети на единицу. Повторите шаги с 4 по 8 для всех подсетей.

Заполните таблицу 2.11, чтобы задать каждую из подсетей.

Таблица 2.11

Номер подсети	Адрес подсети	Диапазон адресов хостов	Широковещательный адрес
0			
1			
2			
3			
4			
5			
6			
7			

Содержание отчета:

1. Титульный лист.
2. Цель работы.
3. Заполненные таблицы.
4. По каждому пункту лабораторной должны быть приведены выводы по работе.

Контрольные вопросы:

1. Методика назначения подсетей на основе другого сетевого адреса и сетевого адреса с классовым адресом.

2.9. Лабораторная работа 9

Знакомство с сетевым симулятором Cisco Packet Tracer.

Цель работы: Познакомиться с средой проектирования сетей

Ход работы: Знакомимся с главным окном программы (рис 2.10)

В нижней части окна программы расположены устройства, подключаемые к сети (рис 2.11). Маршрутизаторы (роутеры) используется для поиска оптимального маршрута передачи данных на основании специальных алгоритмов маршрутизации, например, выбор маршрута (пути) с наименьшим числом транзитных узлов.

Коммутаторы – это устройства, работающие на канальном уровне модели OSI и предназначенные для объединения нескольких узлов в пределах одного или нескольких сегментах сети. Передаёт пакеты коммутатор на основании внутренней таблицы – таблицы коммутации, следовательно, трафик идёт только на тот MAC-адрес, которому он предназначается, а не повторяется на всех портах (как на концентраторе).

Концентраторы. Это менее интеллектуальный вариант устройства, объединяющего сетевые узлы.

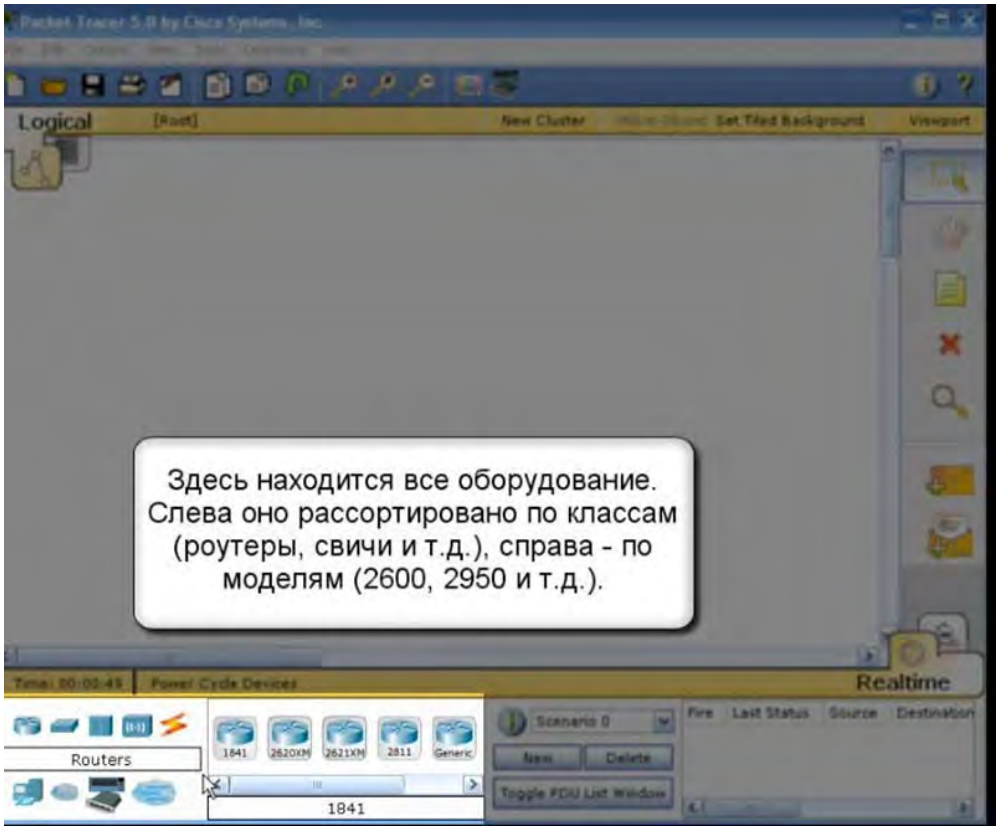


Рисунок 2.10. Главное окно

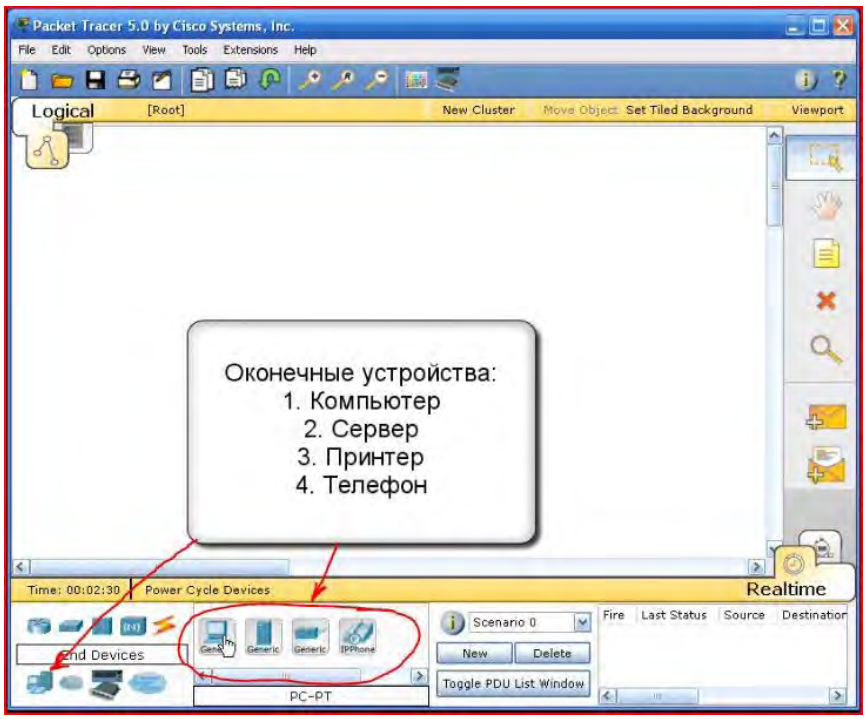


Рисунок 2.11. Доступное оборудование

Он просто повторяет пакет, принятый на одном порту на всех остальных портах. Всё по технологии Ethernet. В настоящее время выпускаются очень редко. Никакой защиты. Его можно сравнить с «тройником» как для силовой сети.

Рисунок 2.12. Пользовательские устройства и облако для многопользовательской работы

Кастомные девайсы, которые можно комплектовать самостоятельно и сохранять для последующей работы. Ну и создание произвольного подключения, к которой мы обязательно вернёмся и рассмотрим подробней, когда будем касаться интеграции с реальной сетью.

Или если представить всю информацию компактно получим окно (Рис. 2.13).

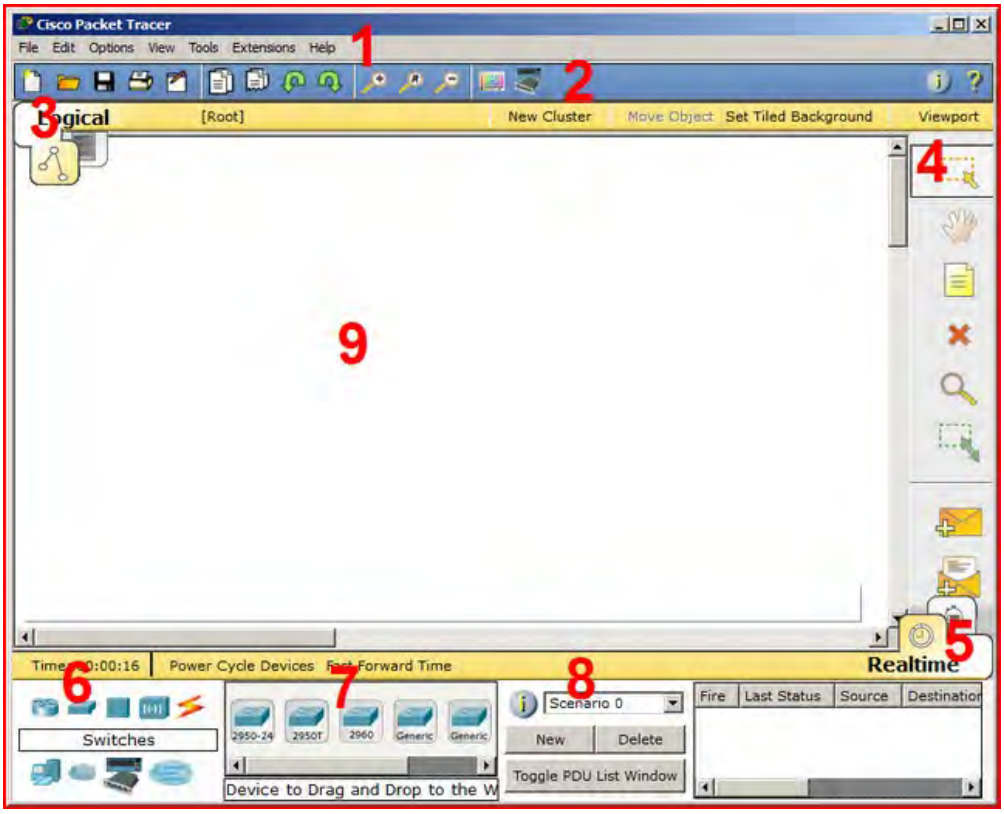


Рисунок 2.13. Главное меню программы

1. Панель, содержащая следующие вкладки:

• Файл – содержит операции открытия/сохранения документов;

•Правка-стандартные операции «копировать/вырезать, отменить/повторить».

• Настройки – говорит само за себя.

• Вид – масштаб рабочей области и панели инструментов

• Инструменты – цветовая палитра и кастомизация конечных устройств.

• Расширения – мастер проектов, многопользовательский режим и несколько шаблонов, которые из CPT (так называют Cisco Packet Tracer), которые могут сделать целую лабораторию.

• Помощь.

2. Панель инструментов, часть которых просто дублирует пункты меню;
3. Переключаетль между логической и физической организацией.
4. Ещё одна панель инструментов, содержит инструменты выделения, удаления, перемещения, масштабирования объектов, а так же формирование произвольных пакетов.
5. Переключатель между реальным режимом (Real-Time) и режимом симуляции.
6. Панель с группами конечных устройств и линий связи.
7. Сами конечные устройства, здесь содержатся всевозможные коммутаторы, узлы, точки доступа, проводники. Как детальки для конструктора (Drag and Drop).
8. Панель создания пользовательских сценариев.

9. Рабочее пространство.

Ниже представлен пример размещения цветовых областей (рис 2.14).

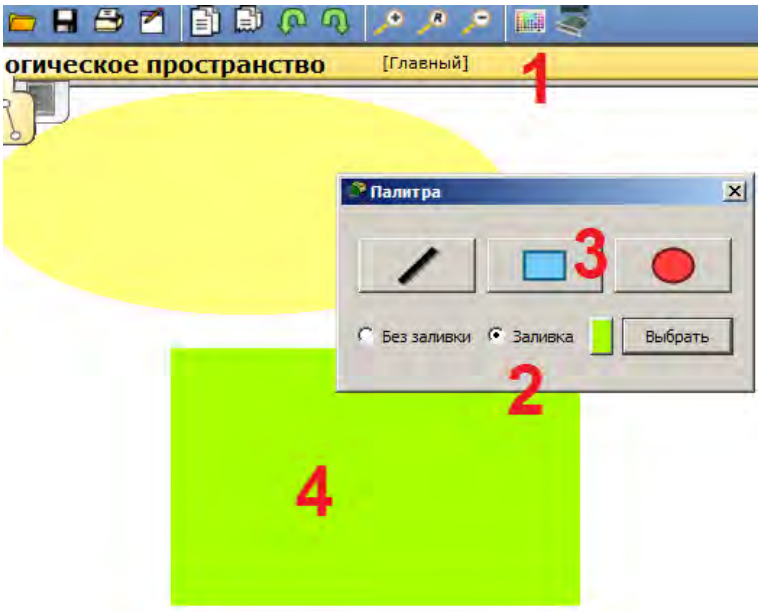


Рисунок 2.14. Палитра

Полезно использовать, когда отделяется визуально одна подсеть от другой, например. Для этого необходимо:

1. На панели инструментов выбирать соответствующий значок;
2. Выбирать режим области «Заливка», например;
3. Выбирать цвет и форму;
4. Нарисовать область на рабочем пространстве.

Можно также добавить подпись и перемещать/масштабировать эту область.

Рассмотрим работу с логической диаграммой. Разместим на схеме два маршрутизатора, как показано ниже и выберем медный кроссовый кабель (рис 2.15).

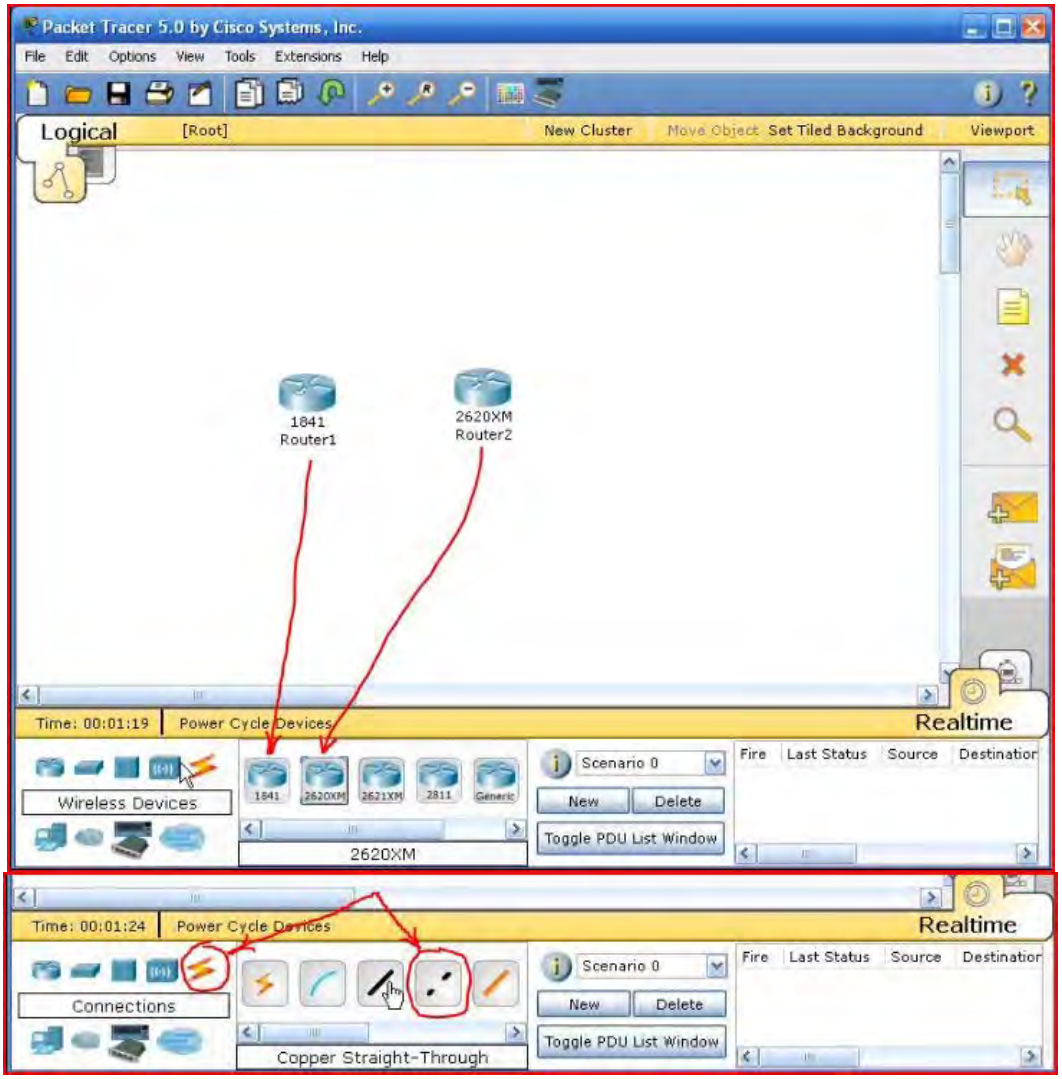


Рисунок 2.15. Моделирование сети

Соединить порты роутеров (рис 2.16).

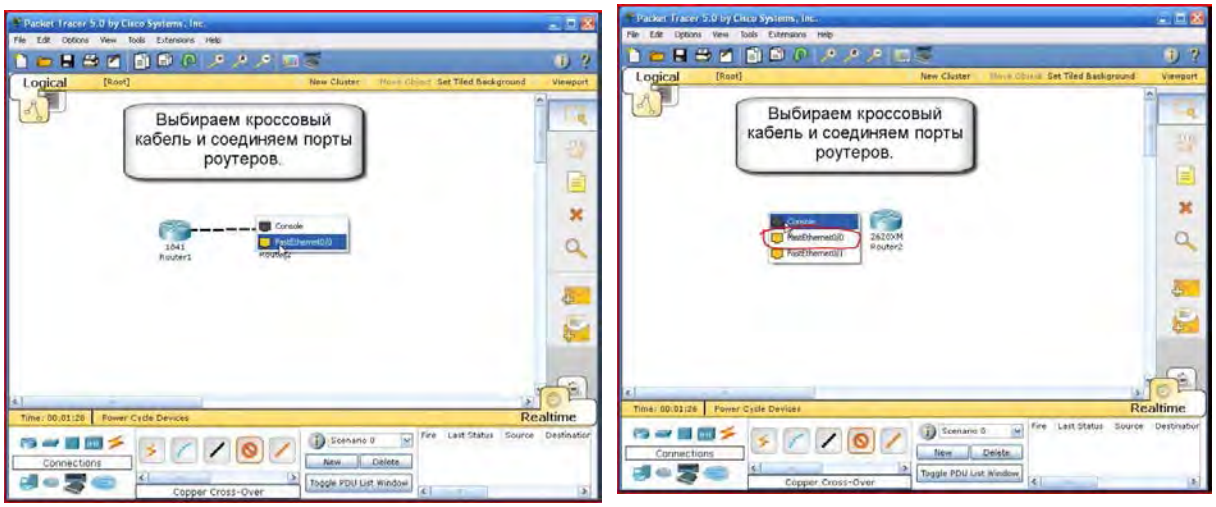


Рисунок 2.16. Соединение роутеров

Не забывайте подписывать оборудование – метка на панели справа (рис 2.17).

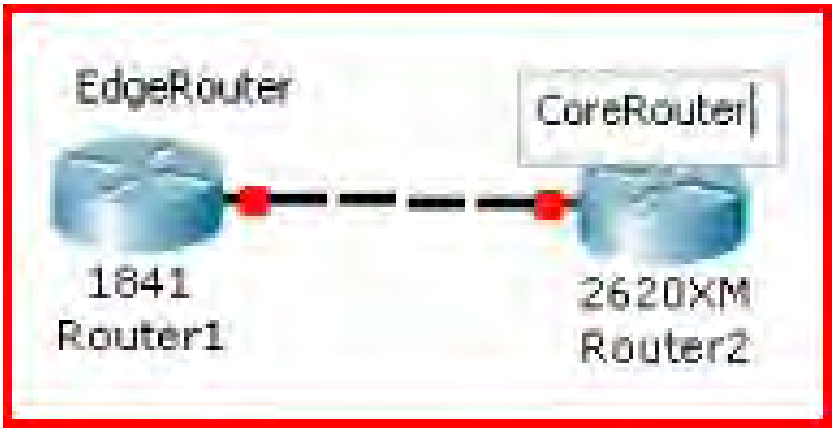


Рисунок 2.17. Подписи

Удалим при помощи кнопки правый роутер и соединение, надпись.

Оставим на схеме роутер 1841. Кликом на роутере открываем его **физическую конфигурацию** (рис. 2.18).

Физическое комплектование Маршрутизатора заключается в дополнении его модульных составляющих и последующей их настройке.

Выбираем плату WIC-2T (Рисунок 2.19). Устанавливаем в пустое пространство (цифра 3), возле выключателя (цифра 1).

Выбираем WIC-1ENET (рисунок 2.20), это однопортовая 10 Мб/с Ethernet карта для 10BASE-T Ethernet LAN. Устанавливаем в другое свободное пространство.

Что нужно знать о модулях WIC (HWIC, VWIC):

1. WIC – WAN interface card. the first original models.
2. HWIC- high-speed wan interface card- the evolution of wic that is now in use on the ISR routers.
3. VIC – voice interface card, support voice only.
4. VIC2 – evolution of the above
5. VWIC – voice and wan interface card. An E1/T1 card that can be user for voice or data.
6. VWIC2 – evolution of the above

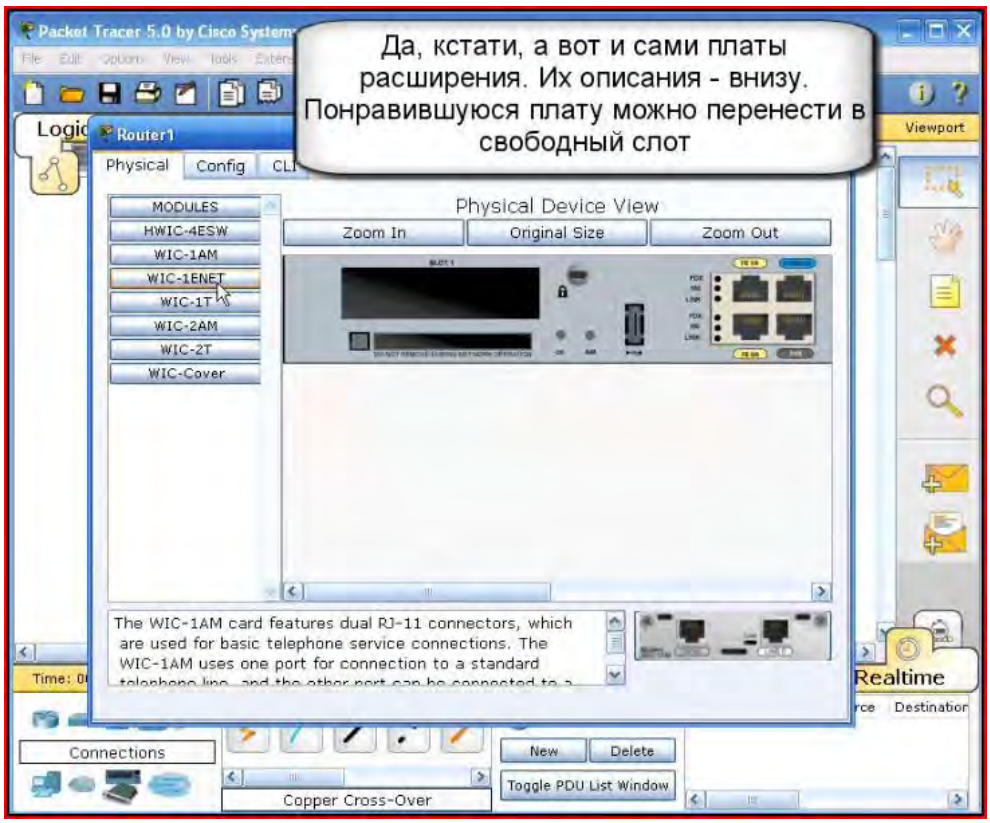


Рисунок 2.18. Физическая конфигурация

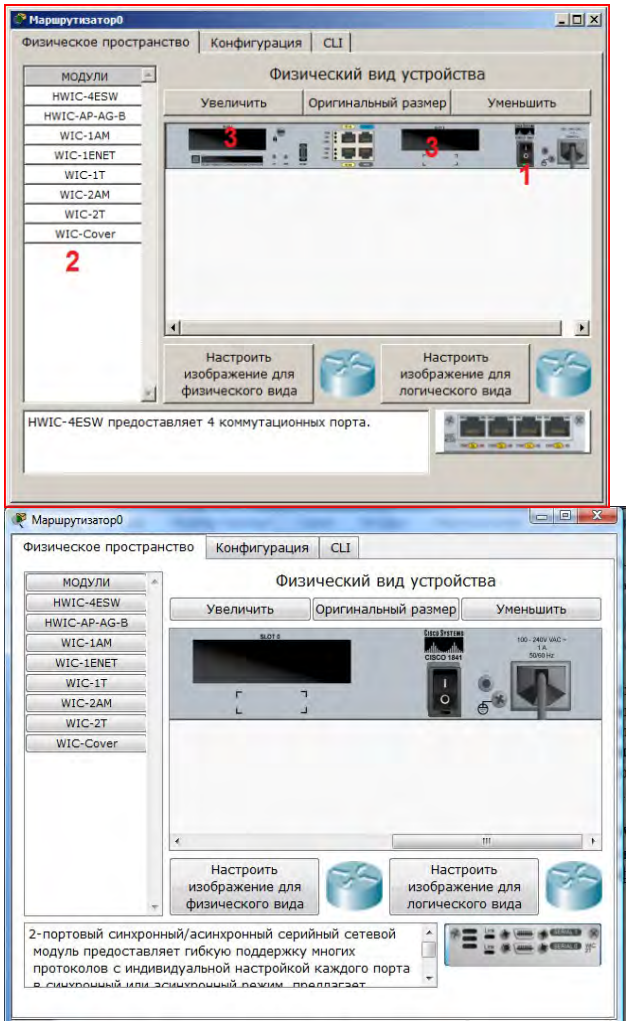


Рисунок 2.19. Изменение физической конфигурации

Иначе говоря, это платы расширения, увеличивающие функционал маршрутизатора. Как, например, для компьютера есть платы, подключаемые к PCI- шине (TV-тюнеры, звуковые карты, USB-разветвители, сетевые карты), так и здесь аналогично подключаются дополнительные платы.

Устройство Cisco можно сравнить с системным блоком со своей операционной системой и многими сетевыми картами, который может обеспечить различный функционал при работе с сетью.

А теперь подробнее о тех модулях, что нам предоставляет Cisco Packet Tracer

• **HWIC – 4ESW** – высокопроизводительный модуль с 4-мя коммутационными портами Ethernet под разъем RJ-45. Позволяет сочетать в маршрутизаторе возможности коммутатора.

• **HWIC-AP-AG-B** – это высокоскоростная WAN-карта, обеспечивающая функционал встроенной точки доступа для роутеров линейки Cisco 1800 (модульных), Cisco 2800 и Cisco 3800. Данный модуль поддерживает радиоканалы Single Band 802.11b/g или Dual Band 802.11a/b/g.

• **WIC-1AM** включает в себя два разъема RJ-11 (телефоннка), используемых для подключения к базовой телефонной службе. Карта использует один порт для соединения с телефонной линией, другой может быть подключен к аналоговому телефону для звонков во время простоя модема.

• **WIC-1ENET** – это однопортовая 10 Мб/с Ethernet карта для 10BASE-T Ethernet LAN.

• **WIC-1T** предоставляет однопортовое последовательное подключение к удаленным офисам или устаревшим серийным сетевым устройствам, например SDLC концентраторам, системам сигнализации и устройствам packet over SONET (POS).

• **WIC-2AM** содержит два разъема RJ-11, используемых для подключения к базовой телефонной службе. В WIC-2AM два модемных порта, что позволяет использовать оба канала для соединения одновременно.

• **WIC-2T** – 2-портовый синхронный/асинхронный серийный сетевой модуль предоставляет гибкую поддержку многих протоколов с индивидуальной настройкой каждого порта в синхронный или асинхронный режим.

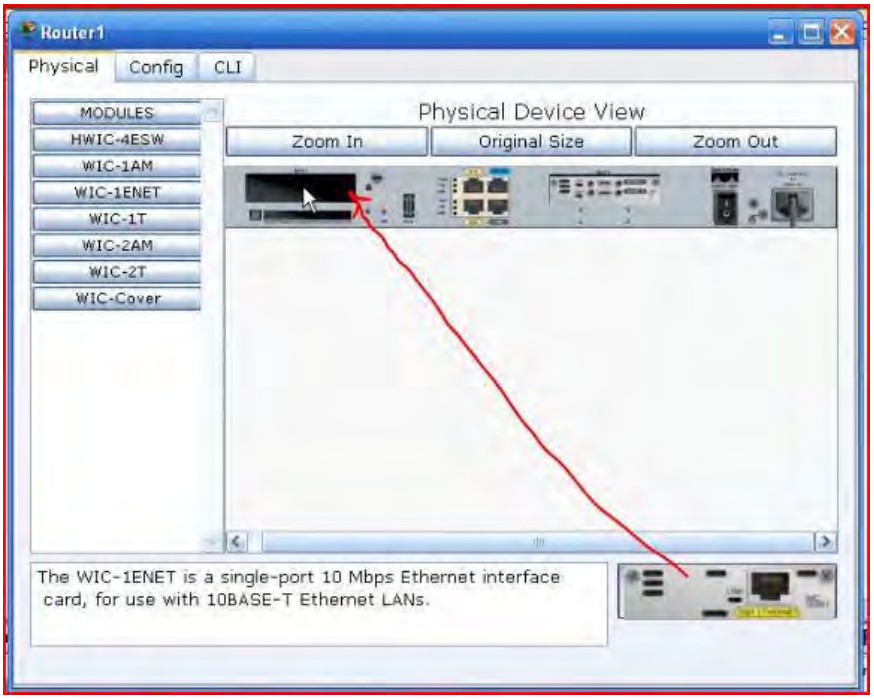


Рисунок 2.20 – Последующее изменение физической конфигурации

Применения для синхронной/асинхронной поддержки представляют:

• низкоскоростную агрегацию (до 128 Кб/с);

• поддержку dial-up модемов;

• синхронные или асинхронные соединения с портами управления другого оборудования и передачу устаревших протоколов типа Bi-sync и SDLC.

• WIC-Cover - стенка для WIC слота, необходима для защиты электронных компонентов и для улучшения циркуляции охлаждающего воздушного потока.

Включите устройство. Рассмотрите работу с командной строкой (CLI) (рис. 2.21). Здесь мы можем прописывать различные команды для маршрутизатора.

Рекомендуется все настройки делать в консоли (CLI). Пока настраивать оборудование не будем. Так же это можно делать во ввкладке Config (рис. 2.22).

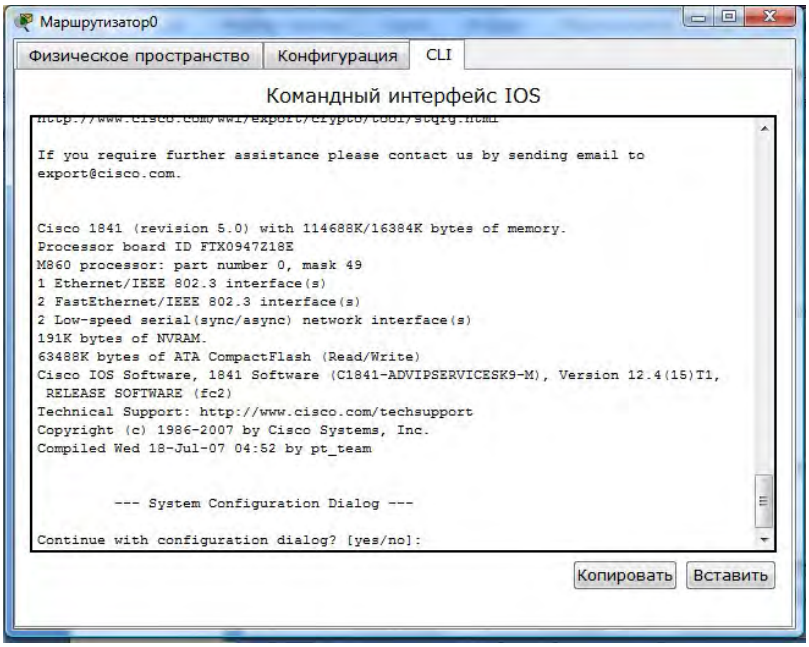


Рисунок 2.21. Командная строка

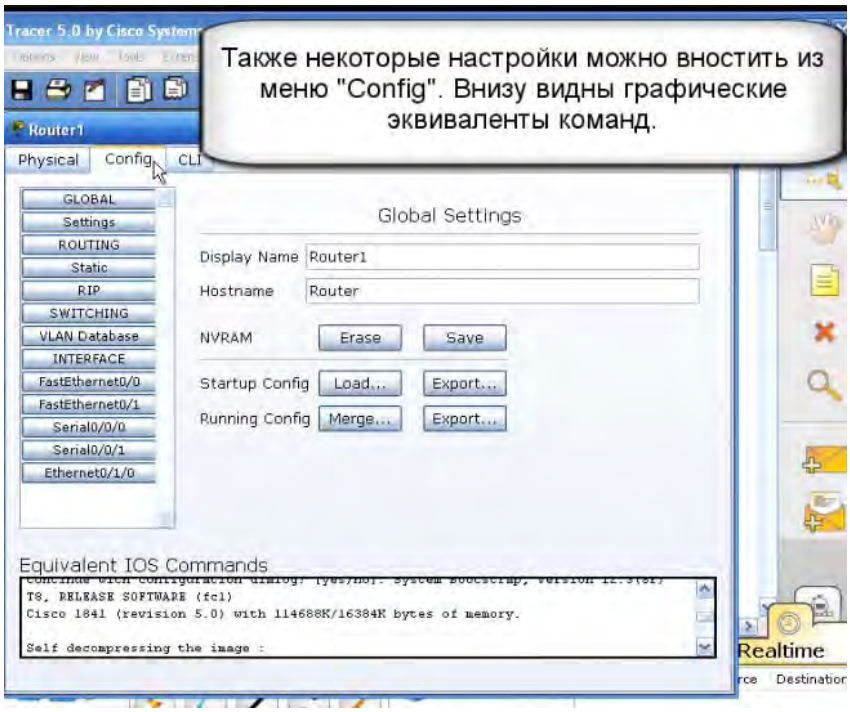


Рисунок 2.22. Настройка во вкладке Config

Удалите роутер. Посмотрим, как устроен компьютер и сервер. Попробуем настроить их. Выносим компьютер и сервер (рис. 2.23).

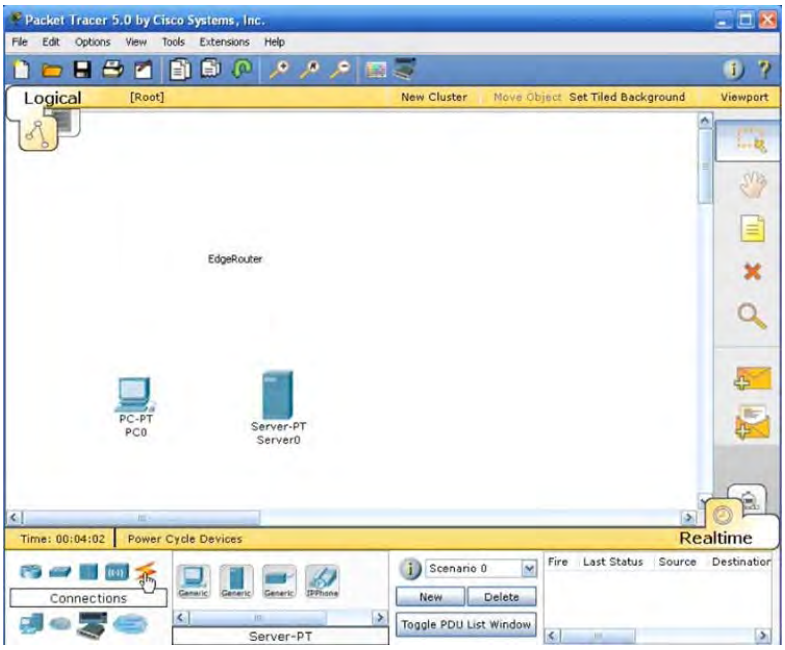


Рисунок 2.23. ПК и сервер

Соединяем обязательно кроссовым кабелем (рис. 2.24).

Рисунок 2.24. Соединяемый порт ПК.

Щелкаем на компьютер, переходим в окно настройки (рис. 2.25).

Настроим IP-адрес. Subnet Mask определяет, какие адреса являются локальными (к ним компьютер будет обращаться напрямую), а какие нет (к ним обращение будет идти через маршрутизатор), Default Gateway — адрес шлюза, он же маршрутизатор (роутер), DNS- сервер — приложение, предназначенное для ответов на DNS-запросы по соответствующему протоколу.

Рисунок 2.25. Окно настройки

Настраиваются: IP-адрес, Маска подсети, Основной шлюз, DNS-сервер (рис. 2.26).

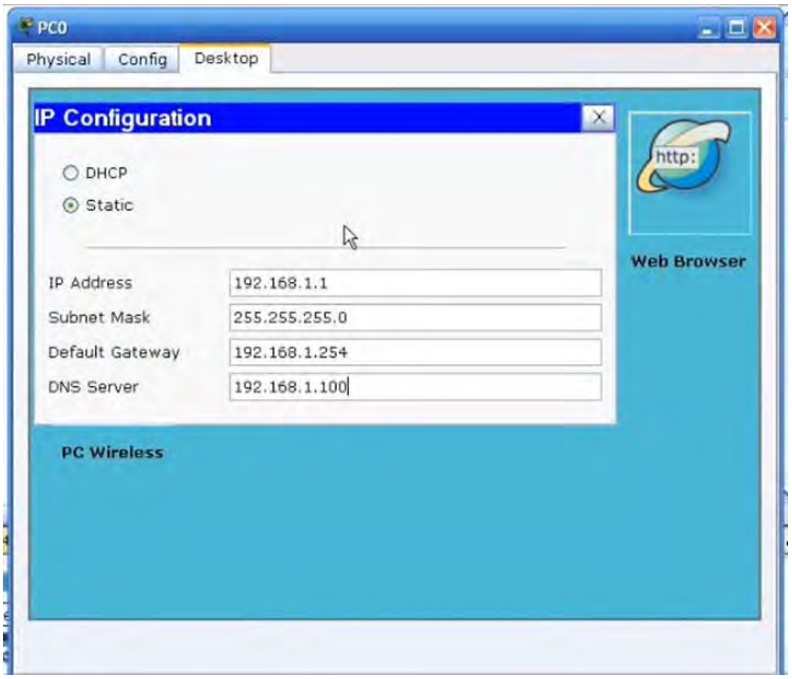


Рисунок 2.26. Настройка ПК

Настроим сервер. Переходим в настройки FastEthernet (рис. 2.27).

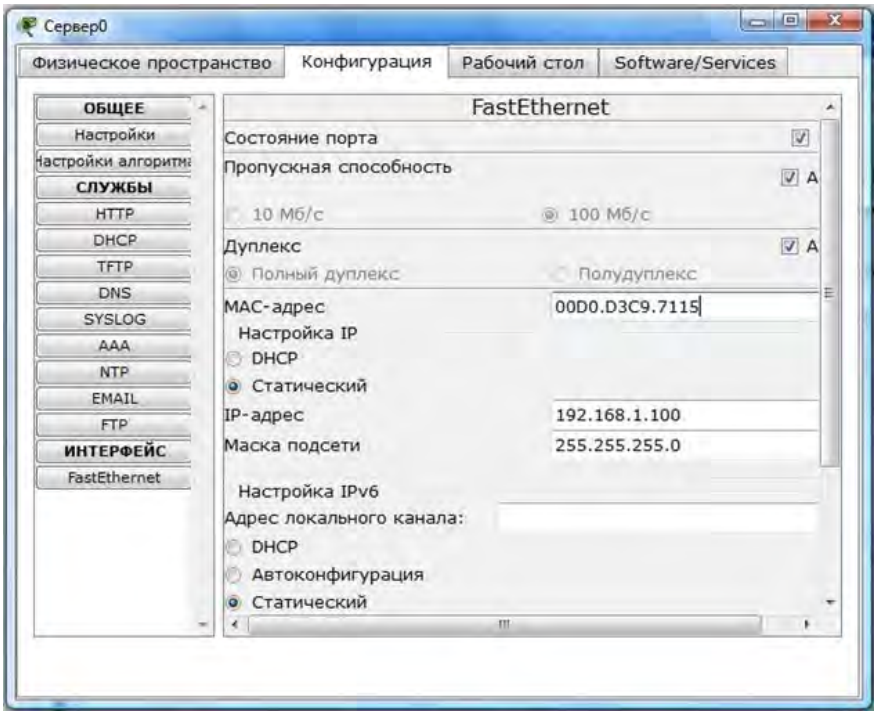


Рисунок 2.27. Настройки FastEthernet сервера

Переходим в настройки DNS. Вводим имя домена, IP адрес (рис.2.28).

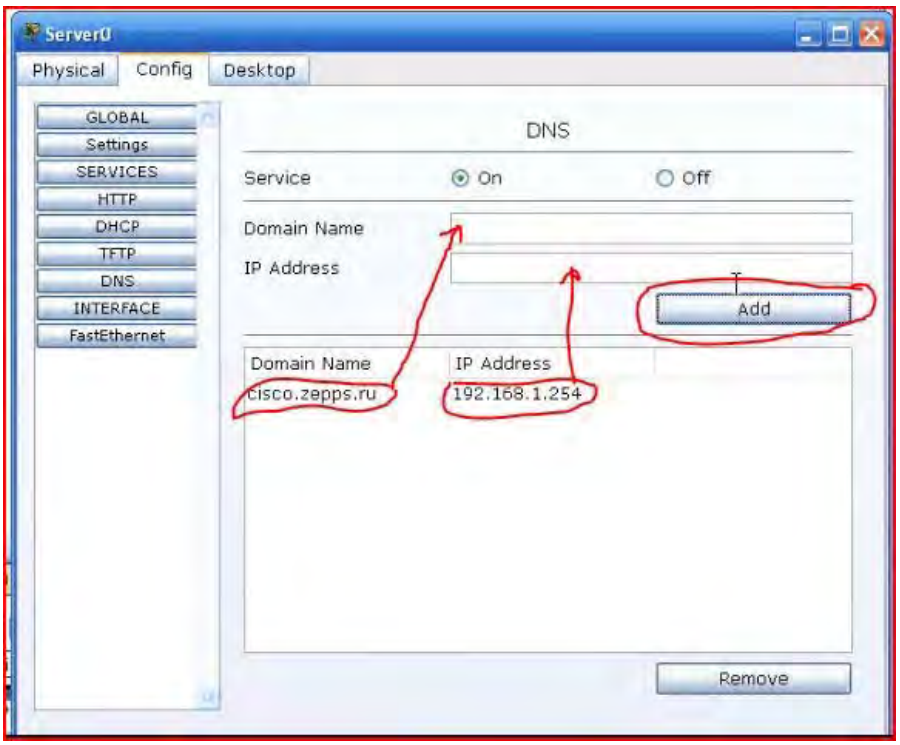


Рисунок 2.28. Настройки DNS

Зайдем в окно настройки компьютера. Проверим настройки терминала (рис. 2.29).

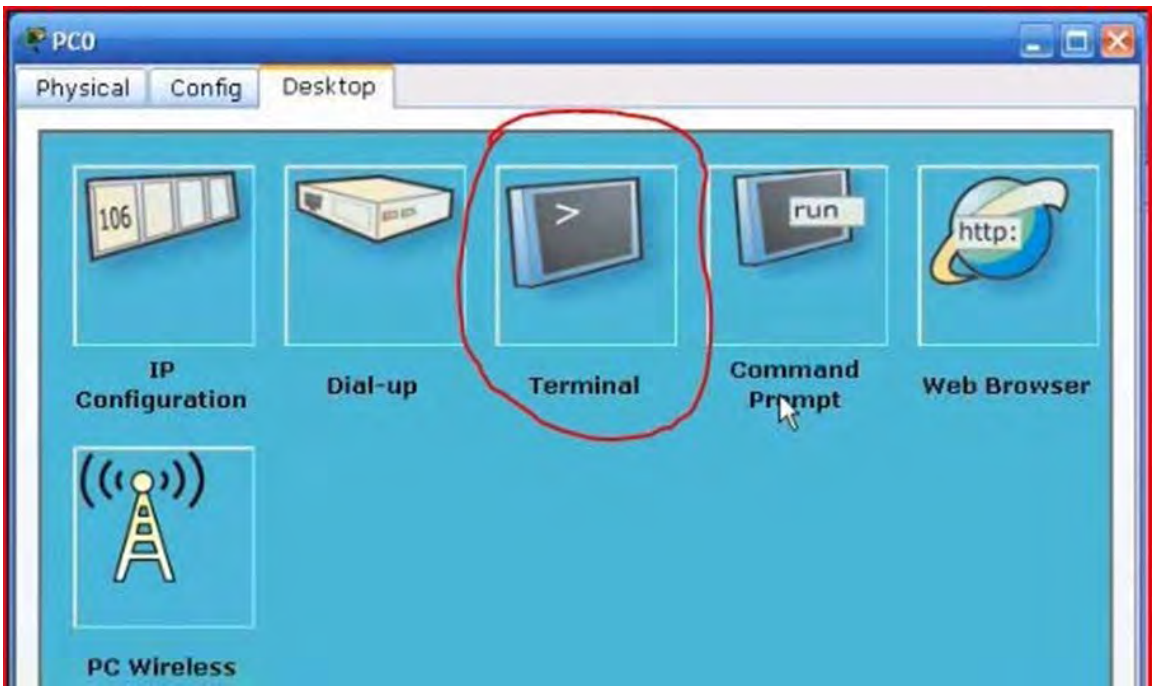


Рисунок 2.29. Кнопка вызова терминала

На компьютере настроен терминал, паритет и управление потоком отключены (рис. 2.30).

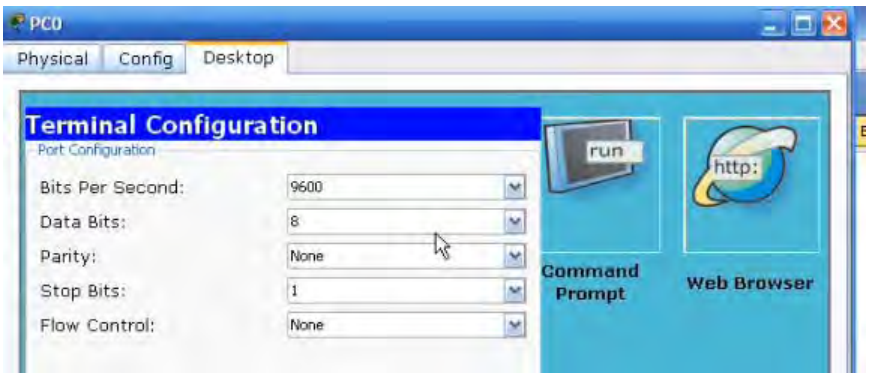


Рисунок 2.30. Настройки терминала

Это терминал для подключения. Зайдем в командную строку ПК (рис. 2.31).

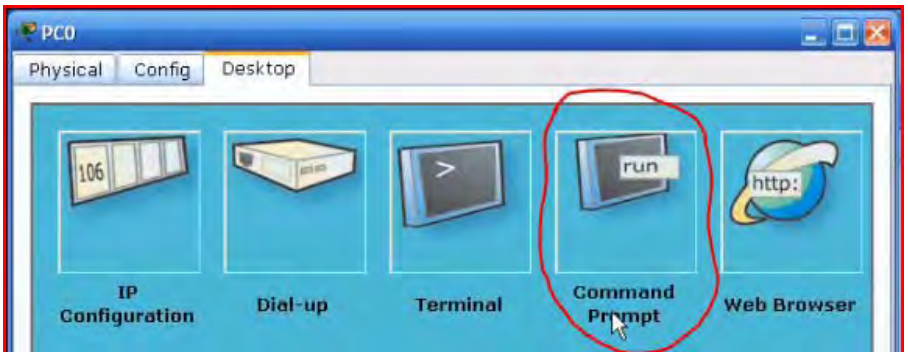


Рисунок 2.31. Кнопка вызова командной строки

Пропингуем сеть командой ping 192.168.1.100 (рис. 2.32).

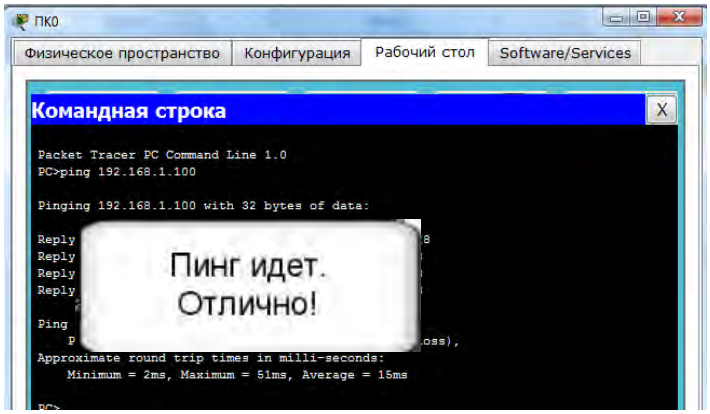


Рисунок 2.32. Результат выполнения команды ping

Удалите соединение между компьютером и сервером. Соединим компьютер и сервер через свич (рис. 2.33).

Соедините компьютер с свитчем, подождите, пока соединение установится. Подключите сервер.

Проведем указатель с конвертом от компьютера к серверу. Щелкнем на сервере мышью. После того как исчезнут желтые точки, т.е. будет установлен канал связи попробуем пропинговать еще раз. Проверим ping, все работает, команда – ping 192.168.1.100 (рис. 2.34).

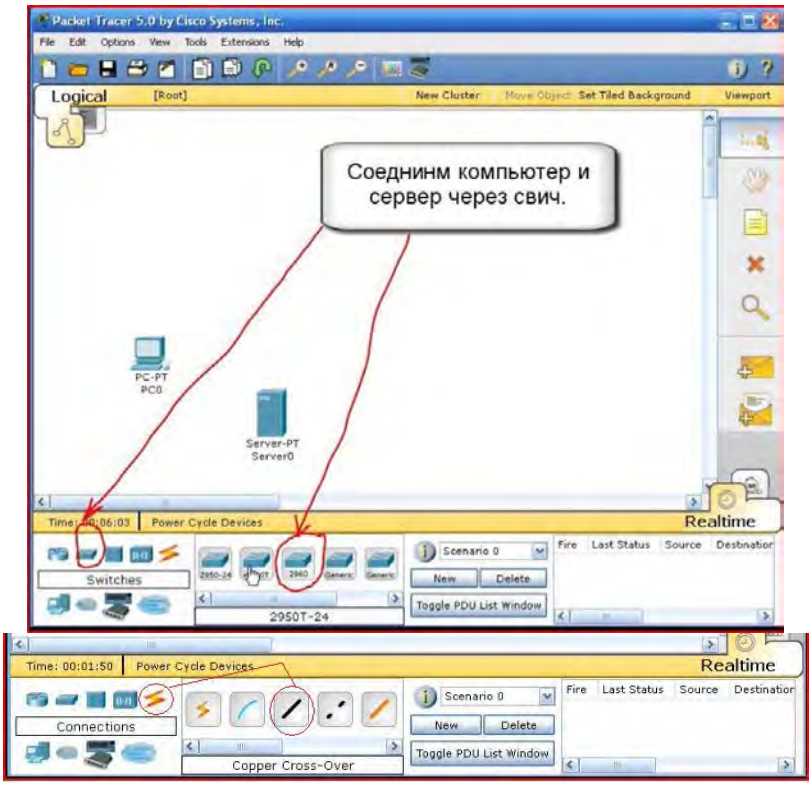


Рисунок 2.33. Соединение компьютера и сервера

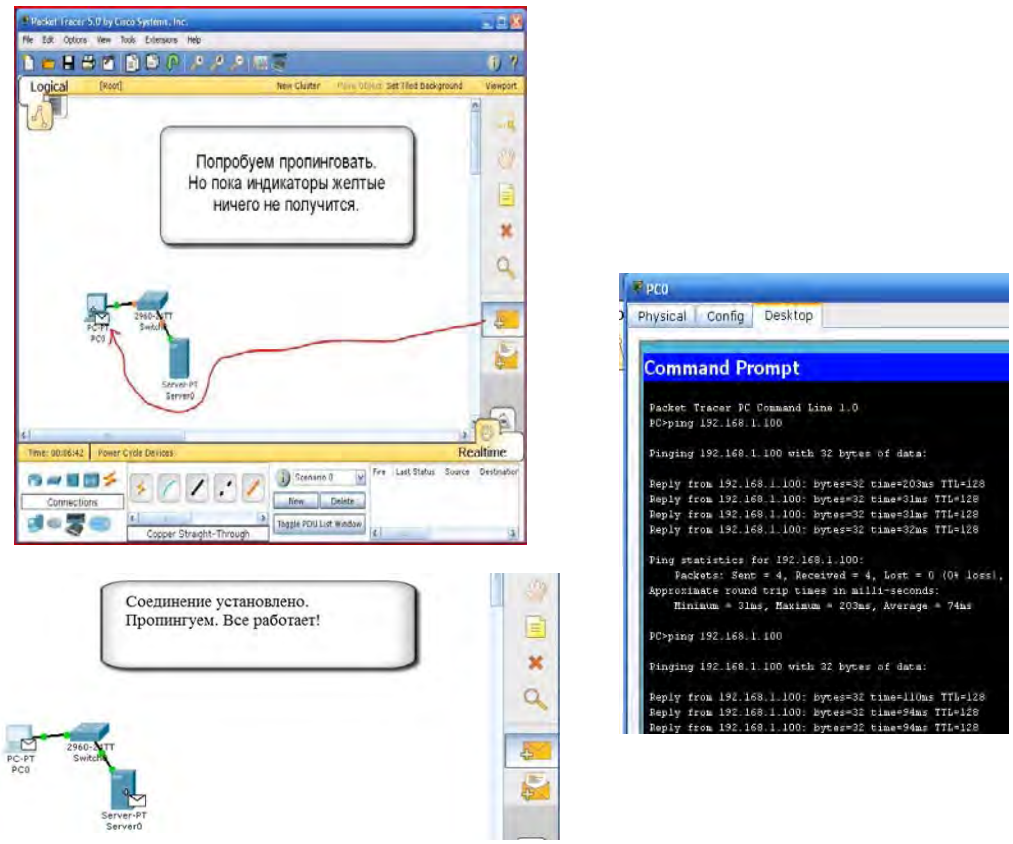


Рисунок 2.34. Успешное соединение при помощи свитча.

Добавим роутер 2621XM. Подпишем его, добавим соединение от свитча к роутеру (рис. 2.35).

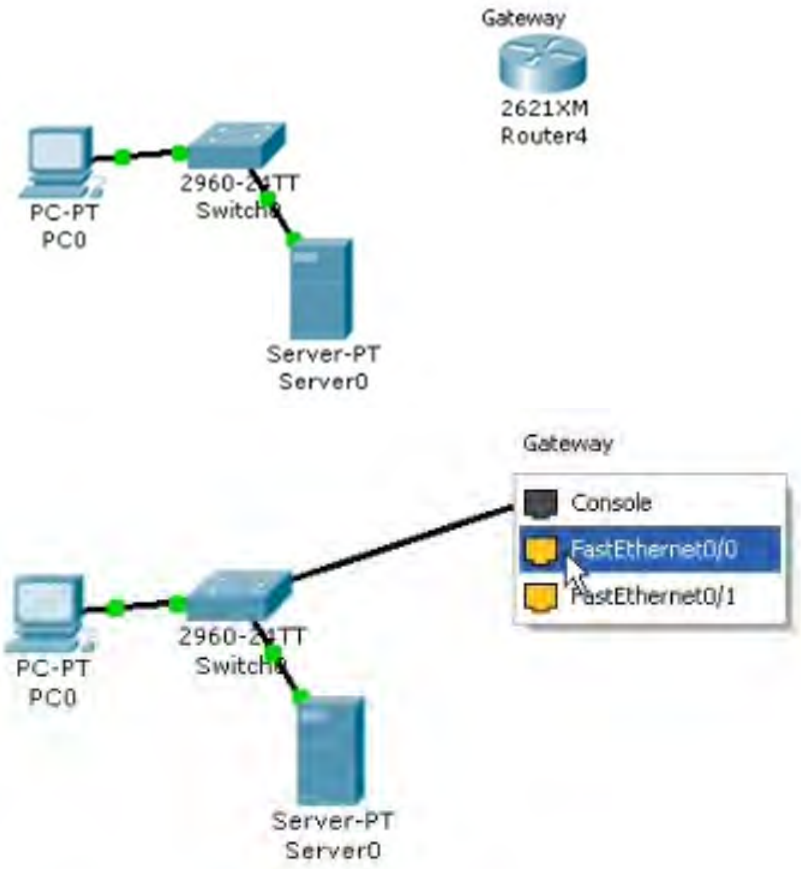


Рисунок 2.35. Подключение роутера

Добавляем устройства (сервер в интернете) (рис. 2.36).

Рисунок 2.36. Добавление устройства

Добавим сервер, соединим устройства кроссовым кабелем (рис. 2.37).

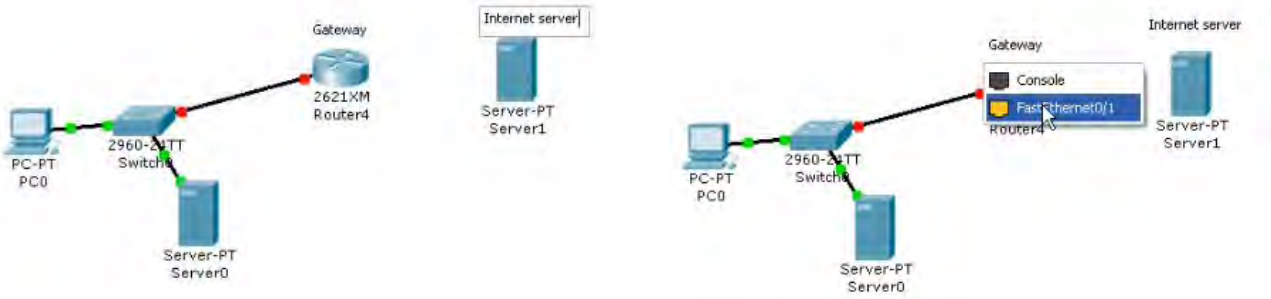


Рисунок 2.37. Добавление Internet-сервера

Настраиваем IP-адрес сервера 1 (рис. 2.39).

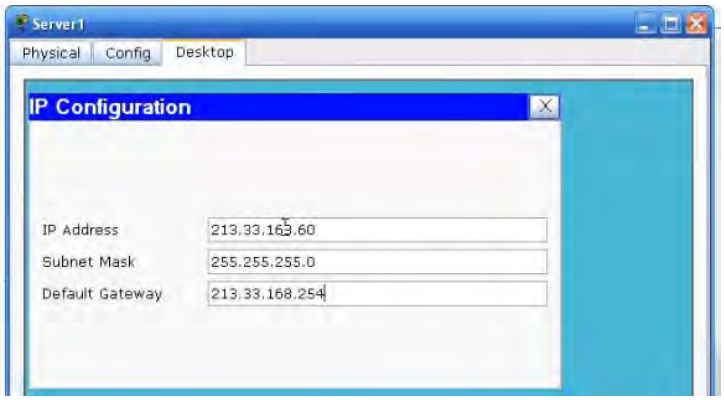


Рисунок 2.39. Настройки сервера 1

IP-адрес 213.33.163.60
Маска подсети 255.255.255.0
Основной шлюз 213.33.168.254
Подправим HTML – страничку (рис. 2.40).

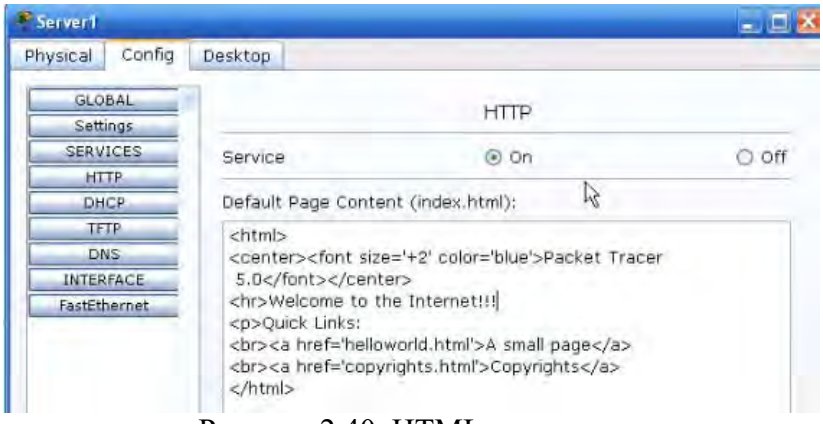


Рисунок 2.40. HTML-страницы

Настроим роутер (рис. 2.41):

Рисунок 2.41 – Настройка роутера

Добавим DNS-запись на сервере (рис. 2.42):

Рисунок 2.42. DNS-запись

Сделаем настройки на компьютере для Web-браузера (рис. 2.43):

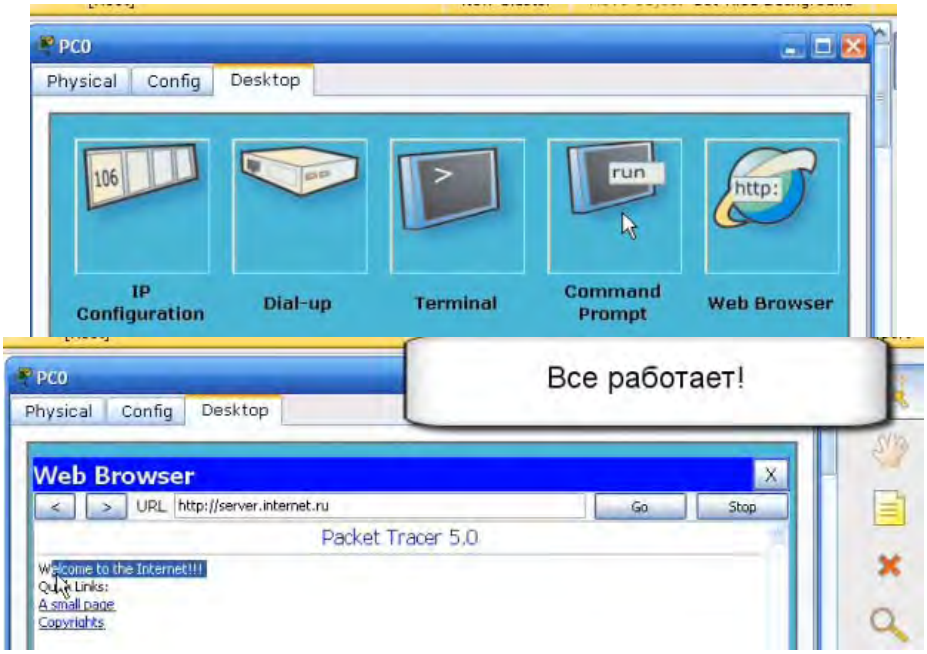


Рисунок 2.43. Настройки для WEB-браузера

Для верности посмотрим, как идут наши пакеты с помощью команды tracert (рис. 2.43):

Рисунок 2.43. Результат выполнения команды tracert

Подключаемся консольным кабелем (рис. 2.44):

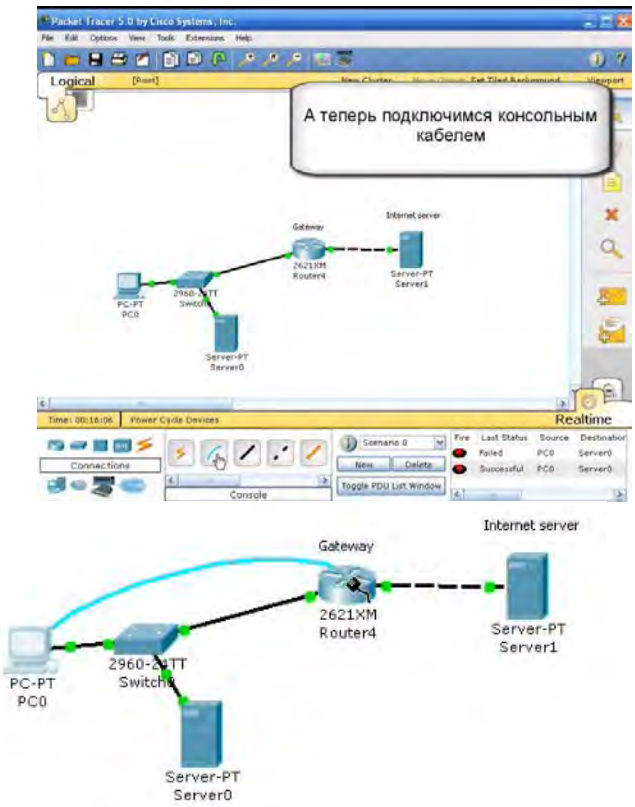


Рисунок 2.44 – Подключение консольным кабелем

Посмотрим, что роутер сообщает о интерфейсах (рисунок 2.45):

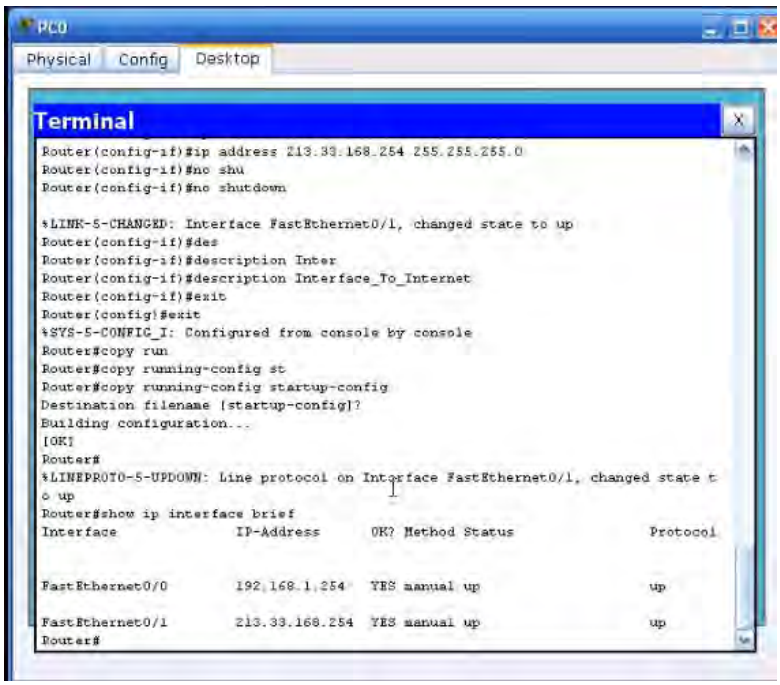


Рисунок 2.45. Системное сообщение роутера

Проверим канал связи (рис. 2.46).

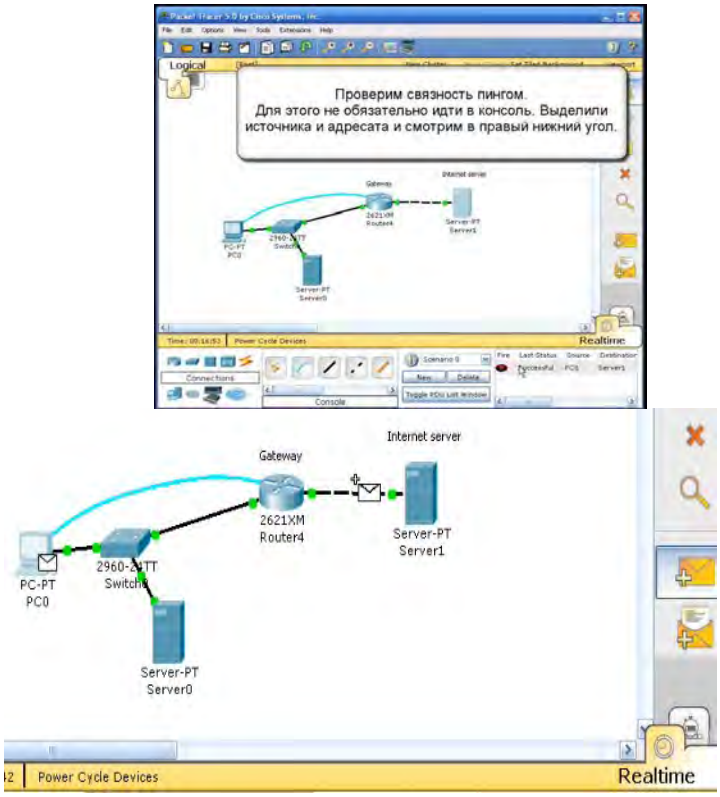


Рисунок 2.46. Проверка канала связи

Сформируем сложный запрос (рис. 2.47).

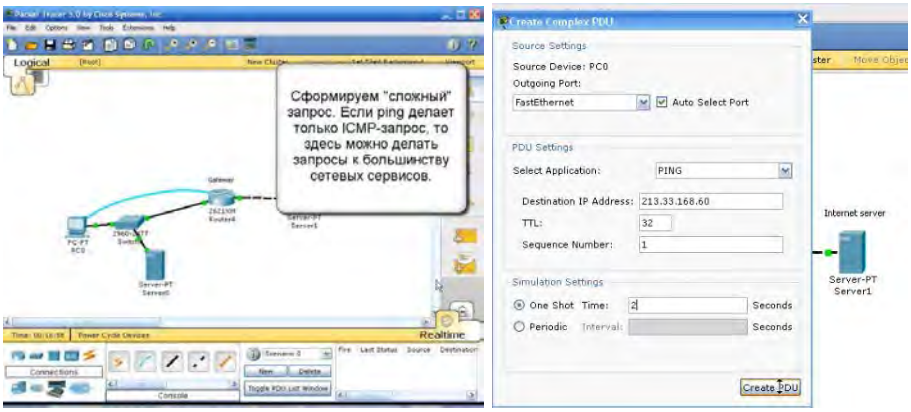


Рисунок 2.47 – Формирование сложного запроса

Отключим один интерфейс (рис. 2.48).

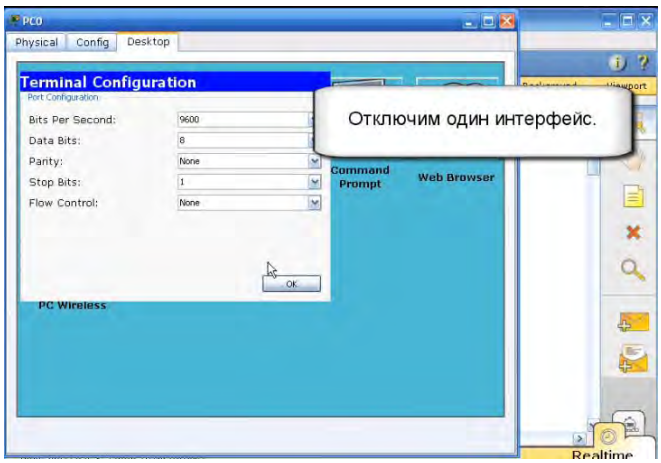


Рисунок 2.48. Отключение интерфейса

Пинг не пройдет (рис. 2.49).

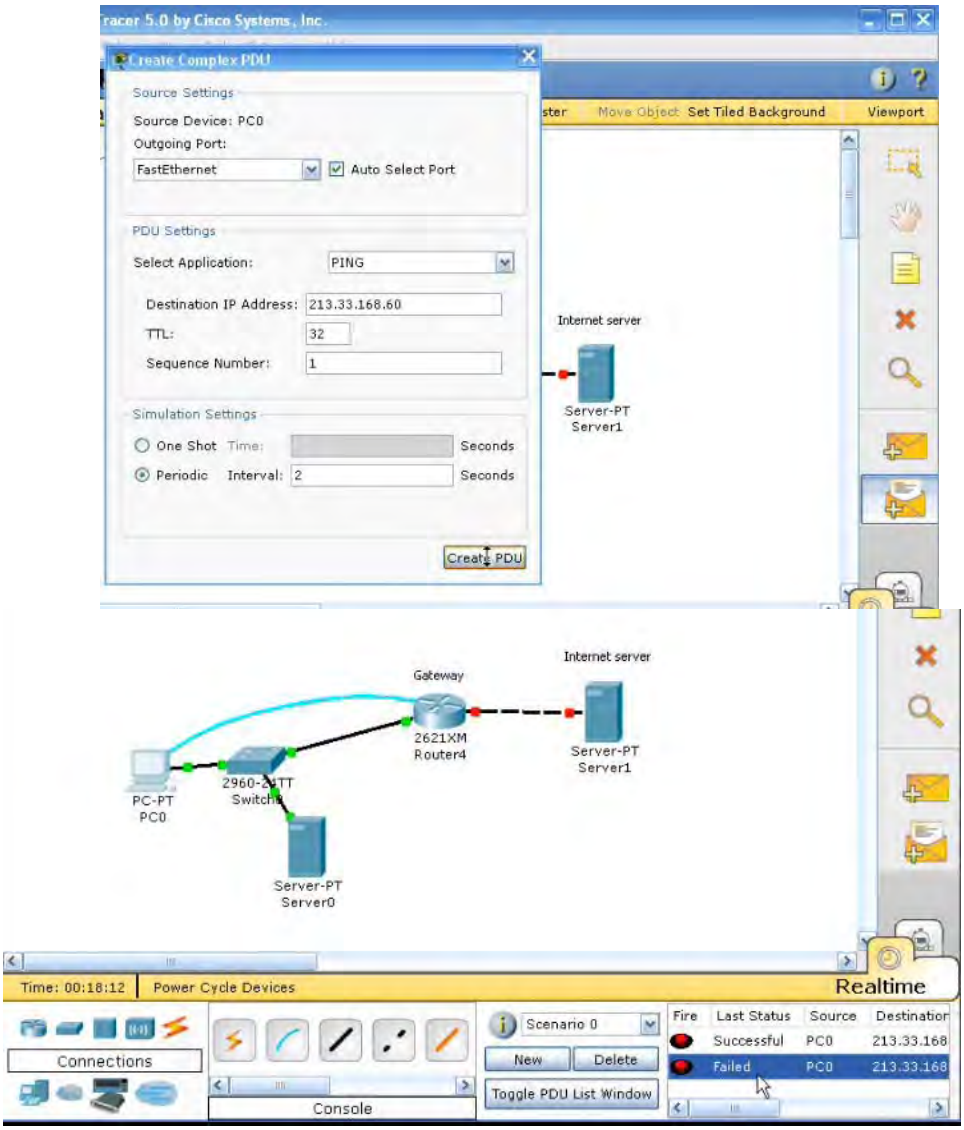


Рисунок 2.49. Формирование запроса

Содержание отчета:

1. Титульный лист.
2. Цель работы.
3. Реализация всех шагов лабораторной работы с предоставлением скриншотов.

Контрольные вопросы:

1. Назначение пакета Cisco Packet Tracer.
2. Возможности пакета Cisco Packet Tracer.
3. Добавление устройств.
4. Соединение устройств.

2.10. Лабораторная работа 10

Соединение двух сетей.

Цель работы: Научиться соединять две сети в эмуляторе Cisco Packet Tracer.

Ход работы:

Симулятор Cisco Packet Tracer позволяет проектировать свои собственные сети, создавая и отправляя различные пакеты данных, сохранять и комментировать свою работу. Студенты могут изучать и использовать такие сетевые устройства, как коммутаторы второго и третьего уровней, рабочие станции, определять типы связей между ними и соединять их. После того, как сеть спроектирована, можно приступать к конфигурированию выбранных устройств посредством терминального доступа или командной строки.

Отличительной особенностью данного симулятора является наличие в нем «Режима симуляции» (рис. 2.50). В данном режиме все пакеты, пересылаемые внутри сети, отображаются графически. Эта возможность позволяет студентам наглядно продемонстрировать, по какому интерфейсу в данные момент перемещается пакет, какой протокол используется и т.д.

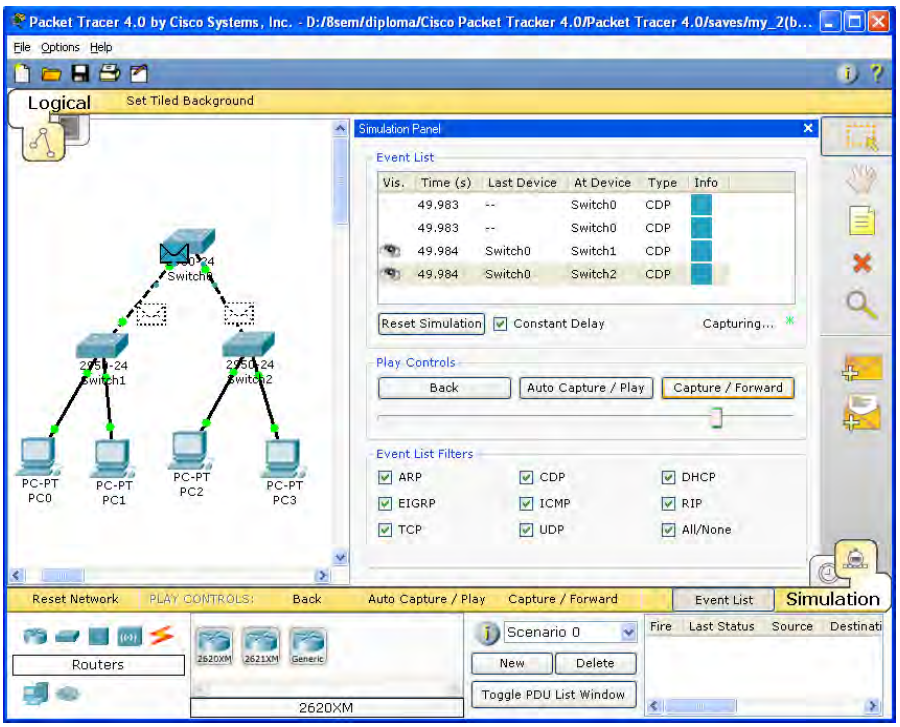


Рисунок 2.50. Режим «Симуляции» в Cisco Packet Tracer

Однако, это не все преимущества Packet Tracer: в «Режиме симуляции» студент может не только отслеживать используемые протоколы, но и видеть, на каком из семи уровней модели OSI данный протокол задействован (рис. 2.51).

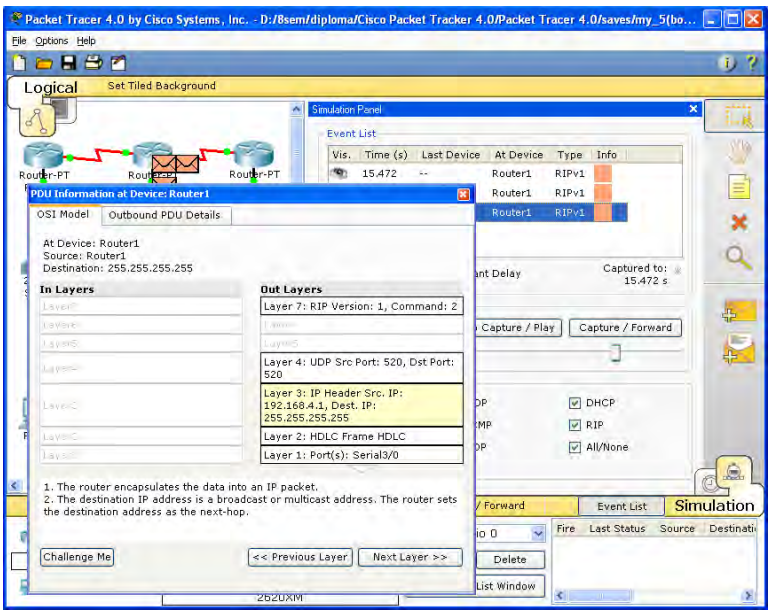


Рисунок 2.51. Анализ семиуровневой модели OSI в Cisco Packet Tracer

Packet Tracer способен моделировать большое количество устройств различного назначения, а так же немало различных типов связей, что позволяет проектировать сети любого размера на высоком уровне сложности:

Моделируемые устройства:
Коммутаторы третьего уровня:
• Router 2620 XM;
• Router 2621 XM;
• Router-PT.
Коммутаторы второго уровня:
• Switch 2950-24;
• Switch 2950T;
• Switch-PT;
• соединение типа «мост» Bridge-PT.
Сетевые концентраторы:
• Hub-PT;
• повторитель Repeater-PT.
Оконечные устройства:
• рабочая станция PC-PT;
• сервер Server-PT;
• принтер Printer-PT.
Беспроводные устройства:
• точка доступа AccessPoint-PT.
• Глобальная сеть WAN.
Типы связей:
• консоль;
• медный кабель без перекрещивания (прямой кабель);
• медный кабель с перекрещиванием (кросс-кабель);
• волоконно-оптический кабель;
• телефонная линия;
• Serial DCE;
• Serial DTE.
Протоколы, доступные для отслеживания:
• ARP;
• CDP;
• DHCP;
• EIGRP;
• ICMP;
• RIP;
• TCP;
• UDP.
Описание терминального режима

Маршрутизатор конфигурируется в командной строке операционной системы Cisco IOS. Подсоединение к маршрутизатору осуществляется через Telnet на IP-адрес любого из его интерфейсов или с помощью любой терминальной программы через последовательный порт компьютера, связанный с консольным портом маршрутизатора. Последний способ предпочтительнее, потому что процесс конфигурирования маршрутизатора может изменять параметры IP-интерфейсов, что приведет к потере

соединения, установленного через Telnet. Кроме того, по соображениям безопасности доступ к маршрутизатору через Telnet следует запретить.

В рамках данного курса конфигурация маршрутизаторов будет осуществляться посредством терминала.

При работе в командной строке Cisco IOS существует несколько контекстов (режимов ввода команд).

Контекст пользователя открывается при подсоединении к маршрутизатору; обычно при подключении через сеть требуется пароль, а при подключении через консольный порт пароль не нужен. В этот же контекст командная строка автоматически переходит при продолжительном отсутствии ввода в контексте администратора. В контексте пользователя доступны только простые команды (некоторые базовые операции для мониторинга), не влияющие на конфигурацию маршрутизатора. Вид приглашения командной строки:

router>

Вместо слова router выводится имя маршрутизатора, если оно установлено.

Контекст администратора (контекст "exec") открывается командой enable, поданной в контексте пользователя; при этом обычно требуется пароль администратора. В контексте администратора доступны команды, позволяющие получить полную информацию о конфигурации маршрутизатора и его состоянии, команды перехода в режим конфигурирования, команды сохранения и загрузки конфигурации. Вид приглашения командной строки:

router#

Обратный переход в контекст пользователя производится по команде **disable** или по истечении установленного времени неактивности. Завершение сеанса работы - команда **exit.**

Глобальный контекст конфигурирования открывается командой config terminal ("конфигурировать через терминал"), поданной в контексте администратора. Глобальный контекст конфигурирования содержит как непосредственно команды конфигурирования маршрутизатора, так и команды перехода в контексты конфигурирования подсистем маршрутизатора, например:

контекст конфигурирования интерфейса открывается командой **interface имя_интерфейса** (например, **interface serial0**), поданной в глобальном контексте конфигурирования;

контекст конфигурирования процесса динамической маршрутизации открывается командой **router протокол номер_процесса** (например, **router ospf 1**, поданной в глобальном контексте конфигурирования.

Существует множество других контекстов конфигурирования. Некоторые контексты конфигурирования находятся внтури других контекстов конфигурирования.

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

router(config)# /глобальный/

router(config-if)# /интерфейса/

rounter(config-router)# /динамической маршрутизации/

rounter(config-line)# /терминальной линии/

Выход из глобального контекста конфигурирования в контекст администратора, а также выход из любого подконтекста конфигурирования в контекст верхнего уровня производится командой **exit** или **Ctrl-Z.** Кроме того, команда **end**, поданная в любом из контекстов конфигурирования немедленно завершает процесс конфигурирования и возвращает оператора в контекст администратора.

Любая команда конфигурации вступает в действие немедленно после ввода, а не после возврата в контекст администратора.

Упрощенная схема контекстов представлена на рис. 2.52.

Все команды и параметры могут быть сокращены (например, "**enable**" - "**en**", "**configure terminal**" - "**conf t**"); если сокращение окажется неоднозначным, маршрутизатор сообщит об этом, а по нажатию табуляции выдаст варианты, соответствующие введенному фрагменту.

В любом месте командной строки для получения помощи может быть использован вопросительный знак:

router#? /список всех команд данного контекста с комментариями/

router#co? /список всех слов в этом контексте ввода, начинающихся на "co" - нет пробела перед "?"/

router#conf ? /список всех параметров, которые могут следовать за командой config - перед "?" есть пробел/

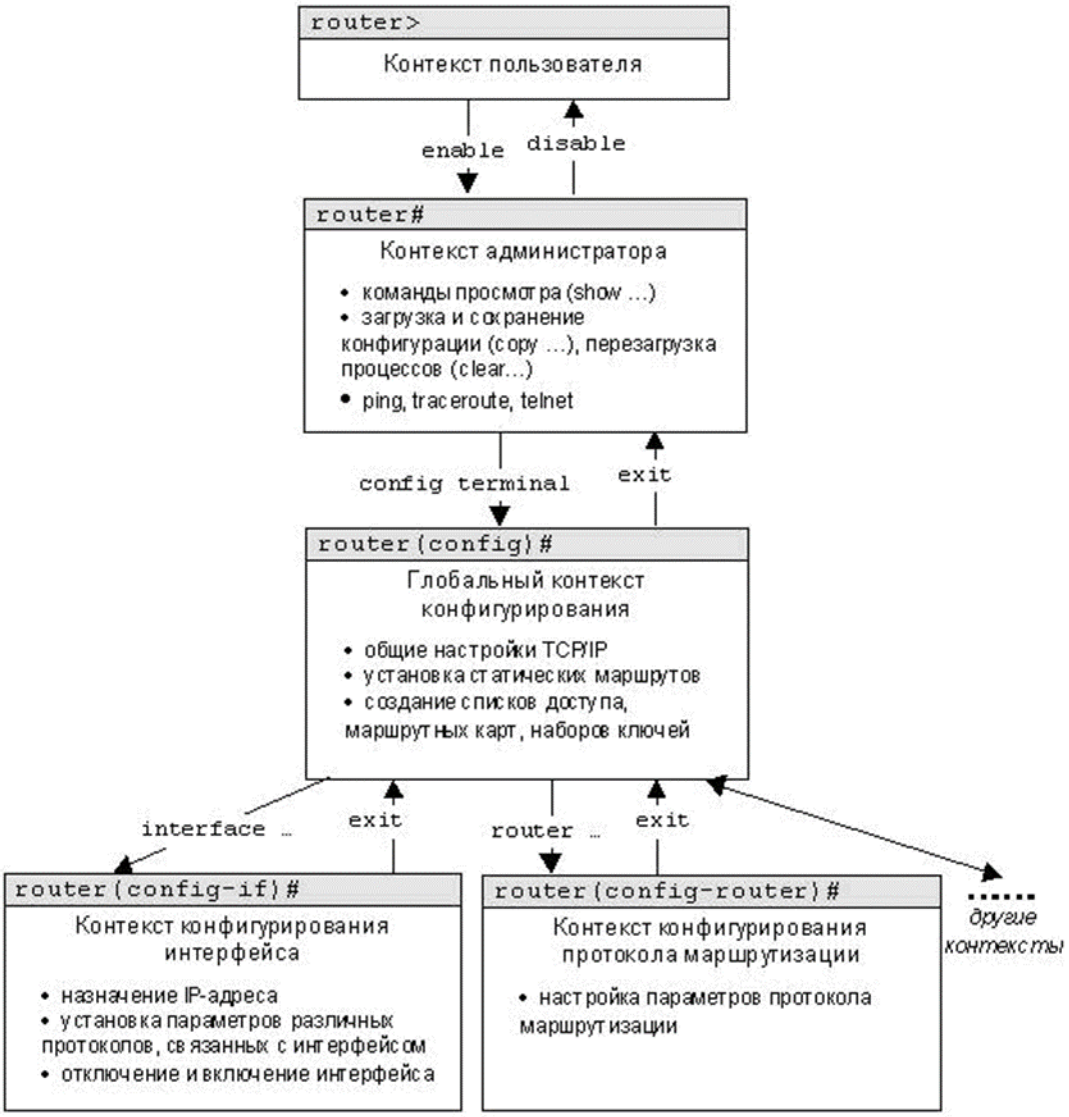


Рисунок 2.52. Схема контекстов Cisco IOS

Список команд

Данный список команд сгруппирован в соответствии с контекстами, в котором они [команды] применяются. В данном списке собраны те команды конфигурирования, которые необходимы для выполнения всех лабораторных работ.

Глобальный контекст конфигурирования

Команда «Access-list»

Критерии фильтрации задаются в списке операторов разрешения и запрета, называемом списком доступа. Строки списка доступа сравниваются с IP-адресами и другой информацией пакета данных последовательно в том порядке, в котором были заданы, пока не будет найдено совпадение. При совпадении осуществляется выход из списка. При этом работа списка доступа напрямую зависит от порядка следования строк.

Списки доступа имеют 2 правила: permit – разрешить, и deny – запретить. Именно они определяют, пропустить пакет дальше или запретить ему доступ.

Списки доступа бывают 2-ух типов: standard – стандартные (номера с 1 до 99) и extended – расширенные (номера с 100 до 199). Различия заключаются в возможности фильтровать пакеты не только по ip-адресу, но и по другим параметрам.

Формат команды (стандартные списки доступа):

access-list номер_списка/имя правило A.B.C.D a.b.c.d , где A.B.C.D a.b.c.d – ip-адрес и подстановочная маска соответственно.

Пример выполнения команды:

Данная команда означает, что данный список доступа блокирует любые пакеты с

ip-адресами 192.168.3.1 - 192.168.3.3.

Команда «Enable secret»

Обычно при входе в привилегированный режим требуется ввести пароль. Данная функция позволяет предотвратить несанкционированный доступ в данный режим, ведь именно из него можно изменять конфигурацию устройства. Данная команда позволяет установить такой пароль.

Формат команды:

enable secret пароль

```
Router(config)#access-list 10 deny 192.168.3.0 0.0.0.3
Router(config)#
```

Пример выполнения команды:

```
Switch(config)#enable secret
123 Switch(config)#
%SYS-5-CONFIG_I: Configured from console by
console Switch#exit

Switch con0 is now
available Press RETURN to
get
```

После того, как был установлен пароль, при попытке входа в привилегированный режим, коммутатор будет требовать от пользователя его ввести – в противном случае вход будет невозможен.

Команда «Interface»

Команда для входа в режим конфигурирования интерфейсов конфигурируемого устройства. Данный режим представляет собой одно из подмножеств режима глобального конфигурирования и позволяет настраивать один из доступных сетевых интерфейсов (fa 0/0, s 2/0 и т.д.). Все изменения,

вносимые в конфигурацию коммутатора в данном режиме, относятся только к выбранному интерфейсу.

Формат команды (возможны 3 варианта):

interface *тип порт*

interface *тип слот/порт*

interface *тип слот/подслот/порт*

Примеры выполнения команды:

Switch(config)#interface *vlan*

1 Switch(config-if)#

Router(config)#int

После введения данной команды с указанным интерфейсом пользователь имеет возможность приступить к его конфигурированию. Необходимо заметить, что, находясь в режиме конфигурирования интерфейса, вид приглашения командной строки не отображает имя данного интерфейса.

Команда «Ip route»

Статическая маршрутизация предполагает фиксированную структуру сети: каждый маршрутизатор в сети точно знает, куда нужно отправлять пакет, чтобы он был доставлен по назначению. Для этого можно прописать статические маршруты, используя данную команду. Команда может быть записана в двух форматах:

Первый формат команды:

ip route A.B.C.D a.b.c.d A1.B1.C1.D1 ,

где A.B.C.D и a.b.c.d – сетевой адрес и маска подсети, куда необходимо доставить пакеты, A1.B1.C1.D1 – ip-адрес следующего маршрутизатора в пути или адрес сети другого маршрутизатора из таблицы маршрутизации, куда должны переадресовываться пакеты;

Второй формат команды:

ip route A.B.C.D a.b.c.d *выходной_интерфейс_текущего_маршрутизатора*

Примеры выполнения команды:

Router(config)#ip route *76.115.253.0 255.0.0.0*

76.115.252.0 Router(config)#

Router(config)#ip route *0.0.0.0 0.0.0.0*

Serial2/0 Router(config)#

Данной командой указывается маршрут, по которому пакеты из одной подсети будут доставляться в другую. Маршрут по умолчанию (Router(config)#ip route 0.0.0.0 1.1.1.1 serial 2/0) указывает, что пакеты, предназначенные узлам в другой подсети должны отправляться через данный шлюз.

Команда «Router rip»

RIP – Routing Information Protocol – протокол динамической маршрутизации. При его использовании отпадает необходимость вручную прописывать все маршруты – необходимо лишь указать адреса сетей, с которыми нужно обмениваться данными. Данная команда позволяет включить rip-протокол.

Пример выполнения команды:

Router(config)#router rip

Данная команда включает rip-протокол на данном маршрутизаторе. Дальнейшая настройка производится из соответствующего контекста маршрутизации, описанного отдельно.

Контекст конфигурирования интерфейса

Команда «Ip access-group»

Данная команда используется для наложения списков доступа. Список накладывается на конкретный интерфейс, и указывается один из 2-ух параметров: in (на входящие пакеты) или out (на исходящие). Необходимо знать, что на каждом интерфейсе может быть включен только один список доступа.

Формат команды:

ip access-group *номер_списка/имя_параметр*

Пример выполнения команды:

Router(config-if)# ip access

group *10 in Router(config-if)#*

В данном примере да выбранный интерфейс накладывается список доступа под номером 10: он будет проверять все входящие в интерфейс пакеты, так как выбран параметр in.

Команда «Bandwidth»

Данная команда используется только в последовательных интерфейсах и служит для установки ширины полосы пропускания. Значение устанавливается в килобитах.

Формат команды:

bandwidth *ширина_полосы_пропускания*

Пример выполнения команды:

Router(config)#interface

serial 2/0 Router(config-if)#bandwidth

После выполнения данной команды ширина полосы пропускания для serial 2/0 будет равна 560 kbits.

Команда «Clock rate»

Для корректной работы участка сети, где используется последовательный сетевой интерфейс, один из коммутаторов 3-его уровня должен предоставлять тактовую частоту. Это может быть оконечное кабельное устройство DCE (расшифровать). Так как маршрутизаторы CISCO являются по умолчанию

устройствами DTE, то необходимо явно указать интерфейсу на предоставление тактовой частоты, если этот интерфейс работает в режиме DCE. Для этого используют данную команду (значение устанавливается в битах в секунду).

Формат команды:

clock rate *тактовая_частота*

Пример выполнения команды:

```
Router(config)#interface
serial 2/0
Router(config-if)#clock
```

После выполнения данной команды тактовая частота для serial 2/0 будет равна

56000 bits per second.

Команда «Ip address»

Каждый интерфейс должен обладать своим уникальным ip-адресом – иначе взаимодействие устройств по данному интерфейсу не сможет быть осуществлено. Данная команда используется для задания ip-адреса выбранному интерфейсу.

Формат команды:

ip address A.B.C.D a.b.c.d ,

где A.B.C.D a.b.c.d – ip-адрес и маска подсети соответственно.

Пример выполнения команды:

```
Switch(config)#interface vlan 1
Switch(config-if)#ip address 172.16.10.5
255.255.0.0  Switch(config-if)#
```

Результат можно проверить командой

Switch#show ip interface vlan 1

Данной командой интерфейсу vlan 1 назначен ip-адрес 172.16.10.5 с маской подсети 255.255.0.0.

Команда «No»

Данная команда применяется в случае необходимости отменить действие какой- либо команды конфигурирования.

Формат команды:

no *команда_которую_следует_отменить*

```
Switch(config-if)# no shutdown
%LINK-5-CHANGED: Interface Vlan1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed
state to up Switch(config-if)#
```

В данном примере использовалась команда shutdown, которая отключает выбранный интерфейс. В итоге после выполнения no shutdown интерфейс включается.

Контекст администратора.

Команда «Configure terminal»

Для конфигурирования устройства, работающего под управлением IOS, следует использовать привилегированную команду configure. Эта команда переводит контекст пользователя в так называемый «режим глобальной конфигурации» и имеет три варианта:

• конфигурирование с терминала;

• конфигурирование из памяти;

• конфигурирование через сеть.

В рамках данного лабораторного курса конфигурирование будет производиться **только** посредством терминала.

Из режима глобальной конфигурации можно делать изменения, который касаются устройства в целом. Также данный режим позволяет входить в режим конфигурирования определенного интерфейса.

Пример выполнения команды:

Router#configure terminal

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

Switch#show startup-config

Using 1540 bytes

!

version 12.1

!

Переход в режим глобальной конфигурации, о чем свидетельствует изменившийся вид приглашения командной строки.

Команда «Copy»

После настройки коммутатора рекомендуется сохранять его текущую конфигурацию. Информация помещается в энергонезависимую память и хранится там столько, сколько нужно. При необходимости все настройки могу быть восстановлены или сброшены.

Формат команды:

copy *running-config startup-config* – команда для сохранения конфигурации

copy *startup-config running-config* – команда для загрузки конфигурации

Пример выполнения команды:

```
Switch#copy running-config startup-
config Building configuration...
[OK]
Switch#
```

В данном примере текущая конфигурация коммутатора была сохранена в энергонезависимую память.

Команда «Show»

Show (англ. - показывать) – одна из наиболее важных команд, использующихся при настройке коммутаторов. Она применяется для

просмотра информации любого рода и применяется практически во всех контекстах. Эта команда имеет больше всех параметров.Здесь будут рассмотрены только те параметры, которые требуются в рамках данного курса. Другие параметры студент может изучить самостоятельно. **Параметр «running-config» команды «Show».** Для просмотра текущей работающей конфигурации коммутатора используется данная команда. Пример выполнения команды:

```
Switch#show running-config
!
version 12.1
!
hostname Switch

...
```

На экран выводится текущие настройки коммутатора.

Параметр «startup-config» команды «Show»

Для просмотра сохраненной конфигурации используется данная команда. Пример выполнения команды:

```
Switch #show

startup-config

startup-config is

not present
```

Если энергонезависим память не содержит информации, тогда коммутатор выдаст сообщение о том, что конфигурация не была сохранена.

Параметр «ip route» команды «Show»

Данная команда применяется для просмотра таблицы маршрутов.

Пример выполнения команды:

```
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B – BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o – ODR
P - periodic downloaded static route
Gateway of last resort is 0.0.0.0 to network 0.0.0.0
C    192.168.1.0/24 is directly connected, FastEther-
net0/0 C        192.168.2.0/24 is directly connected, Se-
rial2/0
S    192.168.3.0/24 is directly connected, Seri-
al2/0 S192.168.4.0/24 is directly connected, Se-
rial2/0 S       192.168.5.0/24 is directly con-
nected, Serial2/0 S*   0.0.0.0/0 is directly
connected, Serial2/0 Router#
```

Параметр «ip protocols» команды «Show».

Данная команда используется для просмотра протоколов маршрутизации, включенных на данном устройстве.

Пример выполнения команды:

Router#show ip protocols
Routing Protocol is ‘‘ip"
Sending updates every 30 seconds, next due in 18
seconds Invalid after 180 seconds, hold down 180,
flushed after 240 Outgoing update filter list for all
interfaces is not set Incoming update filter list for all
interfaces is not set Redistributing: rip
Default version control: send version 1, receive any
version
Interface Send Recv Triggered RIP Key-
1 2
FastEther-net0/0 Seri-
1 2 1
Automatic network summarization is in
effect
Maximum path:
4 Routing for
Networks:
192.168.1.0
192.168.2.0
Gate-Distance Last Up-
192.168.2.2 120
Distance: (default is 120)
Router#

Выводится информация о включенных протоколах маршрутизации.

Команда «Ping».

Для проверки связи между устройствами сети можно использовать данную команду. Она отправляет эхо-запросы указанному узлу сети и фиксирует поступающие ответы.

Формат команды: **ping** A.B.C.D

Контекст пользователя

Команда «Enable».

Выполнение конфигурационных или управляющих команд требует вхождения в привилегированный режим, используя данную команду.

При вводе команды маршрутизатор перешел в привилегированный режим. Для выхода из данного режима используется команда disable или exit.

Также следует отметить, что в данном контексте можно пользоваться командой show для просмотра некоторой служебной информации.

Контекст конфигурирования маршрутизации

Команда «Network»

Данной командой указывают адреса сетей, которые будут доступны данному маршрутизатору.

Формат команды:

network A.B.C.D , где A.B.C.D – адрес сети

Пример выполнения команды:

Router(config-router)#network *192.168.3.0*

Данная команда означает, что пакеты, направленные в подсеть 192.168.3.0 будут отправляться через данный шлюз.

Приглашение от роутера по умолчанию будет выглядеть так: **Router>**

Это значит, что мы находимся в пользовательском режиме. Из этого режима доступно совсем немного команд. Все эти команды позволяют лишь наблюдать за работой роутера, но не дают возможности вносить изменения в конфигурацию. Из этого режима можно выполнить, например, команду **Ping или show ip interface.**

Для того, чтобы изменять рабочую конфигурацию (читай, настройку) роутера, необходимо войти в привилегированный режим. Привилегированный режим может быть защищен паролем. Для того чтобы войти в привилегированный режим, нужно набрать команду **enable.** После этого приглашение командной строки изменится на **Router#**

Здесь уже доступно намного больше команд. В этом режиме можно вносить изменения в рабочую конфигурацию и сохранять измененную конфигурацию в ПЗУ.

Но основная настройка роутера ведется из режима глобальной конфигурации. В него можно попасть из привилегированного режима выполнением команды **configure terminal**. Приглашение изменится на **Router(config)#.** Как вы уже заметили, приглашение командной строки говорит о том, в каком режиме вы находитесь.

1. соединим две сети с помощью нашего маршрутизатора.

2. Сеть Internal имеет диапазон адресов 192.168.10.1/24, адрес роутера в нем — 192.168.10.254, сетевой адаптер — FastEthernet0/0

3. Сеть External имеет диапазон адресов 10.54.0.0/16, адрес роутера в нем — 10.54.1.1, сетевой адаптер — FastEthernet0/1.

4. В режиме глобальной конфигурации вводим команду Interface FastEthernet0/0. Приглашение станет таким: **Router(config-if)#.** Интерфейс по умолчанию не имеет никакого адреса и даже выключен. Сначала введем IP-адрес. Это делается следующей командой: ip address 192.168.10.254 255.255.255.0.

5. Помните, что интерфейс выключен. Включается он командой **no shutdown.**

Если все хорошо, то пробежит надпись:

Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

6. Первая строка говорит о том, что с сетевым интерфейсом все хорошо с точки зрения физического и канального уровня (сетевой кабель подключен и

на другом его конце работает совместимое оборудование). Т.е строка говорит о готовности интерфейса на физическом уровне, для Ethernet это фактически означает, что интерфейс не отключён и контроллер порта исправен. Вторая строка говорит о том, что Сетевой уровень (IP Layer) тоже работает как надо.

7. Дальше нужно выйти из режима конфигурации интерфейса FastEthernet0/0, войти в интерфейс FastEthernet0/1 и настроить его параметры IP. С этим вы и сами справитесь.

8. Проверить, правильно ли все настроено, можно вернувшись в привилегированный режим (команда exit) и выполнив команду show ip interface brief. Она покажет информацию о состоянии сетевых интерфейсов. Вывод команды будет примерно таким:

Router#show ip interface brief

Interface IP-Address OK? Method Status Protocol FastEthernet0/0 192.168.10.254 YES manual up up FastEthernet0/1 10.54.0.1 YES manual up up

9. Готово. Роутер может передавать пакеты из одной сети в другую и обратно.

10. Все изменения и настройки, которые мы сейчас вносили, сохранены только в оперативной памяти роутера. Чтобы конфигурация сохранилась и после перезагрузки, ее нужно скопировать в ПЗУ. Делается это так - из привилегированного режима вводится команда copy running-config startup-config. Теперь перезагрузка не страшна!

11. Если вы включаете роутер, у которого отсутствует конфигурация, то IOS предложит воспользоваться визардом для настройки основных параметров работы роутера.

Содержание отчета:

1. Титульный лист.
2. Цель работы.
3. Реализация всех шагов лабораторной работы с предоставлением скриншотов.

Контрольные вопросы:

1. Типы связей.
2. Контекст пользователя.
3. Контекст администратора.

2.11. Лабораторная работа 11

Служебные утилиты для работы в Интернет. Изучение протокола HTTP.

Цель работы: Изучение структуры IP-адреса, ознакомление с наиболее популярными утилитами для диагностики сетевой конфигурации и сетевых соединений, ознакомление с основами протокола HTTP.

Ход работы:

IP-адрес состоит из двух частей: номера сети и номера узла в сети.

Самой распространенной является запись IP-адреса в виде четырех чисел, разделенных точками, каждое из которых представляет значение байта в десятичной форме, например, 213.180.204.11. Запись адреса не предусматривает специального разграничительного знака между номером сети и номером узла.

Для разделения этих частей обычно используется 2 подхода:

- С помощью маски (RFC 950, RFC 1518), представляющей собой число в паре с IP-адресом. С помощью операции «логическое И» над этими двумя числами выделяется номер сети.

- С помощью классов адресов (RFC 791).

Вводится пять классов адресов: A,B,C,D,E (табл. 1).

A,B,C – используются для адресации сетей, D и E – имеют специальное назначение. Признаком, на основании которого IP-адрес относят к тому или иному классу, являются значения нескольких первых битов адреса.

Таблица 10.1 Распределение адресов в IP сетях.

Класс	Первые биты	Наименьший номер сети	Наибольший номер сети	Максимальное число узлов
A	0	1.0.0.0 (0 - не используется)	126.0.0.0 (127 - зарезервирован)	2^{24} (3 байта)
B	10	128.0.0.0	191.255.0.0	2^{16} (2 байта)
C	110	192.0.0.0	223.255.255.0	2^{8} (1 байт)
D	1110	224.0.0.0	239.255.255.255	групповые адреса
E	11110	240.0.0.0	247.255.255.255	зарезервировано

В рамках IP протокола существуют ограничения при назначении IP-адресов, а именно

• номера сетей и номера узлов не могут состоять из двоичных нулей или единиц;

• если IP-адрес состоит только из двоичных нулей, то он называется неопределенным адресом и обозначает адрес того узла, который сгенерировал этот пакет;

• если в поле номера сети стоят только нули, то по умолчанию считается, что узел назначения принадлежит той же самой сети, что и узел, который отправил пакет; такой адрес может быть использован только в качестве адреса отправителя;

• если все двоичные разряды IP-адреса равны 1, то пакет с таким адресом назначения должен рассылаться всем узлам, находящимся в той же сети, что и источник этого пакета; такой адрес называется ограниченным широковещательным, поскольку пакет не сможет выйти за границы сети;

• если в поле адреса назначения в разрядах, соответствующих номеру узла, стоят только единицы, то пакет рассылается всем узлам сети, номер которой указан в адресе назначения; такой тип адреса называется широковещательным;

• если первый октет адреса равен 127, то такой адрес называется внутренним адресом стека протоколов; он используется для тестирования программ, организации клиентской и серверной частей приложений, установленных на одном компьютере;

• групповые адреса, относящиеся к классу D, предназначены для экономичного распространения в Интернете, большой корпоративной сети аудио- или видеопрограмм.

Стандартным классам сетей можно поставить в соответствие следующие значения маски:

• класс A – 255.0.0.0;
• класс B – 255.255.0.0;
• класс C – 255.255.255.0;

Рассмотрим следующий пример:

Исходные данные	*IP адрес*	62.76.167.21
	Маска сети	255.255.255.0
Логическая операция	И	
Результат	*Адрес сети*	62.76.167.0
	Номер компьютера	21

Для определения сетевых настроек компьютера и сетевого оборудования, диагностики и получения другой информации, относящейся к интернет-протоколам, широко используются специальные утилиты.

1. Утилита ipconfig

Ipconfig - это утилита командной строки для вывода деталей текущего соединения компьютера с сетью и контроля над клиентским сервисом DHCP. DHCP (Dynamic Host Configuration Protocol) - это сетевой протокол, позволяющий компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP.

Синтаксис команды:

Pconfig/ключи

Команда ipconfig/all - отображает полную информацию по всем сетевым адаптерам. Пример вывода для Windows:

2. Утилита *netstat*

Netstat – служебная программа, отображающая статистику протокола и текущих сетевых подключений TCP/IP:

3. Утилита telnet

Telnet - сетевой протокол для реализации текстового интерфейса по сети. Название «telnet» имеет также утилита, реализующая клиентскую часть протокола. Исторически telnet служил для удалённого доступа к интерфейсу командной строки операционных систем. Протокол telnet может использоваться для выполнения отладки других протоколов на основе транспорта TCP.

Утилита telnet поддерживает следующие команды:

• Close – закрытие текущего подключения.
• Display – отображение параметров операции.
• Open – подключение к сайту.
• Quit – выход из telnet.
• Set – установление параметров.
• Send – отправление строки на сервер.
• Status – вывод сведений о текущем состоянии.
• Unset – сброс параметров.

Используя утилиту telnet можно, например, вручную отправить запрос клиента и получить ответ сервера по протоколу HTTP.

Для этого выполним следующую последовательность действий:

1. Запуск утилиты telnet
2. Установление соединения с веб-сервером с помощью команды: open имя_хоста 80
1. Формирование запроса клиента
2. Получение ответа сервера

Пример

1. Устанавливаем соединение:*open localhost 80.*
2. Формируем строку состояния запроса клиента

GET HTTP://LOCALHOST/PERLCALC.HTML HTTP/1.0
<ENTER><ENTER>

3. Получаем ответ сервера.

Видно, что ответ веб-сервера localhost содержит строку состояния (с кодом успешного завершения 200), поля заголовка (Server, Date, Content-type и др.) и тело, содержащее HTML код запрошенного клиентом документа http://localhost/perlcalc.html.

Порядок выполнения работы

Задание 1

1. С помощью утилиты **ipconfig** (запускается в командной строке командой ipconfig) определите IP-адрес и маску подсети для своего компьютера.

2. Определите класс подсети, в которой находится ваш компьютер без использования маски подсети и по маске подсети.

3. Определите адрес подсети, в которой находится ваш компьютер, с использованием функции “Логическое И” над IP-адресом и маской подсети. Следует иметь в виду, что операция “Логическое И” должна производиться с двоичным представлением операндов.

Задание 2

С помощью утилиты ping (запускается в командной строке командой ping) проверьте доступность хостов, минимальное, среднее и максимальное время приема- передачи ICMP пакетов до них. Можно рассмотреть хосты, например, в следующей последовательности:

1. Веб-сервер Университета в Кембридже: www.cam.ac.uk;
2. Веб-сервер Университета в Калифорнии: www.ucla.edu;
3. Веб-сервер Университета в Токио: www.u-tokio.ac.jp;
4. Веб-сервер компании Майкрософт: www.microsoft.com.

Обратите внимание, что в последнем случае ICMP-пакеты блокируются веб-сервером.

Задание 3

С помощью утилиты tracert (запускается в командной строке командой tracert) определите маршруты следования и время прохождения пакетов до хостов, приведенных в задании 2.

Задание 4

1. С помощью утилиты netstat (запускается в командной строке командой netstat) посмотрите активные текущие сетевые подключения и их состояние на вашем компьютере.

2. Запустите несколько экземпляров веб-браузера, загрузив в них веб-страницы с разных веб-серверов. Посмотрите с помощью netstat, какие новые сетевые подключения появились в списке.

3. Закрывайте браузеры и с помощью netstat проверяйте изменение списка сетевых подключений.

Задание 5

1. Запустите сеанс telnet (запускается в командной строке командой telnet). При этом появится подсказка Microsoft Telnet>. С полным списком команд можно ознакомиться с помощью команды help.

2. Разрешите режим отображения вводимых с клавиатуры символов с помощью команды set localecho.

3. В соответствии с протоколом HTTP необходимо установить соединение с веб-сервером. Для этого с помощью команды open устанавливается соединение, например, open www.yandex.ru 80.

4. Сформируйте клиентский запрос. Как минимум он должен содержать строку состояния, например:

GET HTTP://WWW.YANDEX.RU/INDEX.HTML HTTP/1.0

Если поля запроса отсутствуют, то ввод заканчивается двумя нажатиями клавиши <ENTER> для вставки пустой строки после заголовка.

Следует обратить внимание на то, что при вводе нельзя допускать ошибок, поскольку при попытке их исправить с помощью клавиши <BACKSPACE>, ее нажатие интерпретируется как часть запроса.

5. Изучите полученный ответ сервера. Обратите внимание на код ответа в строке состояния ответа веб-сервера в строке состояния и поля заголовка ответа.

Если ответ сервера очень большой (в первую очередь из-за размера документа в теле ответа), то содержимое ответа сервера в окне интерпретатора

командной строки обрезается с начала. В этом случае рекомендуется для просмотра заголовка вместо метода GET использовать метод HEAD.

Содержание отчета:

1. Титульный лист.
2. Цель работы.
3. Результаты работы всех утилит и команд, представленных в лабораторной работе, с предоставлением скриншотов.

Контрольные вопросы:

1. Классовая IP адресация.
2. Утилита ipconfig.
3. Утилита netstat.
4. Утилита telnet.

2.12. Лабораторная работа 12

Проектирование простейшей сети в симуляторе Cisco Packet Tracer.

Цель работы: Получение навыков по проектировке ЛВС.

Ход работы: Как известно, локальная вычислительная сеть – это компьютерная сеть, покрывающая обычно относительно небольшую территорию или небольшую группу зданий. В нашем случае это всего-навсего 6 рабочих станций, определенным образом связанных между собой. Для этого мы будем использовать сетевые концентраторы (хабы) и коммутаторы (свитчи).

Необходимо спроектировать сеть, изображенную на рисунке 2.52.

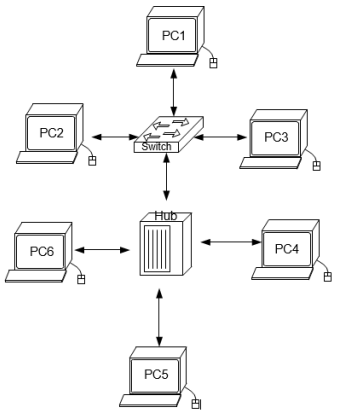


Рисунок 2.52. Проектируемая сеть

1. В нижнем левом углу Packet Tracer выбираем устройства «Сетевые коммутаторы», и, в списке справа, выбираем коммутатор 2950-24, нажимая на него левой кнопкой мыши, вставляем его в рабочую область. Так же поступает с «Сетевым концентратором (Hub-PT)» и «Рабочими станциями (PC-PT)».

2. Далее необходимо соединить устройства, как показано на рис.1, используя соответствующий интерфейс. Для упрощения выбираем в нижнем левом углу Packet Tracer 4.0 «Тип связи» и указываем «Автоматически выбрать тип соединения»: нажимая на данный значок левой кнопкой мыши, затем нажимаем на необходимое нам устройство, и соединяем с другим все тем-же нажатием.

3. Далее идет самый важный этап – настройка. Так как мы используем устройства, работающие на начальных уровнях сетевой модели OSI (коммутатор на 2ом, концентратор – на 1ом), то их настраивать не надо. Необходима лишь настройка рабочих станций, а именно: IP-адреса, маски подсети, шлюза.

Ниже приведена настройка лишь одной станции (PC1) – остальные настраиваются аналогично.

Производим двойной щелчок по нужной рабочей станции, в открывшемся окне выбираем вкладку Рабочий стол, далее – Конфигурация интерфейса, и производим соответствующую настройку:

Обратите внимание! IP-адреса всех рабочих станций должны находиться в одной и той-же подсети (то есть из одного диапазона), иначе процесс ping не выполнится.

4. Когда настройка завершена, можно переходить ко второй части работы – к запуску ping-процесса. Например, запускать его будем с PC5 и проверять наличие связи с PC1.

Важно! Вы можете сами выбрать, откуда ему запускать ping-процесс, главное, чтобы выполнялось условие: пакеты должны обязательно пересылаться через коммутатор и концентратор.

Для этого производим двойной щелчок по нужной рабочей станции, в открывшемся окне выбираем вкладку «Рабочий стол», далее – «Командная строка».

```
PC>ping 192.168.0.1
```

После ввода должна появиться следующая информация:

```
Pinging 192.168.0.1with 32 bytes of data:
Reply from 192.168.0.1: bytes=32 time=183ms TTL=120 Reply from
192.168.0.1: bytes=32 time=90ms TTL=120 Reply from
192.168.0.1: bytes=32 time=118ms TTL=120 Reply from
192.168.0.1: bytes=32 time=87ms TTL=120 Ping statistics for
192.168.0.1:

    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:

    Minimum = 87ms, Maximum = 183ms, Average = 119ms PC>
```

Это означает, что связь установлена, и данный участок сети работает исправно.

5. Перейдите в режим моделирования и инициализируйте ping-процесс снова.

Кнопка «Автоматически» подразумевает моделирование всего ping-процесса в едином процессе, тогда как «Пошагово» позволяет отображать его пошагово.

Чтобы узнать информацию, которую несет в себе пакет, его структуру, достаточно нажать правой кнопкой мыши на цветной квадрат в графе «Информация».

Моделирование прекращается либо при завершении ping-процесса, либо при закрытии окна «Редактирования» соответствующей рабочей станции.

Содержание отчета:

1. Титульный лист.
2. Цель работы.
3. Реализация всех шагов лабораторной работы с предоставлением скриншотов.
4. Скриншот результата выполнения ping процесса.

Контрольные вопросы:

1. Настройки рабочих станций.
2. Принцип работы коммутатора.
3. Принцип работы концентратора.

2.13. Лабораторная работа 13

Настройка статической маршрутизации на оборудовании Cisco

Цель работы: Изучение процессов настройки статических маршрутов на маршрутизаторах Cisco.

Схема сети (рис.12.1):

• Коммутаторы S1, S2, S3 (3 шт.);

• Маршрутизаторы R1, R2, R3 (3 шт.);

• Персональные компьютеры C1, C2, C3 (3 шт.);

• Схема сети представлена на рис. 2.53.

Задать IP адреса сетевым интерфейсам маршрутизаторов, интерфейсам управления коммутаторов и сетевым интерфейсам локальных компьютеров;

• Установить связь на физическом и канальном уровнях между соседними маршрутизаторами по последовательному сетевому интерфейсу;

• Добиться возможности пересылки данных по протоколу IP между соседними объектами сети (C1-S1, C1-R1, S1-R1, R1-R2, R2-S2, R2-C2, и т.д.);

• Настроить на маршрутизаторе R2 статические маршруты к сетям локальных компьютеров C1, C3

• Настроить на маршрутизаторах R1, R3 маршруты «по умолчанию» к сетям локальных компьютеров C2-C3 и C1-C2 соответственно;

• Добиться возможности пересылки данных по протоколу IP между любыми объектами сети (ping);

• Переключившись в «Режим симуляции» рассмотреть и пояснить процесс обмена данными по протоколу ICMP между устройствами (выполнив команду Ping с одного компьютера на другой), пояснить роль протокола ARP в этом процессе. Детальное пояснение включить в отчет.

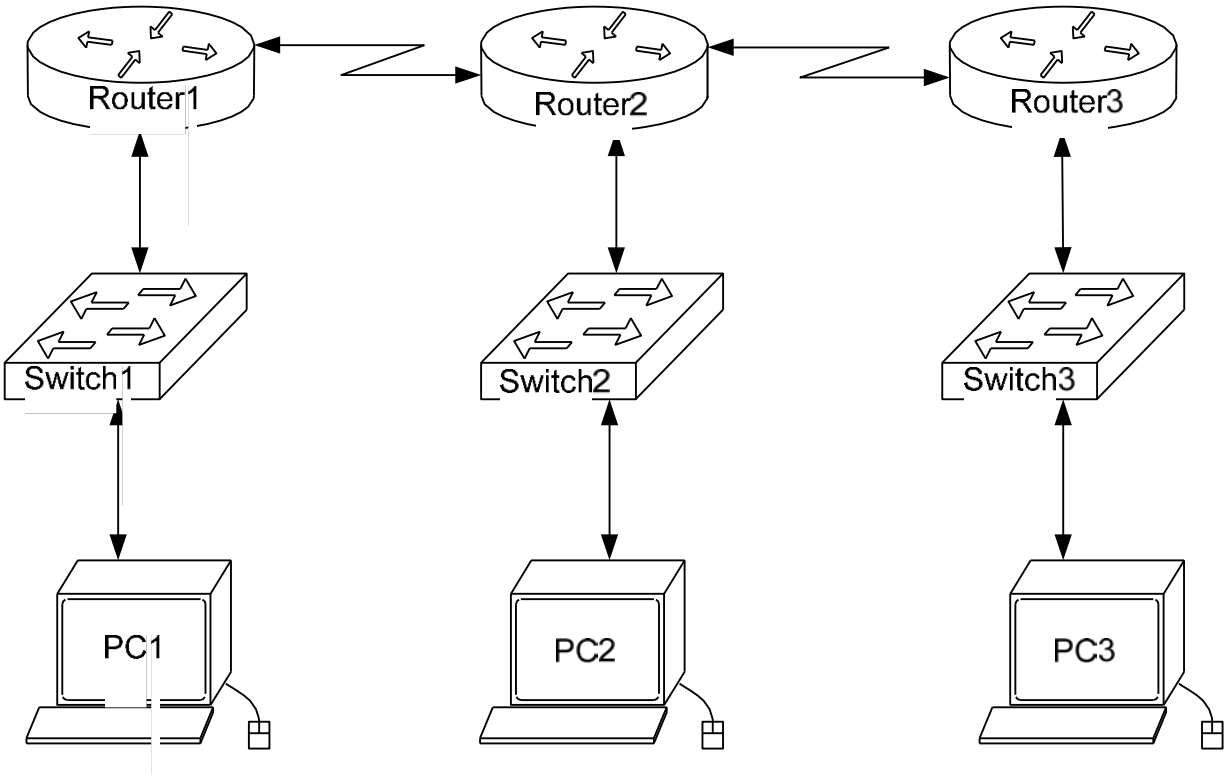


Рисунок 2.53. Схема сети

Содержание отчета:

1. Титульный лист.
2. Цель работы.
3. Модель сети.
4. Процесс настройки маршрутизации и таблица маршрутизации.
5. Скриншот результата выполнения ping процесса.

Контрольные вопросы:

1. Статическая маршрутизация.
2. Протокол TCP/IP.
3. Протокол ARP.

2.14. Лабораторная работа 14

Настройка протоколов маршрутизации RIP на оборудовании Cisco.

Целью работы:

Настройка протоколов динамической маршрутизации на оборудовании Cisco.

Ход работы:

Конфигурация сети:

• Коммутаторы S1, S2;

• Маршрутизаторы R1, R3;

• Персональные компьютеры C1, C2;

• Схеме сети выбрать на свое усмотрение

Задать IP адресасетевым интерфейсам маршрутизаторов, интерфейсам управления коммутаторов и сетевым интерфейсам локальных компьютеров;

Установить связь на физическом и канальном уровнях междусоседними маршрутизаторами по последовательному сетевому интерфейсу;

Добиться возможности пересылки данных по протоколу IP между соседними объектами сети (C1-S1, C1-R1, S1-R1, R1-R2, R2-S2, R2-C2, и т.д.);

Выявить невозможность пересылки данных по протоколу IP между удаленными объектами сети, просмотреть существующую таблицу маршрутизации;

• Включить поддержку протокола RIP на всех маршрутизаторах сети;

• Подключить к протоколу RIP требуемые сети;

• Просмотреть обновленную таблицу маршрутизации;

• Посмотреть список протоколов маршрутизации работающих на узлах сети;

• Удостовериться в возможности пересылки данных по протоколу IP между любыми объектами сети.

Содержание отчета:

1. Титульный лист.
2. Цель работы.
3. Модель сети.
4. Процесс настройки маршрутизации и таблица маршрутизации.

Контрольные вопросы:

1. Динамическая маршрутизация.
2. Протокол TCP/IP.
3. Протокол RIP.

КОНТРОЛЬ ЗНАНИЙ

1. Компьютерные сети: определение, компоненты, назначение.
2. Интерфейс, протокол, стек протоколов.
3. Модель OSI.
4. MAC-адрес.
5. IP-адрес.
6. NetBios-имя.
7. DNS-имя.
8. Стандартные топологии КС.
9. Классификация КС по территориальному признаку.
10. Линии связи: проводные и кабельные. Радиоканалы наземной и спутниковой связи.
11. Аппаратура линий связи, передачи данных.
12. Аппаратура пользователя линий связи, промежуточная аппаратура линий связи.
13. Характеристики линий связи.
14. Стандарты кабелей: медный неэкранированный, витая пара.
15. Стандарты кабелей: коаксиальный кабель, волоконно-оптический кабель.
16. Совместная среда передачи данных: протоколы случайного и поочередного доступа.
17. Протоколы передачи данных канального уровня.
18. Стандарт IEEE 802.
19. Стандарт Ethernet.
20. Стандарт Token Ring.
21. Стандарт FDDI.
22. Структура Глобальных Сетей.
23. Модель стека TCP/IP.
24. Протокол IP.
25. Структура IP адреса, классовая и бесклассовая IP адресация.
26. Протокол TCP.
27. Протокол UDP.
28. Подсети и маски подсети.
29. Протокол ICMP.
30. Служба WINS.
31. Служба DHCP.
32. Служба DNS.

Содержание

3. ВСПОМОГАТЕЛЬНЫЙ РАЗДЕЛ

УЧЕБНАЯ ПРОГРАММА

ЛОКАЛЬНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Учебная программа по учебной дисциплине «Локальные вычислительные сети» разработана для направления специальности 1-53 01 01 «Автоматизация технологических процессов и производств».

Целью изучения учебной дисциплины является подготовка специалистов в области автоматизации технологических процессов с углубленными знаниями по проектированию и настройке локальных вычислительных сетей (**ЛВС**) предприятия.

Основная задача учебной дисциплины - обучение студентов базовым методам и средствам разработки, тестирования, эксплуатации, администрирования локальных вычислительных сетей, а также элементов компьютерной сети «Интернет».

Учебная дисциплина базируется на знаниях, полученных при изучении таких дисциплин как: «Каналы передачи данных», «Электроника и схемотехника», «Микропроцессорная техника», «Технические средства автоматизации», обеспечивает базу для параллельного изучения дисциплины: «Технические средства автоматизации систем учета энергопотребления» и для выполнения соответствующего раздела дипломного проекта.

В результате изучения учебной дисциплины «Локальные вычислительные сети» студент должен:

знать:

– концепции построения локальных и глобальных сетей;
– методы объединения компьютеров и устройств в сети;
– основные функции и режимы взаимодействия компьютеров;
– наиболее эффективные методы взаимодействия в конкретной конфигурации;
– сетевой сервис широко используемых операционных систем;
– принципы разработки программ организации клиент-сервисного взаимодействия;

уметь:

– анализировать уровень эффективности сетевых решений;
– использовать операционные системы и предлагать сетевые решения для разрабатываемых

прикладных задач;

– работать с сетевым аппаратным оборудованием и программными средствами;
– использовать компьютерные сети;

владеть:

– навыками самостоятельного проектирования локальных вычислительных сетей;

– методиками диагностирования неполадок локальных вычислительных сетей;

– навыками оптимизации пропускной способности локальных вычислительных сетей.

Освоение учебной дисциплины «Локальные вычислительные сети» обеспечивает формирование следующих компетенций:

– АК-7. Иметь навыки, связанные с использованием технических устройств, управлением .

– СЛК-3. Обладать способностью к межличностным коммуникациям.

– СЛК-6. Уметь работать в команде.

– ПК-1. Разрабатывать технологию жизнеобеспечения систем автоматизации в области химико-технологических процессов, технологических процессов сбора, передачи и обработки информации энергопотребления, производств лесной, легкой, пищевой, машиностроительной, энергетической и аграрной промышленностей.

– ПК-2. Использовать современные информационные, компьютерные технологии программирования контроллеров, эксплуатировать технические средства систем автоматизации.

– ПК-10. Осуществлять выбор перспективных материалов, датчиков и приборов для обеспечения ресурсосберегающих технологических процессов.

– ПК-11. Внедрять современные микропроцессорные системы автоматизации, осуществлять переналадку оборудования. техники безопасности и противопожарной безопасности на вверенном участке работы, обучать персонал приемам безопасной работы.

– ПК-17. Анализировать и оценивать собранные данные.

– ПК-29. Заниматься научным анализом и совершенствованием современных технологий производств на основе применения средств автоматизации.

Согласно учебному плану для очной формы получения высшего образования на изучение учебной дисциплины отведено всего 152 ч., из них аудиторных - 86 часов.

Распределение аудиторных часов по курсам, семестрам и видам занятий приведено в таблице 1.

Таблица 1.

Очная форма получения высшего образования					
Курс	Семестр	Лекции, ч.	Лабораторные занятия, ч.	Практи-ческие занятия, ч.	Форма текущей аттестации
4	7	32	54	–	курсовой проект, экзамен

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Раздел 1. Основные понятия и определения ЛВС

Тема 1.1. Классификация и характеристики локальных вычислительных сетей

Основные параметры, состав и назначения компонентов компьютерных сетей.

Тема 1.2. Топологии ЛВС

Топологии: звездообразная, кольцевая, шинная, древовидная, сотовая, полносвязная: принципы работы, области применения, достоинства и недостатки.

Раздел 2. Физическая среда передачи данных

Тема 2.1. Параметры линий связи

Полоса пропускания и пропускная способность, теорема Шеннона. Методы кодирования информации в линиях связи.

Тема 2.2. Методы уплотнения линий связи

Дискретные (немодулированные) и гармонические (модулированные) системы передачи информации по линиям связи, их характеристики и области применения.

Тема 2.3. Передающая среда

Коаксиальный кабель, кабель «витая пара», волоконно-оптический кабель,
радио и инфракрасный передающие каналы. Классификация, маркировка, параметры.

Раздел 3. Методы доступа к ресурсам ЛВС

Тема 3.1. Методы доступа в типовых архитектурах ЛВС

Реализации методов доступа: Ethernet, Arcnet , Token Ring. Аппаратная реализация, основные преимущества и недостатки.

Тема 3.2. Логическое и физическое структурирование сетей.

Сетевое оборудование: интерфейсы, повторитель, мост, концентратор, маршрутизатор.

Тема 3.3. Система адресации в ЛВС

Маршрутизация и системы адресации компьютеров в ЛВС. Элементы промышленных сетей.

Раздел 4. Основы администрирования и управления в ЛВС

Тема 4.1. Методы обеспечения безопасности и сохранения данных.

Технические угрозы: Ошибки в программном обеспечении, различные DoS- и DDoS-атаки, компьютерные вирусы, черви, троянские программы, технические средства съема информации. Методы защиты: шифрование данных, электронная цифровая подпись, сеть VPN.

Тема 4.2. Защита ЛВС от компьютерных вирусов

Использование антивирусных пакетов. Архивирование информации. Резервирование информации. Ведение базы данных о вирусах и их характеристиках.

Тема 4.3. Модели администрирования и регистрации в сети.

Доменная модель. Модель рабочей группы. Учетные записи глобальные и локальные.

Тема 4.4. Функции и архитектура систем управления сетями.

Управление конфигурацией сети. Обработка ошибок, анализ производительности и надежности, учет работы сети.

Тема 4.5. Мониторинг и анализ локальных сетей

Классификация средств мониторинга и анализа. Анализаторы протоколов. Сетевые анализаторы. Протоколы SNMP, SNMPv3.

ТРЕБОВАНИЯ К КУРСОВОМУ ПРОЕКТУ

Цель: проектирование и расчет технических параметров локальных вычислительных сетей.

Исходные данные: чертеж производственных и офисных помещений предприятия, спецификация сетевого оборудования и сегментов компьютерной сети по варианту.

Содержание: обоснование конфигурации и параметров сети Ethernet и Fast Ethernet (расчёт параметров - PDV, PW), проектирование физической и логической топологии сети, распределение адресного пространства подсетей и узлов ЛВС, выбор сетевого оборудование и оценка стоимости проекта ЛВС.

Графический материал пояснительной записки должен иллюстрировать все этапы и результаты разработки.

Примерный объем работы- 30 страниц. Примерное количество часов на выполнение -60.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

№	Название раздела, темы, занятия, вопросов	Кол. ауд. часов				Материальное обеспечение	Литература	Формы контроля
		Лекции	Практические занятия	Лабораторные занятия	Управляемая самостоятельная работа			
1	2	3	4	5	6	7	8	9
1	**ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ ЛВС**	**6**		**10**				
1.1.	Классификация и характеристики локальных вычислительных сетей	2		4		ЛВС	[1]	Курсовой проект, защита ЛР
1.2.	Топологии ЛВС	4		6		ЛВС	[2,3]	Курсовой проект, защита ЛР
2	**ФИЗИЧЕСКАЯ СРЕДА ПЕРЕДАЧИ ДАННЫХ**	**6**		**12**				
2.1	Параметры линий связи.	2		4		Комп. Презент.	[4]	Курсовой проект, защита ЛР
2.2.	Методы уплотнения линий связи	2		4		Комп. Презент	[4]	Защита ЛР
2.3.	Передающая среда.	2		4		Комп. Презент	[4]	Защита ЛР
3	**МЕТОДЫ ДОСТУПА К РЕСУРСАМ ЛВС.**	10		18				
3.1.	Методы доступа в типовых архитектурах ЛВС.	4		6		Комп. Презент	[1,4]	Защита ЛР
3.2.	Логическое и физическое структурирование сетей. Сетевые ПЛК.	2		6		ЛВС	[4,5]	Защита ЛР
3.3.	Система адресации в ЛВС	4		6		ЛВС	[4,6]	Курсовой проект , защита ЛР

1	2	3	4	5	6	7	8	9
4	**ОСНОВЫ АДМИНИСТРИРОВАНИЯ И УПРАВЛЕНИЯ В КОМПЬЮТЕРНЫХ СЕТЯХ.**	10		14				
4.1.	Методы обеспечения безопасности и сохранения данных.	2		2		**Комп. Презент**	[1-4]	Курсовой проект, защита ЛР
4.2.	Защита ЛВС от компьютерных вирусов.	2		2		**Комп. Презент**	[1-4]	Защита ЛР
4.3.	Модели администрирования и регистрации в сети.	2		2		**ЛВС**	[1-4]	Курсовой проект, защита ЛР
4.4.	Функции и архитектура систем управления сетями.	2		4		**ЛВС**	[1-4]	Защита ЛР
4.5.	Мониторинг и анализ локальных сетей.	2		4		**ЛВС**	[1-4]	Защита ЛР
	Всего	32		54			[1-4]	
	Всего аудиторных часов	86						

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Примерный перечень тем лабораторных занятий

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Список литературы

Литература

1. Яшин, В.Н. Аппаратные средства персонального компьютера: Учебное пособие / В.Н. Яшин. - М.: ИНФРА-М, 2008. — 254 с.
2. Кузин А.В., Компьютерные сети: Учебное пособие / А.В. Кузин. - 3-е изд., перераб. и доп. - М. ИНФРА-М, 2011. —192 с.
3. Олифер, В.Г. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 6-е изд. / В.Г. Олифер, Н.А. Олифер. - СПб.: Питер, 2016. — 996 с.

Дополнительная литература

4. Описание и инструкция пользователя SCADA-системы фирмы «InSAT».
5. Инструкция по программированию логических программируемых контроллеров фирмы Овен. М.: Овен, 2012.
6. Таненбаум Э. Компьютерные сети. 5-е изд. / Э.Таненбаум, Д. Уэзеролл — СПб.: Питер, 2016. — 960 с.: ил.
7. Сергеев А. Н. Основы локальных компьютерных сетей: Учебное пособие /А.Н. Сергеев — СПб.: Издательство «Лань», 2016. — 184 с.: ил.

Средства диагностики результатов учебной деятельности

Оценка уровня знаний студента производится по десятибалльной шкале в соответствии с критериями, утвержденными Министерством образования Республики Беларусь.

Для оценки достижений студента рекомендуется использовать следующий диагностический инструментарий:

- устный во время лабораторных занятий;
- проведение текущих контрольных работ (заданий) по отдельным темам;
- защита выполненных на или лабораторных работ;

– собеседование при проведении индивидуальных и групповых консультаций;

– защита курсового проекта;

– сдача экзамена.

Перечень тем лабораторных занятий

Лабораторная работа 1. Изучение программных средств тестирования параметров соединения в компьютерных сетях и проверки настройки протокола tcp/ip

Лабораторная работа 2. Ознакомление с интерфейсом программы Netemul. Соединение ЭВМ в сеть

Лабораторная работа 3. Маршрутизаторы в Netemul

Лабораторная работа 4. Разрешение адресов по протоколу **arp**

Лабораторная работа 5. Динамическая маршрутизация по протоколу **rip**. Получение сетевых настроек по DHCP

Лабораторная работа 6. Преобразование десятичных чисел в двоичные и двоичных в десятичные

Лабораторная работа 7. Классификация способов сетевой адресации

Лабораторная работа 8. Вычисление масок подсети

Лабораторная работа 9. Знакомство с сетевым симулятором Cisco Packet Tracer

Лабораторная работа 10. Соединение двух сетей

Лабораторная работа 11. Служебные утилиты для работы в интернет. Изучение протокола http

Лабораторная работа 12. Проектирование простейшей сети в симуляторе Cisco Packet Tracer

Лабораторная работа 13. Настройка статической маршрутизации на оборудовании Cisco

Лабораторная работа 14. Настройка протоколов маршрутизации rip на оборудовании Cisco

Список компьютерных программ

1. Симулятор ЛВС NETEMUL.
2. Симулятор и система проектирования ЛВС Packet Tracer фирмы Cisco.
3. Симулятор и пакет программ CodeSys фирмы Овен.

4. Симулятор и пакет программ MasterSCADA фирмы InSAT.

Учебный стенд:

Компьютерный сетевой комплекс локальной вычислительной сети **ЛВС** учебной аудитории.

Перечень тем курсовых работ

Проект локальной вычислительной сети (ЛВС) предприятия по вариантам.

Перечень контрольных вопросов и заданий для самостоятельной работы студентов

1. Компьютерные сети: определение, компоненты, назначение.
2. Интерфейс, протокол, стек протоколов.
3. Модель OSI.
4. MAC-адрес.
5. IP-адрес.
6. NetBios-имя.
7. DNS-имя.
8. Стандартные топологии ЛВС.
9. Классификация ЛВС по территориальному признаку.
10. Линии связи: проводные и кабельные. Радиоканалы наземной и спутниковой связи.
11. Аппаратура линий связи, передачи данных.
12. Аппаратура пользователя линий связи, промежуточная аппаратура линий связи.
13. Характеристики линий связи.
14. Стандарты кабелей: медный неэкранированный, витая пара.
15. Стандарты кабелей: коаксиальный кабель, волоконно-оптический кабель.
16. Совместная среда передачи данных: протоколы случайного и поочередного доступа.
17. Протоколы передачи данных канального уровня.
18. Стандарт IEEE 802.
19. Стандарт Ethernet.

20. Стандарт Token Ring.
21. Стандарт FDDI.
22. Структура Глобальных Сетей.
23. Модель стека TCP/IP.
24. Протокол IP.
25. Структура IP адреса, классовая и бесклассовая IP адресация.
26. Протокол TCP.
27. Протокол UDP.
28. Подсети и маски подсети.
29. Протокол ICMP.
30. Служба WINS.
31. Служба DHCP.
32. Служба DNS.

Методические рекомендации по организации и выполнению самостоятельной работы студентов

При изучении дисциплины рекомендуется использовать следующие формы самостоятельной работы:

– решение индивидуальных заданий (задач);

– подготовка рефератов по индивидуальным темам;

– подготовка сообщений, тематических докладов, презентаций по заданным темам;

– изготовление макетов;

– составление тематической подборки литературных источников, интернет-источников;

– проработка тем (вопросов), вынесенных на самостоятельное изучение;

– подготовка курсовой работы по индивидуальным заданиям.

Критерии оценки результатов учебной деятельности

Баллы	Критерии оценки
1 (один)	Отсутствие приращения знаний и компетентности в рамках дисциплины; отказ от ответа
2 (два)	Фрагментарные знания в рамках дисциплины; знание отдельных литературных источников, рекомендованных учебной программой дисциплины; неумение использовать научную терминологию дисциплины, наличие в ответе грубых ошибок; пассивность на практических и лабораторных занятиях, низкий уровень культуры исполнения заданий
3 (три)	Недостаточно полный объем знаний в рамках дисциплины; знание части основной литературы, рекомендованной учебной программой дисциплины; использование научной терминологии, изложение ответа на вопросы с существенными ошибками; слабое владение инструментарием учебной дисциплины, неумение ориентироваться в основных теориях, методах и направлениях дисциплины; пассивность на практических и лабораторных занятиях; низкий уровень культуры исполнения заданий
4 (четыре)	Достаточный объем знаний в рамках дисциплины; усвоение основной литературы, рекомендованной учебной программой дисциплины; использование научной терминологии, логическое изложение ответа на вопросы, умение делать выводы без существенных ошибок; владение инструментарием учебной дисциплины, умение под руководством преподавателя решать стандартные (типовые) задачи; умение ориентироваться в основных теориях, методах и направлениях дисциплины и давать им оценку; работа под руководством преподавателя на практических и лабораторных занятиях, допустимый уровень культуры исполнения заданий
5(пять)	Достаточные знания в объеме учебной программы; использование научной терминологии, грамотное, логически правильное изложение ответа на вопросы, умение делать выводы; владение инструментарием учебной дисциплины, умение его использовать в решении учебных задач; способность самостоятельно применять типовые решения в рамках учебной программы; усвоение основной литературы, рекомендованной учебной программой дисциплины; умение ориентироваться в теориях, методах и направлениях дисциплины и давать им сравнительную оценку; самостоятельная работа на практических и лабораторных занятиях, фрагментарное участие в групповых обсуждениях, достаточный уровень культуры исполнения заданий
6(шесть)	Достаточно полные и систематизированные знания в объеме учебной программы; использование необходимой научной терминологии, грамотное, логически правильное изложение ответа на вопросы, умение делать обобщения и обоснованные выводы; владение инструментарием учебной дисциплины, умение его использовать в решении учебных задач; способность самостоятельно применять типовые решения в рамках учебной программы; усвоение основной литературы, рекомендованной учебной программой дисциплины; умение ориентироваться в теориях, методах и направлениях дисциплины и давать им сравнительную оценку; самостоятельная работа на практических и лабораторных занятиях, периодическое участие в групповых обсуждениях, достаточно высокий уровень культуры исполнения заданий
7(семь)	Систематизированные, глубокие и полные знания по всем разделам учебной программы; использование научной терминологии, грамотное, логически правильное изложение ответа на вопросы, умение делать обоснованные выводы и обобщения; владение инструментарием учебной дисциплины, умение его использовать в постановке и решении научных задач; свободное владение типовыми решениями в рамках учебной программы; усвоение основной и дополнительной литературы, рекомендованной учебной программой дисциплины; умение ориентироваться в основных теориях, методах и направлениях дисциплины и давать им аналитическую оценку; активная самостоятельная работа на практических и лабораторных занятиях, участие в групповых обсуждениях, высокий уровень культуры исполнения заданий

8(восемь)	Систематизированные, глубокие и полные знания по всем поставленным вопросам в объеме учебной программы; использование научной терминологии, грамотное и логически правильное изложение ответа на вопросы, умение делать обоснованные выводы и обобщения; владение инструментарием учебной дисциплины, умение его использовать в постановке и решении научных задач; способность самостоятельно решать сложные проблемы в рамках учебной программы; усвоение основной и дополнительной литературы, рекомендованной учебной программой дисциплины; умение ориентироваться в теориях, методах и направлениях дисциплины и давать им аналитическую оценку; активная самостоятельная работа на практических и лабораторных занятиях, систематическое участие в групповых обсуждениях, высокий уровень культуры исполнения заданий
9 (девять)	Систематизированные, глубокие и полные знания по всем разделам учебной программы; точное использование научной терминологии, грамотное, логически правильное изложение ответа на вопросы; владение инструментарием учебной дисциплины, умение его эффективно использовать в постановке и решении научных задач; способность самостоятельно и творчески решать сложные проблемы в нестандартной ситуации в рамках учебной программы; полное усвоение основной и дополнительной литературы, рекомендованной учебной программой дисциплины; умение ориентироваться в теориях, методах и направлениях дисциплины и давать им аналитическую оценку; систематическая активная самостоятельная работа на практических и лабораторных занятиях, творческое участие в групповых обсуждениях, высокий уровень культуры исполнения заданий
10(десять)	Систематизированные, глубокие и полные знания по всем разделам учебной программы, а также по основным вопросам, выходящим за ее пределы; точное использование научной терминологии, грамотное, логически правильное изложение ответа на вопросы; безупречное владение инструментарием учебной дисциплины, умение его эффективно использовать в постановке и решении научных задач; выраженная способность самостоятельно и творчески решать сложные проблемы в нестандартной ситуации; полное и глубокое усвоение основной и дополнительной литературы по учебной дисциплине; умение свободно ориентироваться в теориях, методах и направлениях дисциплины и давать им аналитическую оценку, использовать научные достижения других дисциплин; самостоятельная творческая работа на практических и лабораторных занятиях, активное творческое участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

Содержание

Список литературы

1. Уэнделл Одом Официальное руководство Cisco по подготовке к сертификационным экзаменам CCNA ICND1, 2-е издание, Уэнделл Одом, 572 стр., с ил. CD-ROM; серия Cisco Press; 2011, Вильямс.

2. Васин Н.Н. Сети и системы передачи информации на базе коммутаторов и маршрутизаторов Cisco Самара: ПГАТИ, 2008. 230 с

3. Уэнделл Одом Официальное руководство Cisco по подготовке к сертификационным экзаменам CCNA ICND2, 2-е издание, Уэнделл Одом, 736 стр., с ил.; CD-ROM; серия Cisco Press; 2012, Вильямс.

4. Дэвид Хьюкаби «Маршрутизаторы Cisco. Руководство по конфигурированию», 2-е издание, Дэвид Хьюкаби, Стив Мак-Квери, Эндрю Уайтейкер, 736 стр., «ВИЛЬЯМС», 2012

6. Фокин В.Г. Оптические системы передачи и транспортные сети. –М.: Эко- Трендз, 2008. - 288с.

7. Олифер В.Г., Олифер Н.А Компьютерные сети. Принципы, технологии, протоколы СПб: Издательство «Питер», 2006. 958 с

8. Фриман Р. Волоконно-оптические сети. -3-е издание. –М.: Техносфера, 2007. - 496с.

9. Фокин В.Г. Малинкин В.Б. Технологии транспортных сетей последнего поколения. Учебное пособие УМО. – Новосибирск, СибГУТИ, 2006. –132с.

10. Безопасность в электросвязи и информационных технологиях. Обзор содержания и применения действующих Рекомендаций МСЭ-Т для обеспечения защищенной электросвязи. – ITU, 2006. -130с.

11. Программа сетевой академии Cisco CCNA 1 и 2. Вспомогательное руководство М.: Издательский дом «Вильямс», 2005. 1168 с.

12. Программа сетевой академии Cisco CCNA 3 и 4. Вспомогательное руководство М.: Издательский дом «Вильямс», 2006. 1000 с.

13. Новиков Ю.В., Кондратенко С.В. Основы локальных сетей Интернет- университет информационных технологий - ИНТУИТ.ру, 2005

14. Олифер В.Г., Олифер Н.А. Основы сетей передачи данных Интернет- университет информационных технологий - ИНТУИТ.ру, 2005

15. Лабораторный практикум. Работа в эмуляторе NETEMUL и Cisco Packet Tracer / С.С. Владимиров – Санкт-Петербург: СПбГУТ, 2014. – 24с.

Содержание