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

СОГЛАСОВАНО
Заведующий кафедрой
_____Г.Н. Здор
___ _____ 2017 г.

СОГЛАСОВАНО
Декан факультета
_____Е.Е. Трофименко
___ _____ 2017 г.

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

Компьютерные сети

для специальности:
I – 53 01 06 «Промышленные роботы и робототехнические комплексы»

Составитель: Дубинин С.В.

Рассмотрено и утверждено
на заседании совета факультета информационных технологий
и робототехники 25 мая 2017 г.,
протокол N 9

Перечень материалов

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

Пояснительная записка

Электронный учебно-методический комплекс разработан для студентов специальности I – 53 01 06 «Промышленные роботы и робототехнические комплексы». Информационное наполнение ЭУМК соответствует программе дисциплины «Программное управление технологическим оборудованием». Комплекс предназначен для студентов дневного и заочного отделений.

Внедрение ЭУМК будет способствовать более эффективному овладению теоретическими и практическими основами разработки, настройки и администрированию компьютерных 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 «Компьютерные сети».

СОДЕРЖАНИЕ

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1. Введение в организацию компьютерных сетей

Вопросы для изучения:

- Компоненты компьютерных сетей. Задачи проектирования компьютерных сетей;
- Уровневая организация взаимодействия по сети;
- Адресация узлов сети. Разрешение адресов;
- Стандартные топологии КС;
- Способы классификации КС.

Компоненты компьютерных сетей. Задачи проектирования компьютерных сетей

Компьютерные сети представляют собой набор узлов – компьютеров и маршрутизаторов, которые объединяются линиями связи. Линии связи включают кабельные системы и сетевое оборудование. Для организации взаимодействия по сети используются прикладные сетевые программы, которые имеют клиент - серверную архитектуру (Web-серверы и браузеры, сетевые СУБД и т.д.). Сетевые операционные системы используются для управления разделяемыми ресурсами сети и обеспечения безопасности (Unix, Linux, Windows). Сетевые программы обмениваются сообщениями по определенным стандартам, которые называются протоколами. Протокол представляет собой «язык», который должны понимать обе стороны. К задачам проектирования КС относят следующие вопросы:

1. Проектирование на аппаратном уровне – выбор соответствующего сетевого оборудования проектирование и монтаж кабельной системы, настройка интеллектуального оборудования. Зависит от выбора стандартной технологии.
2. Выбор транспортных протоколов, которые будут использоваться в сети и поддерживаться всеми узлами сети. Задачи маршрутизации. Адресация узлов сети. Настройка сетевых служб (WINS, DNS, DHCP, и т.д.).
3. Проектирование логической структуры сети (иерархическая – домен или одноранговая сеть, где все узлы равноправны). Обеспечение безопасности.
4. Проектирование пользовательских сетевых приложений (Почта, интернет, сетевые базы данных, файловые серверы и т.д.).

Уровневая организация взаимодействия по сети

Чтобы упростить решение задачи взаимодействия по сети, ее разбивают на несколько уровней. Каждый уровень отвечает за выполнение сво-

их собственных функций. Уровни образуют иерархию, в которой модуль любого уровня взаимодействует только с модулями соседних уровней. При этом нижележащие уровни предоставляют услуги вышележащим.

Интерфейс – это соглашение, которое определяет правила взаимодействия модулей двух соседних уровней.

При передаче данных по сети между двумя узлами выполняется обмен сообщениями между модулями их соответствующих уровней: например, один уровень отвечает за установление и разрыв соединения, второй – за шифрование/дешифрование данных, а третий – это приложение, которое генерирует, принимает, анализирует сообщения.

Протокол описывает правила взаимодействия модулей одного уровня по сети. Протокол определяет порядок обмена сообщениями, виды и формат сообщений, выполняет проверку правильности доставки сообщений. Протокол представляет собой стандарт, которому должны следовать разработчики ПО для совместимости сетевых приложений разных производителей.

Стеком протоколов называют иерархически упорядоченный набор протоколов, каждый из которых необходим в процессе обмена данными по сети. Примеры стеков протоколов: TCP/IP, IPX\SPX, NetBeui.

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.

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.).

Адресация узлов сети. Разрешение адресов

Для успешной передачи данных между отправителем и получателем, необходимо чтобы как отправитель, так и получатель имели адреса, уникально идентифицирующие их в сети. Каждый протокол использует собственный способ адресации узлов. Поскольку на одном и том же узле на разных уровнях функционируют различные протоколы, то один и тот же узел идентифицируется различными типами адресов, которые используются этими протоколами. При этом каждый протокол «знает» только свой собственный способ адресации.

При взаимодействии нескольких протоколов различных уровней в процессе передачи, возникает проблема установления соответствия между различными типами адресов одного и того же компьютера.

Для решения этой задачи используются протоколы разрешения адресов. Выделяют две схемы адресации узлов: плоскую и иерархическую. Адрес, который относится к плоской схеме адресации, уникальным образом определяет узел, но не дает никакой информации о его местонахождении. Адреса иерархической схемы состоят из нескольких частей и позволяют определить местоположение узла.

Наиболее используемыми являются следующие типы адресов:

Аппаратный или **MAC** – адрес компьютера. Состоит из 6 байт представленных в 16-ричном формате. «Зашивается» в сетевом адаптере производителем адаптера. Первые три байта представляют собой идентификатор производителя, последние три байта присваиваются производителем. Такой адрес уникально идентифицирует узел в сети, но не дает никакой информации о его местонахождении и относится к плоской схеме адресации.

Пример: «0A-C0-FF-1G-12-35».

IP- адрес. Адрес протокола IP стека протоколов TCP/IP. Состоит из 4 байт, представленных десятичными числами. Состоит из номера узла и номера сети. Какая часть относится к номеру узла, а какая к номеру сети можно определить либо по классу адреса, либо с использованием маски подсети.

Пример: «10.10.5.200».

NetBios - имя. Используется для идентификации приложений.

Состоит из 16 символов, последний символ используется для определения того каким приложением было зарегистрировано имя. Относится к плоской схеме адресации и имеет привязку к физическому адресу компьютера.

Пример: «M32».

DNS - имя. Представляет собой символьное составное имя, образованное из имени узла и иерархии имен доменов интернет, в которые входит узел.

Пример: «M32.ziet.zt.ua».

Для определения IP- адреса узла по его DNS – имени используется служба доменных имен интернет DNS. Для определения IP – адреса узла по известному NetBios – имени используется служба WINS. Для нахождения аппаратного MAC – адреса по известному IP – адресу используется протокол ARP. Протокол RARP выполняет обратную задачу.

Стандартные топологии КС

Различают два типа топологий компьютерных сетей: логическую и физическую. Логическая топология задает способ передачи данных по сети между узлами. Физическая топология определяет, как компьютеры соединены между собой при помощи кабеля. Основные виды стандартных топологий отображены на рисунке 1.1.

Способы классификации КС

Существует множество различных классификаций компьютерных сетей: в зависимости от используемой топологии, по используемым сте-кам транспортных протоколов и т.д. Наиболее распространена класси-фикация по территориальному признаку.

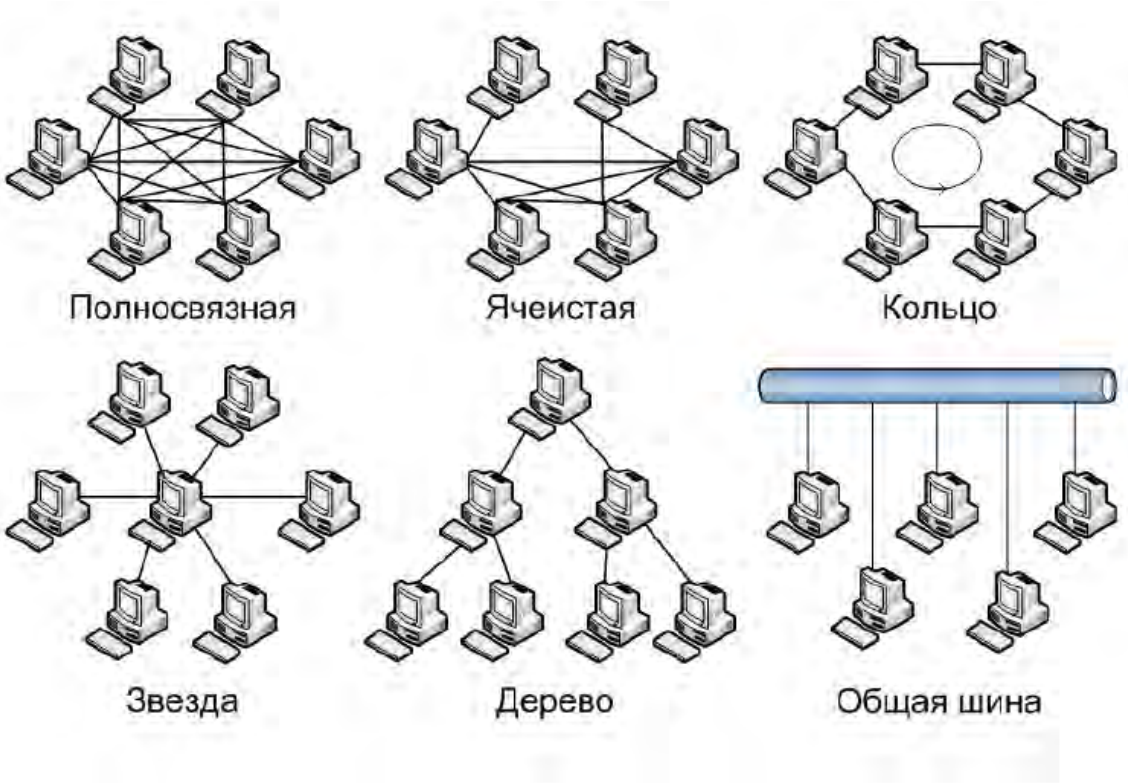

Рисунок 1.1 – Топологии компьютерных сетей

Локальные Сети (Local Area Network -LAN) – сети, которые находятся под единым административным управлением, радиус сети – до нескольких километров, скорость передачи данных высокая.

Глобальные сети (Wide Area Network – WAN) – территориально не ограничены. Из-за использования телефонных линий связи скорость передачи данных сравнительно невысока.

Городские сети (Metropolitan Area Network - MAN) – используются для обслуживания территории города. Городские сети объединяют локальные сети в пределах города и в качестве магистральной линии часто используют оптоволоконные линии связи. Скорость передачи данных достаточно высока. (Содержание)

1.2. Линии связи и структурированные кабельные системы

Вопросы для изучения:

- Линии связи, аппаратура и характеристики линий связи;
- Стандарты кабелей;
- Структурированные кабельные системы. Стандарты СКС. Подсистемы СКС.

Линии связи, аппаратура и характеристики линий связи.

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).

Стандарты кабелей

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.

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. (Содержание)

1.3. Передача данных на канальном и физическом уровнях модели ISO/OSI

Вопросы для изучения:

- Совместная среда передачи данных. Способы доступа к совместной среде передачи данных;
- Протоколы разделения канала;
- Протоколы случайного доступа;
- Протоколы поочередного доступа (Taking –Turns Protocols);
- Протоколы передачи данных канального уровня;
- Передача данных на физическом уровне.

Совместная среда передачи данных. Способы доступа к совместной среде передачи данных

На канальном уровне семиуровневой модели взаимодействия открытых систем рассматривают две задачи:

1. Определение метода доступа к среде передачи данных;
2. Передача данных между двумя узлами сети.

Метод доступа к среде передачи данных позволяет определить какая из станций сети может вести передачу данных в определенный момент времени.

Существует два типа соединений – «точка - точка» и широковещательное. В соединениях **«точка-точка»** каждая пара узлов соединяется единственным каналом связи, и данные перемещаются между этими парами узлов последовательно от отправителя к получателю.

Широковещательное соединение использует общий канал передачи данных для соединения множества узлов. При передаче данных каждый узел получает свою копию данных. В широковещательных сетях возникает проблема совместного доступа, когда несколько станций пытаются вести передачу данных одновременно. В этом случае происходит коллизия, или

т.н. «крах пакета». Чтобы регулировать процесс передачи данных, в широковещательных сетях используются протоколы совместного доступа.

Протоколы совместного доступа разделяют на следующие группы:

- протоколы разделения канала;
- протоколы случайного доступа;
- протоколы поочередной передачи (taking – turns).

Протоколы разделения канала

Для совместного использования канала используются три техники:

- временное разделение канала (Time Division Multiplexing, TDM);
- частотное разделение канала (Frequency Division Multiplexing, FDM);
- кодовое разделение канала (Code Division Multiple Access, CDMA).

Техника временного разделения канала делит все время на временные интервалы - кадры. Каждый узел получает свой такт времени внутри временного кадра, в течение которого он может вести передачу данных. При частотном разделении каждому узлу выделяется свой частотный диапазон, внутри которого он может передавать данные.

В случае кодового разделения канала каждой станции присваивается свой код. Каждый бит передаваемой информации кодируется этим кодом. Станция – получатель может выделить сообщение из общего потока данных, зная код станции – отправителя.

Протоколы случайного доступа

Позволяют вести передачу данных в любой момент времени. При этом возможно возникновение коллизий.

Классическая Aloha – первый алгоритм случайного доступа к среде передачи данных, был разработан в Гавайском университете.

Каждая станция может начинать передачу данных в любой момент времени. Если несколько станций передают данные одновременно, то возникает конфликт – коллизия.

Тактированная Aloha представляет собой усовершенствованный вариант классической схемы Aloha. Все время разбивается на равные интервалы – такты. Все станции имеют право начинать передачу данных только в начале временного такта. Если происходит конфликт, то в коллизии участвуют лишь те станции, которые передают данные внутри одного временного интервала.

Метод доступа с прослушиванием несущей (Carrier Sense Multiply Access, CSMA). Для увеличения вероятности успешной передачи станция проверяет канал на наличие сигнала несущей частоты. При отсутствии сигнала делается вывод о том, что линия свободна, и станция начинает передачу данных.

Метод доступа с прослушиванием несущей и определением коллизий (Carrier Sense Multiple Access/ Collision Detection, CSMA/CD). Станция прослушивает канал как до момента передачи данных, так и во время передачи данных. Если обнаруженный в канале сигнал отличается от данных, которые передает станция, то регистрируется коллизия. Повторная попытка передачи данных выполняется через некоторый случайный интервал времени.

Протоколы поочередного доступа (Taking –Turns Protocols)

Характеризуются отсутствием коллизий являются детерминистическими протоколами, т.е. протоколами, которые позволяют рассчитать время, за которое кадр данных попадет от отправителя к получателю. К методам поочередного доступа относятся следующие виды протоколов:

- протоколы с выбором станции (Polling Protocols);
- протоколы с передачей маркера (Token Pass Protocols).

В протоколах выбора присутствует центральная станция, которая поочередно обращается ко всем узлам сети и предоставляет им право вести передачу данных. Наличие центральной станции является основным недостатком таких сетей.

В протоколах с передачей маркера между станциями поочередно передается кадр специального формата – маркер. Станция, получившая маркер, получает право передавать свои данные.

Если данные для передачи отсутствуют, то маркер передается дальше.

При наличии данных маркер изымается из сети, и станция передает свой кадр данных. После завершения передачи станция генерирует новый маркер.

Протоколы передачи данных канального уровня

На канальном уровне выполняется передача данных, которые поступают от протокола верхнего уровня. Данные оформляются в кадры определенного формата, к ним добавляется управляющая информация, необходимая для передачи данных на канальном уровне. Передача данных по протоколам канального уровня выполняется либо в пределах локальной сети с заданной топологией (Ethernet, Token Ring, FDDI), либо в глобальных сетях по соединению «точка - точка».

Протоколы канального уровня можно классифицировать следующим образом:

- синхронные протоколы;
- асинхронные протоколы;
- символьно - ориентированные протоколы;
- бит - ориентированные протоколы;

- протоколы с установлением соединения;
- датаграммные протоколы;
- протоколы с обнаружением искажений и потерь данных;
- протоколы с восстановлением данных;
- протоколы с использованием сжатия данных.

В протоколах синхронной передачи данные собираются в кадр. Кадр данных предваряется байтом синхронизации. Байт синхронизации содержит определенный код, который оповещает приемник о приходе данных.

Асинхронные протоколы сопровождают каждый байт информации сигналами “старт” и “стоп”. Сигналы используются для оповещения приемника о приходе данных и выполнения действий по синхронизации. Такой режим называют асинхронным потому, что появление следующего байта может быть смещено во времени.

Синхронные протоколы делят на бит - ориентированные и символьно - ориентированные.

В символьно - ориентированных протоколах для синхронизации используются символы, а в бит – ориентированных используется набор битов, называемый флагом.

Протоколы с установлением соединения предоставляют услуги по надежной передаче данных. До начала передачи станция – инициатор соединения посылает кадр с запросом на установление соединения. В ответ станция – получатель отправляет кадр с подтверждением установления соединения и параметрами соединения. Станция – инициатор может отправить кадр с подтверждением того, что параметры соединения приняты. Только после такого обмена сообщениями логическое соединение установлено и может производиться обмен данными. При разрыве соединения станция – инициатор разрыва отправляет другой стороне соответствующее уведомление.

Датаграммные протоколы предоставляют услуги по ненадежной доставке данных. Данные отсылаются без предупреждения и протокол не отвечает за их доставку. Датаграммные протоколы работают достаточно быстро, т.к. не выполняет никаких действий при отправке данных.

Передача данных на физическом уровне

Различают два способа передачи информации:

1. Аналоговая модуляция;
2. Цифровое кодирование.

Аналоговая модуляция – используется при передаче данных по телефонным линиям связи (узкополосные каналы связи). Сигнал имеет синусоидальную форму.

Для кодирования информации используются три способа:

1. Амплитудная модуляция, т.е. изменение амплитуды сигнала несущей частоты;
2. Частотная модуляция, т.е. изменение частоты сигнала;
3. Фазовая модуляция, т.е. изменение фазы сигнала.

Цифровое кодирование – способ представления информации в виде прямоугольных импульсов. Различают два способа цифрового кодирования:

1. Потенциальное кодирование – для представления нулей и единиц используются только значения потенциала сигнала, а его перепады игнорируются;
2. Импульсное кодирование – позволяет представлять данные перепадом потенциала определенного направления. (Содержание)

1.4. Технологии локальных сетей

Вопросы для изучения:

- Стандарты IEEE 802;
- Технология Ethernet;
- Технология Token Ring;
- Технология FDDI.

Стандарты IEEE 802

В 1980г. В институте IEEE был организован комитет 802 целью которого была разработка стандартов локальных сетей. Эти стандарты описывают функционирование локальных сетей на физическом и канальном уровнях.

Канальный уровень делится на два подуровня: уровень логического управления каналом (Logical Link Layer, LLC) и уровень управления доступом к среде передачи данных (Media Access Control, MAC).

Уровень MAC выполняет синхронизацию доступа к совместной среде передачи данных и определяет в какой момент времени станция может начинать передавать имеющиеся данные.

После того как получен доступ к среде, выполняется передача данных в соответствии со стандартами, которые определены на уровне LLC. Уровень LLC отвечает за связь с сетевым уровнем, а также выполняет передачу данных с заданной степенью надежности.

На уровне LLC используются три процедуры передачи данных:

1. LLC1 – передача данных с установлением соединения и подтверждением приема;
2. LLC2 – передача данных без установления соединения и подтверждения приема;

3. LLC3 – передача данных без установления соединения, но с подтверждением приема данных.

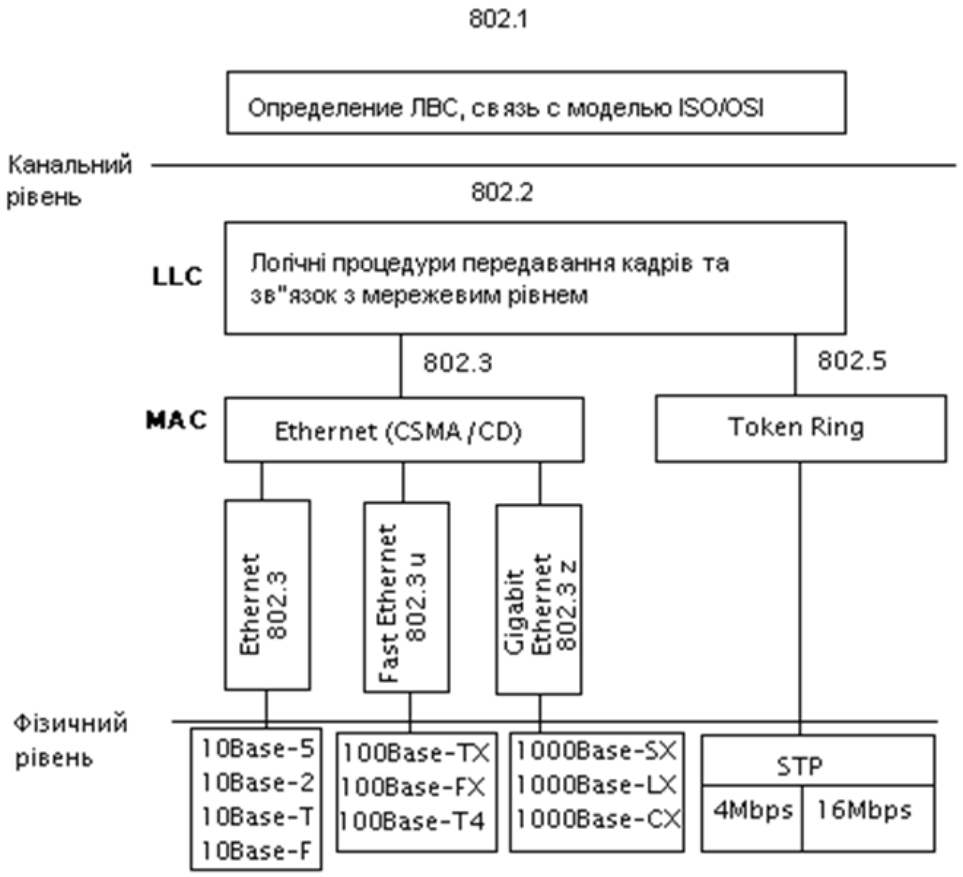

Рисунок 1.2 – Стандарт IEEE 802

На уровне LLC используются три процедуры передачи данных:

1. LLC1 – передача данных с установлением соединения и подтверждением приема;
2. LLC2 – передача данных без установления соединения и подтверждения приема;
3. LLC3 – передача данных без установления соединения, но с подтверждением приема данных.

Протоколы LLC и MAC взаимно независимы – каждый протокол уровня MAC может применяться с любым протоколом уровня LLC и наоборот.

Стандарт 802.1 описывает общие понятия локальных сетей, определяет связь трех уровней стандартов 802 с семиуровневой моделью, а также стандарты построения сложных сетей на основе базовых топологий (internetworking).

К этим стандартам относят стандарты, описывающие функционирование моста/коммутатора, стандарты объединения разнородных сетей при

помощи транслирующего моста, стандарты построения виртуальных сетей(VLAN) на основе коммутаторов.

Технология Ethernet

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

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.

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.

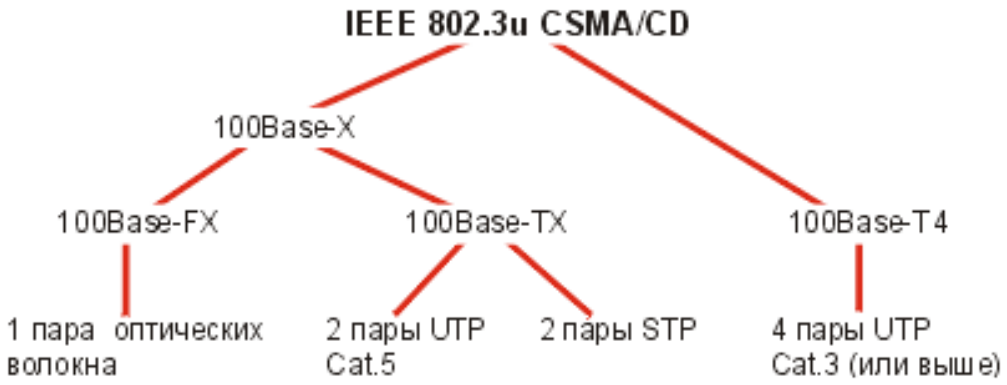

Рисунок 1.3 – Стандарт Ethernet

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

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, по-

сле 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 FDDI

Fiber Distributed Data Interface – Оптоволоконный интерфейс распределенных данных, разработан институтом ANSI с 1986 по 1988г. Является первой технологией локальных сетей, в которой используется оптоволокно. Для повышения безотказности FDDI строится на базе двух оптоволоконных колец, которые образуют основной и резервный пути прохождения данных. Для обеспечения надежности узлы подключают к обоим кольцам. В нормальном режиме работы данные проходят только по первичному кольцу. Если произошел отказ и часть первичного кольца не может передавать данные, то выполняется операция свертывания кольца – то есть объединение первичного кольца с вторичным и образование единого кольца. В сетях FDDI используется маркерный метод доступа к среде передачи данных, который работает на основе алгоритма с ранним освобождением маркера. Технология FDDI поддерживает передачу двух видов трафика – синхронного (звук, видео) и асинхронного(данные). Тип данных определяется станцией. Маркер всегда может быть захвачен на определенный интервал времени для передачи синхронных кадров и лишь в случае отсутствия перегрузок кольца – для передачи асинхронного кадра.

Максимальное число станций с двойным подключением в кольце составляет 500, максимальная длина кольца – 100км. Максимальное расстояние между двумя соседними узлами равно 2км. (Содержание)

1.5. Стандарты глобальных сетей

Вопросы для изучения:

- Структура глобальных сетей;
- Типы глобальных сетей;
- Маршрутизация в глобальных сетях.

Структура глобальных сетей

Глобальные сети предоставляют свои услуги большому количеству абонентов, которые разбросаны по неограниченной территории. Глобаль-

ные сети могут создаваться крупными телекоммуникационными компаниями для оказания платных услуг абонентам (т.н. **публичные или общественные сети**). Если создателем и владельцем сети является крупная корпорация, которая использует эту сеть то сеть называют **частной**.

Оператор сети – это компания, которая поддерживает работоспособность сети.

Поставщик услуг (провайдер) – это компания, которая оказывает платные услуги абонентам сети.

Глобальная сеть строится на основе некоммутируемых (выделенных каналов связи), которыми соединяются коммутаторы сети между собой. Коммутаторы называют центрами коммутации пакетов. Коммутаторы устанавливают там, где производится слияние или разделение потоков данных конечных абонентов или магистральных каналов. Абоненты сети подключаются с помощью выделенных каналов связи, скорость которых ниже чем скорость магистральных каналов, или по коммутируемому соединению (через телефонную сеть).

Конечные узлы глобальной сети включают: маршрутизаторы, локальные сети, компьютеры, мультиплексоры (для одновременной передачи голоса и данных).

Конечные узлы представляют собой устройства, которые вырабатывают данные и относят к устройствам DTE (Data Terminal Equipment) – оконечное оборудование данных.

Чтобы конечные узлы могли передавать данные по каналу связи определенного стандарта, они оснащаются устройствами DCE (Data Circuit Equipment), которые обеспечивают протокол физического уровня данного канала.

Устройства DCE бывают трех типов: модемы для работы по выделенным и коммутируемым аналоговым линиям связи, устройства DSU/CSU для работы по выделенным цифровым линиям связи TDM и TA, терминальные адаптеры для работы по выделенным каналам сети ISDN.

Чтобы пользователи сети могли подключаться к сети при помощи оборудования любого производителя, используется стандарт интерфейса «пользователь – сеть» (UNI User –To-Network Interface).

Для стандартизации взаимодействия коммутаторов в глобальной сети используется интерфейс «сеть – сеть» (NNI – Network – To Network Interface).

Типы глобальных сетей

Различают три основных типа глобальных сетей:

1. Сети выделенных каналов;
2. Коммутации каналов;
3. Коммутации пакетов.

Выделенные линии арендуют у компаний, которые владеют каналами дальней связи, или у телефонных компаний в пределах города. Выделенные линии используют двумя способами:

1. При построения территориальной сети определенной технологии выделенные линии служат для связи коммутаторов;
2. Соединение выделенными линиями конечных абонентов без установки промежуточных коммутаторов.

Сети с коммутацией каналов разделяют на аналоговые телефонные сети и цифровые сети с интеграцией услуг ISDN. Сети с коммутацией пакетов могут использовать следующие технологии: X.25, Frame Relay, ATM, SMDS, TCP/IP.

Территориальные сети также разделяют на две категории в зависимости от способа использования при построении корпоративной сети:

1. Магистральные сети;
2. Сети доступа.

Магистральные сети (backbone networks) используют для образования одноранговых связей между крупными локальными сетями. Магистральные сети должны обеспечивать высокую пропускную способность и быть постоянно доступны.

Сети доступа представляют собой сети, которые используются для связи небольших локальных сетей или отдельных компьютеров с центральной сетью предприятия. Программные и аппаратные средства, которые обеспечивают удаленное подключение, называют средствами удаленного доступа. На стороне клиента это, как правило модем и соответствующее ПО, на стороне сервера – сервер удаленного доступа (RAS – Remote Access Service).

Маршрутизация в глобальных сетях

Internet изначально строилась как сеть, объединяющая большое количество существующих систем. С самого начала в ее структуре выделяли магистральную сеть (care backbone network), а сети, присоединенные к магистрали, рассматривались как автономные системы (autonomous systems, AS).

Магистральная сеть и каждая из автономных систем имели свое собственное административное управление и собственные протоколы маршрутизации. Необходимо подчеркнуть, что автономная система и домен имен Internet — это разные понятия, которые служат разным целям. Автономная система объединяет сети, в которых под общим административным руководством одной организации осуществляется маршрутизация, а домен объединяет компьютеры (возможно, принадлежащие разным сетям), в которых под общим административным руководством одной организации осуществляется назначение уникальных символьных имен. Естественно,

области действия автономной системы и домена имен могут в частном случае совпадать, если одна организация выполняет обе указанные функции.

Шлюзы, которые используются для образования сетей и подсетей внутри автономной системы, называются внутренними шлюзами (interiorgateways), а шлюзы, с помощью которых автономные системы присоединяются к магистрали сети, называются внешними шлюзами (exterior gateways). Магистраль сети также является автономной системой. Все автономные системы имеют уникальный 16-разрядный номер, который выделяется организацией, учредившей новую автономную систему, InterNIC.

Соответственно протоколы маршрутизации внутри автономных систем называются протоколами внутренних шлюзов (interior gateway protocol, IGP), а протоколы, определяющие обмен маршрутной информацией между внешними шлюзами и шлюзами магистральной сети — протоколами внешних шлюзов (exterior gateway protocol, EGP). Внутри магистральной сети также допустим любой собственный внутренний протокол IGP.

Внутренние шлюзы могут использовать для внутренней маршрутизации достаточно подробные графы связей между собой, чтобы выбрать наиболее рациональный маршрут. Однако если информация такой степени детализации будет храниться во всех маршрутизаторах сети, то топологические базы данных так разрастутся, что потребуют наличия памяти гигантских размеров, а время принятия решений о маршрутизации станет неприемлемо большим. Поэтому детальная топологическая информация остается внутри автономной системы, а автономную систему как единое целое для остальной части Internet представляют внешние шлюзы, которые сообщают о внутреннем составе автономной системы минимально необходимые сведения — количество IP-сетей, их адреса и внутреннее расстояние до этих сетей от данного внешнего шлюза. (Содержание)

1.6. Стек протоколов TCP/IP

Вопросы для изучения:

- Модель стека TCP/IP;
- Протокол IP;
- Адресация IP;
- Подсети и маски подсети;
- Маршрутизация IP;
- Протокол ICMP;
- Протокол UDP;
- Протокол TCP;

Модель стека TCP/IP

Стек протоколов TCP/IP был разработан по инициативе Министерства Обороны США с целью создания децентрализованной вычислительной сети, устойчивой к отказам отдельных ее сегментов. Протоколы стека были реализованы в ОС Unix университета Беркли, что способствовало широкому распространению протокола. В настоящее время стек TCP/IP используется в сети Internet, а также для передачи данных в локальных сетях и является самым распространенным стеком протоколов. Протоколы TCP/IP позволяют передавать данные через объединенную сеть, которая состоит из множества разнородных подсетей, к которым подключаются разнородные машины.

Стандарты, которые относятся к работе стека описаны в документах **RFC-Request For Comments.**

Модель стека TCP/IP включает четыре уровня:

1. Прикладной уровень;
2. Транспортный(основной) уровень;
3. Межсетевой уровень;
4. Уровень сетевых интерфейсов.

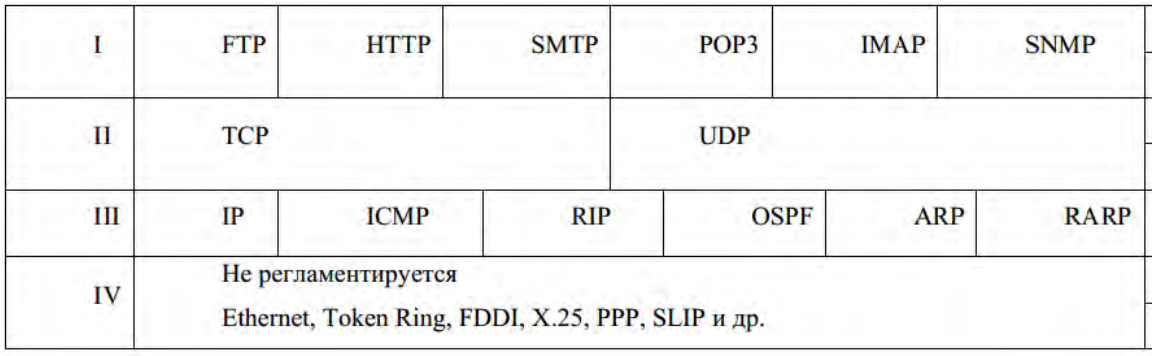

Рисунок 1.4. – Уровни стека TCP/IP.

Протокол IP

Протокол IP (Internet Protocol, RFC 791) является датаграммным протоколом для работы в сетях с коммутацией пакетов. Протокол IP обеспечивает передачу датаграмм от отправителя к получателям, где отправители и получатели являются хост- компьютерами. Каждый хост идентифицируется адресом фиксированной длины.

Протокол Internet обеспечивает при необходимости фрагментацию и сборку датаграмм для передачи данных через сети с малым размером пакетов. Протокол Internet позволяет предоставлять услуги различных типов и качеств.

Две главные функции протокола: **адресация и фрагментация.**

Модули IP используют адреса, помещенные в заголовок IP-пакета, для передачи Internet датаграмм получателям. Выбор пути передачи называется **маршрутизацией**.

Модули IP используют поля в заголовке IP-пакета для фрагментации и восстановления датаграмм, когда это необходимо для их передачи через сети с малым размером пакетов. Каждая датаграмма обрабатывается как независимая единица, которая не имеет связи ни с какими другими датаграммами.

Для формирования услуг IP использует 4 механизма: задание типа сервиса, времени жизни, опций и контрольной суммы заголовка.

Тип обслуживания используется для обозначения требуемой услуги. Тип обслуживания - это набор параметров, который характеризует набор услуг, предоставляемых сетями. Этот способ обозначения услуг должен использоваться шлюзами для выбора рабочих параметров передачи в конкретной сети, для выбора сети, используемой при следующем переходе датаграммы, для выбора следующего шлюза при маршрутизации сетевой Internet датаграммы.

Механизм времени жизни служит для указания предела времени жизни Internet датаграммы. Устанавливается отправителем датаграммы и уменьшается в каждой точке на проходимом датаграммой маршруте. Если параметр времени жизни станет нулевым до того, как датаграмма достигнет получателя, эта датаграмма будет уничтожена.

Механизм опций предоставляет функции управления, которые являются необходимыми или просто полезными при определенных ситуациях. Механизм опций предоставляет такие возможности, как временные штампы, безопасность, специальная маршрутизация.

Контрольная сумма заголовка обеспечивает проверку того, что информация, используемая для обработки датаграмм Internet, передана правильно. Данные могут содержать ошибки. Если контрольная сумма неверна, то Internet датаграмма будет разрушена.

Протокол Internet не обеспечивает надежности коммуникации. Не имеется механизма подтверждений ни между отправителем и получателем, ни между хост- компьютерами. Не имеется контроля ошибок для поля данных, только контрольная сумма для заголовка. Не поддерживается повторная передача, нет управления потоком.

Обнаруженные ошибки могут быть оглашены посредством протокола ICMP (Internet Control Message Protocol), который поддерживается модулем Internet протокола.

Адресация IP

IP-адрес имеет длину 4 байта и обычно записывается в виде четырех чисел, представляющих значения каждого байта в десятичной форме, и разделенных точками, например:

128.10.2.30 - десятичная форма представления адреса,

10000000 00001010 00000010 00011110 - двоичная форма представления адреса.

Адрес состоит из двух логических частей - номера сети и номера узла в сети. Какая часть адреса относится к номеру сети, а какая к номеру узла, определяется значениями первых битов адреса.

Диапазоны номеров сетей, соответствующих каждому классу сетей:

Класс A от 01.0.0.0 до 126.0.0.0
Класс B от 128.0.0.0 до 191.255.0.0
Класс C от 192.0.1.0 до 223.255.255.0
Класс D от 224.0.0.0 до 239.255.255.255
Класс E от 240.0.0.0 до 247.255.255.255

Существует несколько соглашений об особой интерпретации IP-адресов:

Если IP-адрес состоит только из двоичных нулей, 0 0 0 0 0 0……00 00 то он обозначает адрес того узла, который сгенерировал этот пакет.

Если в поле номера сети стоят 0, 0 0 0 0 .....0 Номер узла то по умолчанию считается, что этот узел принадлежит той же самой сети, что и узел, который отправил пакет.

Если все двоичные разряды IP-адреса равны 1, 1 1 1 1 …… 1 1 то пакет с таким адресом назначения должен рассылаться всем узлам, находящимся в той же сети, что и источник этого пакета. Такая рассылка называется ограниченным широковещательным сообщением (limited broadcast).

Если в поле адреса назначения стоят сплошные 1, номер сети 1111.11 то пакет, имеющий такой адрес рассылается всем узлам сети с заданным номером. Такая рассылка называется широковещательным сообщением (broadcast).

Адрес 127.0.0.1 зарезервирован для организации обратной связи при тестировании работы программного обеспечения узла без реальной отправки пакета по сети. Этот адрес имеет название loopback.

Multicast – означает, что данный пакет должен быть доставлен сразу нескольким узлам, которые образуют группу с номером, указанным в поле адреса. Узлы сами идентифицируют себя, то есть определяют, к какой из групп они относятся. Один и тот же узел может входить в несколько групп. Такие сообщения в отличие от широковещательных называются мульти-

вещательными. Групповой адрес не делится на поля номера сети и узла и обрабатывается маршрутизатором особым образом.

Подсети и маски подсети

Использование масок подсети позволяет разделять сеть на несколько подсетей. Маска подсети определяется в документе RFC 950 как 32-битное значение, которое используется для выделения идентификатора сети из IP адреса. Биты маски подсети устанавливаются по следующему правилу:

Все биты которые относятся к идентификатору сети устанавливаются в 1. Все биты которые относятся к идентификатору узла устанавливаются в 0.

Каждому номеру хоста ставится в соответствие маска подсети: либо принятая по умолчанию маска, которая используется для выделения идентификаторов сети на основе классовой адресации, либо маска, которая задается для построения подсетей (суперсетей).

Маска подсети может быть представлена в десятичном формате, разделенном точками, либо в виде десятичного числа которое называют длиной префикса сети (Network prefix length).

Маршрутизация IP

Маршрутизация представляет собой поиск оптимального маршрута при передаче данных от отправителя к получателю Задача маршрутизации разбивается на две подзадачи – маршрутизацию и коммутацию. Коммутация – это процесс продвижения пакетов по сети, который выполняется протоколами IP, IPX. Решение о том, какой пункт будет следующим при перемещении пакета принимается на основании оценок маршрутов, которые описаны в таблицах маршрутизации промежуточных узлов(маршрутизаторов). Таблицы маршрутизации находятся как на конечных узлах – компьютерах, так и на промежуточных – маршрутизаторах. Таблица маршрутизации представляет собой область памяти, данные в которую заносятся изначально при инициализации стека TCP/IP. Данные в таблице затем могут изменяться либо администратором вручную, либо динамически при помощи протоколов, которые называют протоколами маршрутизации. Цель протоколов маршрутизации – это обмен маршрутной информацией между маршрутизаторами.

Таблицы маршрутизации содержат следующую информацию о маршрутах:

- IP – адрес сети или узла;
- маску подсети;
- интерфейс, через который данные передаются на указанный IP – адрес.

- адрес шлюза (следующего хоста), через который данные передаются на указанный IP – адрес;
- метрика (оценка) маршрута – различается в зависимости от того, какой протокол маршрутизации используется. Может содержать количество переходов между промежуточными узлами (число «хопов»), оценку пропускной способности, загруженности, надежности, стоимости канала.

Протоколы маршрутизации можно разделить на 3 группы:

1. Протоколы вектора расстояния: RIP v.1, RIP v.2, IGRP;
2. Протоколы состояния канала: OSPF;
3. Смешанные протоколы: EIGRP.

Протоколы вектора расстояния для обмена информацией используют широковещательную рассылку всего содержимого таблицы маршрутизации. Рассылка выполняется через заданные интервалы времени, независимо от того, произошли ли изменения топологии сети на самом деле. В качестве оценки маршрута используется число «хопов». Диаметр сети ограничен. Используется в небольших сетях.

Протоколы состояния канала выполняют групповую рассылку данных по факту изменения топологии сети («триггерное» обновление). Рассылаются только те данные, которые изменились.

Для оценки маршрута можно использовать множество параметров (пропускная способность, загруженность, надежность канала), по умолчанию используется только пропускная способность канала. Протоколы состояния канала используются в средних и крупных сетях.

Смешанный протокол EIGRP («улучшенный» IGRP) был разработан фирмой CISCO на базе протокола вектора состояний IGRP. Протокол EIGRP поддерживает групповую рассылку, «триггерные» обновления, и способы оценки маршрутов по тем же параметрам, которые используют протоколы состояния каналов.

Протокол ICMP

Протокол ICMP (RFC 792) – протокол контрольных сообщений Internet.

Протокол ICMP используется для передачи сообщения отправителю IP- датаграммы, если ее доставка адресату невозможна.

ICMP использует основные свойства протокола Internet (IP), как если бы ICMP являлся протоколом более высокого уровня. Однако, фактически ICMP является составной частью протокола Internet и должен являться составной частью каждого модуля IP.

Сообщения ICMP должны отправляться в ситуациях, когда датаграмма не может достичь своего адресата, когда шлюз не имеет достаточно места в своем буфере для передачи какой-либо датаграммы, или, когда шлюз

приказывает хост-компьютеру отправлять информацию по более короткому маршруту.

Сообщения ICMP протокола, как правило, оповещают об ошибках, возникающих при обработке датаграмм.

Чтобы проблемы с передачей сообщений не вызывали появление новых сообщений, чтобы это в свою очередь не привело к лавинообразному росту количества сообщений, циркулирующих в сети, что нельзя посылать сообщения об ошибках доставки ICMP сообщений.

Протокол UDP

UDP (User Datagram Protocol — протокол пользовательских датаграмм) — один из ключевых элементов TCP/IP, набора сетевых протоколов для Интернета. С UDP компьютерные приложения могут посылать сообщения (в данном случае называемые датаграммами) другим хостам по IP-сети без необходимости предварительного сообщения для установки специальных каналов передачи или путей данных. Протокол был разработан Дэвидом П. Ридом в 1980 году и официально определён в RFC 768.

UDP использует простую модель передачи, без неявных «рукопожатий» для обеспечения надёжности, упорядочивания или целостности данных. Таким образом, UDP предоставляет ненадёжный сервис, и датаграммы могут прийти не по порядку, дублироваться или вовсе исчезнуть без следа. UDP подразумевает, что проверка ошибок и исправление либо не нужны, либо должны исполняться в приложении. Чувствительные ко времени приложения часто используют UDP, так как предпочтительнее сбросить пакеты, чем ждать задержавшиеся пакеты, что может оказаться невозможным в системах реального времени.

UDP обеспечивает многоканальную передачу (с помощью номеров портов) и проверку целостности (с помощью контрольных сумм) заголовка и существенных данных.

Протокол TCP

Transmission Control Protocol, RFC 793, протокол TCP обеспечивает сервис надежной доставки данных между двумя процессами при помощи двусторонних потоков данных. На вход модуля протокола TCP со стороны приложения подается поток октетов (байт). Поток байт разбивается на сегменты переменной длины. Каждому октету присваивается порядковый номер. Сегмент содержит информацию о порядковом номере своего первого октета.

Чтобы на одном компьютере много процессов могли пользоваться услугами протокола TCP, протокол TCP предоставляет набор адресов или портов. Вместе с адресами сетей и хост-компьютеров они образуют сокет

(socket). Каждое соединение уникальным образом идентифицируется парой сокетов. Любой сокет может одновременно использоваться во многих соединениях.

При установлении и разрыве соединения используется последовательность обмена сообщениями, которая называется трехкратным квитированием (от слова квитанция).

Для обеспечения надежной доставки данных используется механизм подтверждений, который заключается в следующем: при отправке сегмента данных его копия помещается в очередь неподтвержденных сегментов. Если по истечении определенного интервала времени подтверждение приема не получено, то производится повторная передача этого сегмента.

Чтобы не подтверждать каждый отправленный сегмент и управлять потоком данных, используется механизм скользящего окна. Размер окна определяет количество сегментов, которые могут быть переданы без подтверждения. Размер окна устанавливается станцией – получателем, и, если ее буфер переполнен, то размер окна может быть уменьшен до нуля.

Если у отправителя есть данные с высоким приоритетом, они могут быть помечены как «срочные», и должны быть приняты станцией – получателем даже если ее буфер заполнен данными (буфер очищается и записываются «срочные» данные). (Содержание)

1.7. Службы WINS, DNS, DHCP

Вопросы для изучения:

- Служба DNS;
- Служба WINS;
- Служба DHCP.

Службы DNS и DHCP являются ключевыми сетевыми службами в любой корпоративной сети, построенной на базе стека протоколов TCP/IP. Более того, в среде Windows Server наличие службы DNS является одним из обязательных условий развертывания службы каталога Active Directory. Служба DNS осуществляет разрешение символических доменных имен в соответствующие им IP-адреса. Удобным дополнением к службе DNS в среде Windows Server является служба DHCP, упрощающая процесс конфигурации сетевых хостов (в том числе выделение хосту IP-адреса). Кроме того, в среде Windows многими администраторами традиционно используется служба WINS, осуществляющая разрешение символических NetBIOS-имен в соответствующие IP-адреса. Хотя роль этой службы в Windows Server была значительно уменьшена за счет реализации механизма динамической регистрации доменных имен, служба WINS может по-прежнему использоваться для организации процесса разрешения имен (например, в процессе перехода с Windows NT на Windows Server).

В рамках Windows Server указанные службы могут работать в тесном взаимодействии, обеспечивая простой и эффективный способ конфигурации хостов, а также разрешения символических имен в IP- адреса.

Служба DNS

Служба доменных имен, Domain Name System, DNS, является одним из важнейших компонентов сетевой инфраструктуры Windows Server. Служба доменных имен осуществляет разрешение, или преобразование, символьных имен в IP-адреса. Клиенты доменов на базе Active Directory используют службу DNS для обнаружения контроллеров домена.

Доменная структура каталога отображается на пространство имен DNS. Поэтому процесс проектирования доменной структуры каталога должен происходить одновременно с формированием пространства имен DNS. Ошибки, допущенные при проектировании пространства имен DNS, могут стать причиной недостаточной производительности сети и, возможно, даже привести к ее отказу.

Возможности DNS-клиентов.В составе Windows Server имеется служба DNS-клиента. DNS-клиент осуществляет взаимодействие с DNS-сервером с целью разрешения доменных имен в IP- адреса. При этом реализация DNS-клиента в Windows Server характеризуется следующими возможностями:

Клиентское кэширование. Ресурсные записи (RR), полученные как ответы на запросы, добавляются в клиентский кэш. Эта информация хранится в течение заданного времени и может использоваться для ответа на последующие запросы;

Кэширование отрицательных ответов. В дополнение к кэшированию положительных ответов на запросы от серверов DNS, служба DNS также кэширует отрицательные ответы на запросы. Отрицательный ответ приходит, если ресурсная запись с запрошенным именем не существует. Кэширование отрицательных ответов предотвращает повторные запросы для несуществующих имен, снижающие производительность клиентской службы;

Блокировка неотвечающих серверов DNS. Клиентская служба DNS использует список поиска серверов, упорядоченных по предпочтению. Этот список включает все серверы DNS, настроенные для каждого из активных сетевых подключений в системе. Система способна перестраивать этот список, основываясь на следующих критериях: предпочтительные серверы DNS имеют высший приоритет, а остальные серверы DNS чередуются. Неотвечающие серверы временно удаляются из списка.

Структура DNS. Для правильного формирования пространства имен DNS администратор должен ясно понимать структуру службы DNS, ее основные компоненты и механизмы. Для начала определимся с используе-

мой терминологией. Службой DNS называется служба, выполняющая преобразование символических доменных имен в IP-адреса в ответ на запросы клиентов. Компьютер, на котором функционирует экземпляр службы DNS, называется DNS-сервером. Компьютер, обращающийся к DNS-серверу с запросом на разрешение имени, называется DNS-клиентом. Клиент DNS функционирует на уровне прикладного программного интерфейса (API), осуществляя разрешение доменных имен прозрачно для пользователей и приложений. Основная задача DNS-клиента заключается в передаче запроса на разрешение доменного имени DNS-серверу. В ответ на свой запрос клиент должен получить либо IP-адрес, либо сообщение о невозможности разрешить предоставленное серверу доменное имя. Клиент DNS передает полученный IP-адрес приложению, инициировавшему процесс разрешения имени.

Пространство имен DNS. Основным компонентом пространства имен DNS являются домены (domain). Домен рассматривается как группа сетевых хостов, объединенных по некоторому логическому признаку. Домены соединяются друг с другом при помощи отношений “родитель-потомок”, образуя тем самым некоторую иерархию. Положение домена в этой иерархии определяет уровень домена.

В основании иерархического пространства имен DNS лежит домен, который называется корневым доменом (root domain). Корневой домен является формальным элементом, символизирующим иерархичность пространства доменных имен, выступая в качестве родительского контейнера для всех доменов первого уровня.

Если домены выступают в роли контейнеров или узлов рассматриваемой иерархической структуры, то в качестве листьев выступают сведения о ресурсах этих доменов. Служба DNS определена в рамках стека протоколов TCP/IP, в котором для обозначения любого объекта сети используется понятие хост (host).

Любой объект пространства имен DNS, будь это домен или хост, имеет имя, уникальное в пределах родительского контейнера. Это имя может быть образовано из символов латинского алфавита, цифр и знака тире (“—”). Некоторые версии DNS (включая реализацию DNS в Windows Server) допускают использование в именах объектов символа подчеркивания (“_”), а также символов в формате UTF-8.

Схемы запросов. Рассмотрим процесс разрешения доменных имен в IP-адреса, определенный в рамках спецификации службы DNS (RFC 1034 и RFC 1035). Процесс разрешения доменного имени предполагает строго регламентированное взаимодействие DNS-клиента и цепочки DNS-серверов. Взаимодействие начинается с момента, когда пользователь или приложение используют доменное имя для ссылки на некоторый хост. Соединение с любым хостом осуществляется только на уровне IP-адресов. Поэтому DNS-клиент должен выполнить разрешение доменного имени.

Всю работу по разрешению доменного имени выполняет DNS-сервер. В зависимости от обстоятельств, в процесс разрешения имени может быть вовлечен один сервер или несколько DNS-серверов.

Служба WINS

Служба WINS (Windows Internet Name Service) обеспечивает поддержку распределенной базы данных для динамической регистрации и разрешения NetBIOS- имен. Служба WINS отображает пространство имен NetBIOS и адресное пространство IP друг на друга и предназначена для разрешения NetBIOS-имен в маршрутизируемых сетях, использующих NetBIOS поверх TCP/IP. Следует напомнить, что NetBIOS-имена используются ранними версиями операционных систем Windows как основной способ именования сетевых ресурсов. Служба WINS была разработана с целью упрощения процесса управления пространством имен NetBIOS в сетях на базе TCP/IP. Основное назначение службы WINS заключается в разрешении NetBIOS-имен в IP-адреса. Процесс разрешения строится на основе базы данных WINS-сервера, содержащей отображения пространства NetBIOS-имен на пространство IP-адресов. Входя в сеть, клиент регистрирует свое имя в базе данных WINS-сервера. При завершении работы клиент отправляет сообщение WINS-серверу, извещая его об освобождении им зарегистрированного имени.

Посредник WINS. В спецификации службы WINS описываются три участника: WINS-сервер, WINS- клиенты, а также посредники WINS (WINS proxy). WINS-сервер обрабатывает запросы на регистрацию имен от WINS-клиентов, регистрирует их имена и соответствующие им IP- адреса, а также отвечает на запросы разрешения имен от клиентов, возвращая IP-адрес по имени, при условии, что это имя находится в базе данных сервера. В сети может быть установлено несколько WINS-серверов. Базы данных всех существующих WINS-серверов синхронизируются в результате процесса репликации. Под посредником WINS понимается специальный WINS-клиент, который может обращаться к WINS-серверу от имени других хостов, не способных обратиться к WINS- серверу самостоятельно. WINS-посредники используются для поддержки хостов, осуществляющих разрешение NetBIOS – имен методом широковещательных рассылок. Аналогичным методом (путем рассылки широковещательных сообщений) данный тип хостов информирует другие сетевые хосты о занимаемом им NetBIOS-имени. При этом принято говорить, что данный хост работает в режиме b-узла. Поскольку широковещательные сообщения не ретранслируются маршрутизаторами, для нормальной работы сети возникает необходимость устанавливать WINS-серверы в каждой подсети (либо отказаться от клиентов, работающих в режиме b-узла). В качестве альтернативы

администратор может сконфигурировать один из WINS-клиентов в качестве WINS- посредника.

Хост, функционирующий в режиме b-узла, не подозревает о существовании WINS- посредника. Этот хост рассылает широковещательные запросы, которые принимаются всеми хостами подсети, в том числе и WINS-посредником. WINS-посредник переадресует эти сообщения WINS-серверу, информируя последний о регистрации или освобождении соответствующего имени. Аналогичным образом хост, функционирующий в режиме b- узла, обращается с широковещательным запросом на разрешение имени. Посредник WINS проверяет собственный локальный кэш имен, и, если в нем не обнаружено запрашиваемое имя, переадресует запрос WINS-серверу.

Репликация WINS. Если в сети используется несколько WINS-серверов, для поддержания их баз данных в синхронизированном состоянии администратор может настроить между ними репликацию. В показанном на рисунке примере репликация осуществляется в обе стороны. То есть содержимое базы данных одного WINS-сервера реплицируется на другой WINS-сервер и наоборот. Однако возможны и другие варианты: как однонаправленная репликация, так и сложные топологии репликации (например, образующие кольцо). После настройки механизма репликации между WINS-серверами каждый из них будет располагать сведениями обо всех именах NetBIOS, зарегистрированных в корпоративной сети. Благодаря этому любой клиент будет иметь возможность разрешать NetBIOS-имена независимо от того, на каком из WINS-серверов эти имена были зарегистрированы. Участвующие в репликации WINS-серверы называются партнерами по репликации. В зависимости от того, как именно WINS-сервер инициирует процедуру репликации, он может выступать в одном из трех качеств:

Передающий партнер (Push Partner). В этом сценарии WINS-сервер инициирует процесс репликации самостоятельно, извещая своих партнеров об изменении своей базы данных путем отправки специального сообщения;

Принимающий партнер (Pull Partner). В этом случае WINS-сервер запрашивает репликацию изменений у своего партнера по репликации с определенной периодичностью.

Передающий/принимающий партнер (Push/Pull Partner). Этот сценарий предполагает использование обоих вышеописанных методов инициации процесса репликации.

Механизм постоянных соединений. Инициируя процесс репликации, WINS-сервер устанавливает соединение с другим сервером, в рамках которого осуществляется передача соответствующих изменений. Установка соединения требует затрат определенных системных ресурсов. При этом

интенсивные изменения могут снизить общую производительность WINS-сервера.

В Windows Server реализация службы WINS поддерживает механизм постоянного соединения с партнером по репликации. Данный механизм позволяет устанавливать соединение только один раз, после чего оно сохраняется активным.

Любое изменение, осуществляемое в базе данных сервера, будет немедленно реплицировано на другие серверы, с которыми установлено постоянное соединение. Благодаря этому базы данных всех WINS-серверов будут всегда находиться в актуальном состоянии.

Служба DHCP

DHCP (англ. 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. Данный протокол работает по модели «клиент-сервер». Для автоматической конфигурации компьютер-клиент на этапе конфигурации сетевого устройства обращается к так называемому серверу DHCP и получает от него нужные параметры. Сетевой администратор может задать диапазон адресов, распределяемых сервером среди компьютеров. Это позволяет избежать ручной настройки компьютеров сети и уменьшает количество ошибок. Протокол DHCP используется в большинстве сетей TCP/IP.

DHCP является расширением протокола BOOTP, использовавшегося ранее для обеспечения бездисковых рабочих станций IP-адресами при их загрузке. DHCP сохраняет обратную совместимость с BOOTP.

Протокол DHCP предоставляет три способа распределения IP-адресов:

Ручное распределение. При этом способе сетевой администратор сопоставляет аппаратному адресу (для Ethernet сетей это MAC-адрес) каждого клиентского компьютера определённый IP-адрес. Фактически, данный способ распределения адресов отличается от ручной настройки каждого компьютера лишь тем, что сведения об адресах хранятся централизованно (на сервере DHCP), и потому их проще изменять при необходимости.

Автоматическое распределение. При данном способе каждому компьютеру на постоянное использование выделяется произвольный свободный IP-адрес из определённого администратором диапазона.

Динамическое распределение. Этот способ аналогичен автоматическому распределению, за исключением того, что адрес выдаётся компьютеру не на постоянное пользование, а на определённый срок. Это называется арендой адреса. По истечении срока аренды IP-адрес вновь считается свободным, и клиент обязан запросить новый (он, впрочем, может оказать-

ся тем же самым). Кроме того, клиент сам может отказаться от полученного адреса.

Некоторые реализации службы DHCP способны автоматически обновлять записи DNS, соответствующие клиентским компьютерам, при выделении им новых адресов. Это производится при помощи протокола обновления DNS, описанного в RFC 2136. Помимо IP-адреса, DHCP также может сообщать клиенту дополнительные параметры, необходимые для нормальной работы в сети. Эти параметры называются опциями DHCP. Список стандартных опций можно найти в RFC 2132.

Некоторыми из наиболее часто используемых опций являются:

• IP-адрес маршрутизатора по умолчанию;
• маска подсети;
• адреса серверов DNS;
• имя домена DNS.

Все сообщения протокола DHCP разбиваются на поля, каждое из которых содержит определённую информацию. Все поля, кроме последнего (поля опций DHCP), имеют фиксированную длину. (Содержание)

1.8. Организация домена. Active Directory. Служба браузеров

Вопросы для изучения:

- Организация домена;
- Active Directory;
- Служба браузеров.

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 дове-

рия. 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.

Служба браузеров

Для начала разберём несколько понятий и терминов. Как известно, список компьютеров в сети можно посмотреть, заглянув в сетевое окружение (My Network Places). Если всё настроено как надо, то мы видим список компьютеров и можем заходить на любые из них и просматривать расшаренные на них папки и принтеры. Как же организовывается такая схема?Базовые настройки и понятия, необходимые для работы сетевого окружения

Возьмём некоторую локальную сеть, в которой есть один домен широковещания, т.е. один компьютер или узел может найти другого по широковещательному запросу или как его ещё называют – броадкасту

(broadcast). Широковещательные запросы свободно проходят через хабы и свитчи и ограничиваются лишь маршрутизаторами, которые не пропускают широковещательные пакеты в другие сети. Если все узлы в сети подключены к свитчу (или свитчам, между которыми на пути нету маршрутизаторов и соединены прямым кабелем), то каждый узел может общаться широковещательными пакетами с любым другим узлом в данной сети. Чаще всего все эти компьютеры будут входить в одну рабочую группу, по умолчанию именуемую Workgroup. Вкратце ситуация происходит следующим образом: при запуске компьютеров в сети начинают происходить выборы главного компьютера, который будет отвечать за списки компьютеров в сетевом окружении и которого называют главным обозревателем или мастер-браузером (master browse server). В выборах участвуют только компьютеры с запущенной службой Computer Browser. После выбора мастер-браузера выбираются ноль или больше резервных обозревателей или резервных браузеров (backup browse server), которые будут обслуживать клиентов. Мастер-браузер (здесь и далее термин “браузер” будет использоваться для обозначения главного обозревателя компьютеров или резервного обозревателя). После прохождения всех выборов каждый узел с запущенной службой Server объявляет себя мастер-браузеру, чтобы тот включил его в общий список компьютеров. Когда все узлы объявят себя мастер-браузеру, то тот в свою очередь сформирует список для сетевого окружения.

Регулярно, через некоторый интервал времени (от 1 до 12 минут) компьютер обращается к мастер-браузеру за списком резервных обозревателей. Получив его, компьютер произвольно выбирает одного из резервных браузеров и запрашивает уже у него список компьютеров в сети. Если же резервных браузеров нету, то сам мастер-браузер будет обслуживать клиента и передавать ему списки компьютеров. Именно этот список можно видеть в папке My Networks Places. Для корректной работы обозревателя требуется наличие компьютера под управлением Microsoft Windows 9x/3.x for Workgroups (данная статья пишется под среду Windows 2000/XP/2003/Vista) и выше со включенным клиентом Client for Microsoft Networks и включенном транспортном протоколе NetBIOS и запущенной службой Server и Workstation.

(Содержание)

1.9. Совместное использование ОС Windows и Linux в сети

Вопросы для изучения:

- ОС Windows;
- ОС Unix;
- ОС Linux;
- Совместное использование ОС Windows и Linux в сети.

ОС Windows

Существует несколько альтернативных возможностей использования ОС Windows, разработанной корпорацией Microsoft. В основном они ориентированы на облегчение задач сетевого администрирования и установки Windows для пользователей, желающих обращаться к сети через ОС Windows. Фирма Microsoft разработала ОС Windows, ориентированную на многопользовательскую работу. Чтобы подчеркнуть её принципиальную новизну, в название добавили символы NT (New Technology – новая технология). Её промышленный выпуск начался в 1993 году. Это была 32-разрядная ОС со встроенной сетевой поддержкой и развитыми многопользовательскими средствами. Windows NT обеспечивает: многозадачность, многопроцессорную работу, переносимость на различные платформы, защиту от несанкционированного доступа, заданный уровень секретности. Описываемые процедуры предусматривают копирование всех файлов Windows в совместно используемый каталог сети. Затем пользователи или супервизоры могут установить Windows на рабочих станциях, обращаясь к программам установки и файлам этого совместно используемого каталога, а не инсталлируя Windows непосредственна на рабочих станциях. Известно несколько методов работы с ОС Windows.

Метод 1. Пользователи полностью устанавливают Windows на своих рабочих станциях, обращаясь к программам установки и файлам в совместно используемом каталоге Windows на сервере. После этого Windows запускается с локального жёсткого диска.

Метод 2. На рабочую станцию копируются только персональные файлы конкретного пользователя. Другие файлы остаются в сети в каталоге, где они используются совместно с другими пользователями. Windows загружается из сети, но считывает и записывает файлы конфигурации конкретного пользователя.

Метод 3. Пользователи полностью запускают Windows из сети. Их личные файлы конфигурации хранятся в персональных каталогах, а совместно используемые файлы – в разделяемом каталоге, с которым могут работать все другие пользователи Windows.

Первый метод иногда считают наилучшим. При этом на жёстком диске рабочей станции должно быть достаточно места для размещения всех файлов Windows. Другой недостаток этого метода заключается то, что пользователь сам отвечает за обновление файлов ОС и приложений в его системе. Когда Windows устанавливается на сервере, эти задачи могут выполнять администраторы сети.

Второй метод позволяет сэкономить пространство на диске рабочей станции, так как там записывается всего несколько файлов ОС, таких как файлы INI. Однако при доступе пользователей к совместно используемым

файлам ОС увеличивается сетевой трафик. Обновления выполняются на сервере, что облегчает задачи управления.

Третий метод самый простой с точки зрения администратора сети, но он создаёт наиболее интенсивный трафик, а потому обычно используется для запуска Windows с бездисковых рабочих станций.

Версия ОС Windows NT 4.0 выпускалась до 2000 года. Ей на смену, вышла версия 5.0 под названием Windows 2000, в основе которой заложена технология NT. Windows 2000, имеет четыре модификации:

- Professional для рабочих станций (поддерживает двухпроцессорную ПЭВМ);
- Server для серверов малых локальных сетей (для четырёхпроцессорной ПЭВМ);
- Advanced Server для серверов больших локальных и удалённых сетей (до 16 процессоров);
- Data Center Server для крупных узлов сетей (поддерживает ЭВМ на 64 процессорах).

Затем в 2001 г. появляется настольная версия Windows XP, а в 2003 г. – серверная ОС – Windows Server 2003. В системах семейства этой серверной ОС немного принципиально новых решений, и они представляют эволюционное развитие серверных продуктов Windows 2000. Это более законченные и надёжные реализации революционных, по сравнению с Windows NT 4.0, изменений, появившихся в Windows 2000. При этом семейство Windows Server 2003 унаследовало ряд возможностей системы Windows XP, отсутствовавших в Windows 2000. Четыре редакции ОС, образуют семейство Windows Server 2003 (Standard, Enterprise, Datacenter и Web Edition), которые в первую очередь различаются по степени масштабируемости и производительности. Windows Server 2003, Standard Edition – универсальная сетевая система общего назначения, предназначенная для корпоративного использования небольшим компаниям или подразделениям крупных фирм при решении различных задач: поддержка служб печати и файловых сервисов, маршрутизация и удалённый доступ, обеспечение работы СУБД и т. д.

Windows Server 2003, Enterprise Edition – платформа для развертывания бизнес-задач любого масштаба, включая службы Интернета. При этом обеспечивается бóльшая производительность и отказоустойчивость, чем при использовании Windows Server 2003, Standard Edition, достигаемые за счёт большего числа поддерживаемых процессоров, кластеризации и увеличенного объёма памяти.

Windows Server 2003, Datacenter Edition самая мощная из всех редакций Windows Server 2003. Она ориентирована на обеспечение максимального уровня производительности и надёжности для критически важных приложений и задач. В ней отсутствует ряд служб, целесообразных для использования в небольших компаниях или группах.

Windows Server 2003, Web Edition – новый продукт в семействе серверов Microsoft, в первую очередь предназначенный для веб-хостинга и поддержки XML веб-служб в небольших организациях и подразделениях.

ОС UNIX

Первоначально в середине 1970-х годов эта ОС создавалась как интерактивная многозадачная система для терминальной работы миникомпьютеров и мэйнфреймов. С тех пор она выросла в одну из наиболее распространённых ОС, несмотря на свой неудобный интерфейс и отсутствие централизованной стандартизации. До 1980 года UNIX использовалась в университетах и правительственных исследовательских центрах. Основанная на наборе простых, но мощных инструментальных средств, эта ОС стала использоваться для разработки программных средств и получила промышленное применение. Первая коммерческая версия системы под названием Xenix выпущена в середине 1970-х годов фирмой Microsoft. Широкому её распространению способствовала бесплатная поставка в форме исходных текстов.

Существенная особенность UNIX – переносимость на различные ЭВМ, так как её сетевая файловая система, лучше других ОС приспособлена для работы в сетях разнообразных компьютеров. Семейство ОС UNIX в основном ориентировано на большие локальные и глобальные сети ЭВМ. ОС UNIX одновременно является операционной средой использования существующих прикладных программ и средой разработки новых приложений. Стандартным языком программирования в данной среде является язык Си (Си++). Это объясняется тем, что, во-первых, ОС UNIX написана на языке Си, а, во-вторых, язык Си является одним из наиболее качественно стандартизованных языков.

В ОС UNIX, как и в любой другой многопользовательской ОС, обеспечивающей защиту пользователей друг от друга и защиту системных данных от любого непривилегированного пользователя, имеется защищённое ядро, управляющее ресурсами компьютера и предоставляющее пользователям базовый набор услуг. Это не очень чётко структурированный монолит большого размера, поэтому программирование на уровне ядра ОС UNIX продолжает оставаться искусством.

Система обладает свойством высокой мобильности – вся ОС, включая её ядро, сравнительно просто переносится на различные аппаратные платформы. Все части системы, не считая ядра, являются полностью машинно-независимыми. Эти компоненты аккуратно написаны на языке Си, и их перенос на новую платформу обычно требует только перекомпиляция исходных текстов в коды целевого компьютера. Небольшая часть ядра машинно-зависимая. Она написана на смеси языков Си и Ассемблера целевого процессора. При переносе системы на новую платформу требуется переписать

эту часть ядра с использованием языка Ассемблера и с учётом специфических черт целевой аппаратуры. Машинно-зависимые части ядра изолированы от основной машинно-независимой части. При хорошем понимании назначения каждого машинно-зависимого компонента переписывание машинно-зависимой части в основном является технической задачей, хотя и требует программистов высокой квалификации.

Средства общения с ядром в ОС UNIX называются системными вызовами. Для обращения к функциям ядра ОС используют “специальные команды” процессора, при выполнении которых возникает особое внутреннее прерывание процессора, переводящее его в режим ядра. В большинстве современных ОС этот вид прерываний называется “trap” – ловушка.

При обработке таких прерываний (дешифрации) ядро ОС распознаёт, что данное прерывание является запросом к ядру со стороны пользовательской программы на выполнение определённых действий, выбирает параметры обращения и обрабатывает его, после чего выполняет “возврат из прерывания”, возобновляя нормальное выполнение пользовательской программы.

Поскольку ОС UNIX стремится обеспечить среду, в которой пользовательские программы полностью мобильны, потребовался дополнительный уровень, скрывающий особенности конкретного механизма возбуждения внутренних прерываний. Он обеспечивается “библиотекой системных вызовов” – обычной библиотекой с заранее реализованными функциями системы программирования языка Си.

Внутри любой функции конкретной библиотеки системных вызовов содержится код, являющийся специфичным для данной аппаратной платформы. Каждому возможному прерыванию процессора соответствует фиксированный адрес физической оперативной памяти.

Когда процессору разрешается прерваться из-за наличия внутренней или внешней заявки на прерывание, происходит аппаратная передача управления на ячейку физической оперативной памяти с соответствующим адресом. Обычно адрес этой ячейки называется “вектором прерывания”.

ОС Linux

Linux – свободно распространяемая версия UNIX, разработана аспирантом Линусом Торвальдсом (Linus Torvalds) в Университете Хельсинки (Финляндия) и впервые появилась в октябре 1991 года. Затем, во время сетевой конференции в 1992 году, он объявил, что в качестве “хобби” приступил к разработке UNIX-подобной компактной ОС для процессора I80386. В рамках UNIX–систем была разработана ОС для ПЭВМ под названием Linux.

Основное внимание в этой ОС уделялось созданию ядра. Вопросы поддержки работы с пользователем, документирования, тиражирования и

т.п. обсуждались. Её особенность – открытый код. ОС поставляется в виде исходного текста, который можно модифицировать под конкретный состав и направление использования ЭВМ. Linux распространяется бесплатно и считается самой быстроразвивающейся ОС в области многопользовательских многозадачных систем. Это гибкая полноценная многозадачная многопользовательская ОС семейства UNIX-подобных ОС, способна работать с X Windows, TCP/IP, Emacs (редактор текста), UUCP, mail и USENET. При этом множество пользователей может одновременно работать на одной машине и выполнять много программ.

X Windows (Система X Window или кратко просто X) – стандартный графический интерфейс для UNIX-машин, благодаря которому пользователь может одновременно видеть на экране компьютера несколько окон, при этом каждое окно имеет независимый login.

UUCP (UNIX-to-UNIX Copy) – старейший механизм передачи файлов, электронной почты и электронных новостей между UNIX-машинами. Классически UUCP-машины связываются друг с другом по телефонным линиям через модем, но UUCP может использовать в качестве транспортного средства и связь по TCP/IP. Практически все важнейшие современные программные пакеты используются под Linux. Это уникальная операционная система. Чтобы эффективно её использовать, важно понимать её философию и особенности проектирования. Это большая и достаточно сложная система для решения сложных задач и организации распределённых вычислений – отличный выбор для персональных вычислений в среде UNIX.

Linux проста в инсталляции и использовании. Она обеспечивает полный набор протоколов TCP/IP для сетевой работы и услуг TCP/IP (FTP, telnet, NNTP и SMTP). Ядро Linux поддерживает загрузку только нужных страниц, то есть с диска в память загружаются те сегменты программы, которые действительно используются. При этом возможно использование одной страницы, физически один раз загруженной в память, несколькими выполняемыми программами. Для увеличения объёма доступной памяти Linux осуществляет разбиение диска на страницы: то есть на диске может быть выделено до 256 Мбайт “пространства для свопинга” (swap space – место обмена). Когда системе необходимо использовать больше физической памяти, она с помощью свопинга выводит неактивные страницы на диск, что позволяет выполнять более объёмные программы и обслуживать одновременно больше число пользователей. Свопинг не исключает наращивания физической памяти, поскольку он снижает быстродействие и увеличивает время доступа.

Выполняемые программы используют динамически связываемые библиотеки, т.е. они могут совместно использовать библиотечную программу, представленную одним физическим файлом на диске. Это позволяет выполняемым файлам занимать меньше места на диске, особенно тем, которые многократно используют библиотечные функции. Есть также ста-

тические связываемые библиотеки для тех, кто желает пользоваться отладкой на уровне объектных кодов или иметь “полные” выполняемые программы, которые не нуждаются в разделяемых библиотеках. В Linux разделяемые библиотеки динамически связываются во время выполнения, позволяя программисту заменять библиотечные модули своими собственными.

Linux идеален для создания UNIX-приложений. Он обеспечивает полную UNIX-среду программирования, включая все стандартные библиотеки, программный инструментарий, компиляторы, отладчики, которые встречаются и в других UNIX-системах.

Профессиональные UNIX-программисты и системные администраторы могут использовать Linux на домашних компьютерах, а с них переносить написанные программы на компьютеры организации (фирмы). Такой метод позволяет экономить время и деньги, обеспечивает комфортабельную работу на домашнем компьютере. Linux прежде всего ориентирован на разработчиков. Однако любой человек, имеющий достаточные знания и навыки, может принять участие в совершенствовании и отладке ядра, переносе в Linux новых программ, написании документации, помощи новичкам.

Совместное использование

В последнее время популярность Linux растет буквально каждый день. Linux является высокопроизводительной некоммерческой операционной системой, одной из разновидностей Unix.

Основными преимуществами Linux являются открытость и мультипоточность, кроме того, в ней есть возможности четкого разграничения ресурсов и уровней доступа пользователей.

На сегодняшний день многие производители программного обеспечения поддерживают эту операционную систему.

Samba — набор программ, которые предназначены для организации доступа клиентов к файловому пространству сервера и принтерам с помощью протоколов SMB (Server Message Block) и CIFS (Common Internet Filesystem).

Первоначально написанный для Unix Samba теперь также работает под управлением и других ОС, в частности OS/2 и VMS. Это означает, что такие средства этих операционных систем, как файл-сервер и сервер печати, могут быть использованы для SMB- и CIFS-клиентов.

В настоящее время существуют соответствующие клиенты для DOS, Windows NT, Windows 95, Linux smbfs, OS/2, Pathworks.

Протокол SMB используется Microsoft Windows NT и 95 для организации доступа к дискам и принтерам.

При помощи SAMBA возможно:

- предоставлять доступ к файловой системе под ОС Linux для Windows-машин;
- получать доступ к файловой системе под ОС Windows для Linux-машин;
- предоставлять доступ к принтерам под ОС Linux для Windows-машин;
- получать доступ к принтерам под ОС Windows для Linux-машин.

Компоненты пакета Samba выполняют следующие функции:

Демон smbd предоставляет службы доступа к файлам и принтерам для клиентов протокола SMB, таких как Windows 95/98, Windows for Workgroups, Windows NT или LanManager. Конфигурация для этого демона задается в файле smb.cfg.

Демон nmbd обеспечивает поддержку сервера имен Netbios для клиентов. Он может запускаться в интерактивном режиме для опроса других демонов службы имен.

Программа smbclient является простым SMB-клиентом для UNIX-машин. Она используется для доступа к ресурсам на других SMB-совместимых серверах (таких как Windows NT), а также позволяет UNIX-станции воспользоваться удаленным принтером, подключенным к любому SMB-серверу (например, к компьютеру с WfWg).

Утилита testparm предназначена для проверки файла конфигурации smb.conf.

Утилита smbstatus позволяет выяснить, кто в данный момент использует сервер smbd. Утилита nmblookup дает возможность запрашивать имена NetBios из UNIX-машин.

При помощи **утилиты make smbcodepages** создаются файлы для описания SMB кодовой страницы.

Утилита smbpasswd дает возможность шифровать пароли.

Каждый компонент детально описан на страницах руководства, поставляемого с пакетом Samba. (Содержание)

1.10. Безопасность сети

Вопросы для изучения:

- XSS Filter;
- SmartScreen Filter;
- Data Execution Prevention;
- HTTPS.

XSS Filter

Фильтры — фрагменты кода, которые могут быть выполнены до и\или после выполнения действия контроллера. Фильтры, при необходимости, могут не допустить выполнения запрошенного действия.

- Защита от Cross-Site Scripting – наиболее распространенный тип атак;
- Эвристические алгоритмы;
- Автоматическая блокировка;

XSS-фильтр работает как компонент IE8, который просматривает все запросы и ответы, проходящие через браузер. Когда фильтр обнаруживает XSS в межсайтовом запросе, он обнаруживает и нейтрализует атаку, если она зависит от ответа сервера. Пользователям не задают вопросы, на которые они не могут ответить – IE просто блокирует вредоносный скрипт от исполнения.

Как можно понять, есть множество интересных и тонких сценариев, которые фильтр должен обрабатывать правильно. Вот некоторые из них:

Фильтр должен быть эффективен, даже если атака направлена на артефакт часто используемых рабочих сред веб-приложений. Например, будет ли атака замечена, если определенный символ в запросе был потерян или модифицирован при повторном запросе;

При фильтрации наш код не должен предоставить новый сценарий для атаки, которая бы отличалась от существующей. Например, представьте, что фильтр можно заставить нейтрализовать закрывающий тэ SCRIPT. В таком случае не доверенный контент с сайта позже может быть запущен как скрипт.

И, конечно, в дополнение ко всему этому нам нужно эффективно бороться со всеми векторами XSS-атак, которые еще не были закрыты другими способами сокращения поверхности для XSS-атаки.

Пользователю не задаются вопросы, на которые он не может ответить – IE просто блокирует вредоносный скрипт от исполнения.

Проще говоря, XSS-фильтр в IE8 призван обеспечить глубокую защиту путем автоматического обнаружения и предотвращения наиболее распространенных XSS-атак, с которыми пользователи сталкиваются на просторах Интернета, без потери производительности или совместимости.

Повсюду, где реализован пользовательский ввод, существует опасность XSS, который по сути является разновидностью Injection-атак (внедрение опасного кода). Если ввод не фильтруется, достаточно оставить сценарий скрипта на странице, и этот скрипт автоматически будет выполняться на браузере любого пользователя, просматривающего данную страницу.

XSS Filter – он анализирует все request и response, выявляет XSS, после чего уведомляет пользователя о грозившей, но предотвращенной угрозе. Компании, которые беспокоятся о том, чтобы их сайт не был «случай-

но» заблокирован этим фильтром, могут решить свою проблему – XSS-фильтр можно отключить передаваемым через http заголовок параметром X-XSS-Protection: 0.

SmartScreen Filter

В Internet Explorer 8 присутствует фильтр SmartScreen Anti-Malware. Благодаря расширенной эвристике и телеметрии он снижает вероятность перехода пользователя по фальшивой ссылке. После ввода URL проводится подробный анализ всей адресной строки, и результаты сравниваются с базой данных сайтов, на которых имеются вредоносные программы или которые являются фишинговыми ресурсами.

В результате работы SmartScreen Anti-Malware при попытке запуска опасного кода, загруженного с сайта, его исполнение блокируется, а пользователю выводится предупреждающее сообщение. Если сайт находится в «черном списке», то Internet Explorer 8 предупредит об опасности, окрасив заголовок вкладки в красный цвет. Адресная строка также поменяет свой цвет, а на странице будет отображаться информация, с чем связаны подобные меры безопасности. Сегодня большинство сайтов используют сочетание данных (MashUp) собственного содержания и информации, полученной с других сайтов, например, интерактивные веб-карты с дополнительными слоями. Однако более половины веб-приложений уязвимы для атак межсайтового выполнения сценариев (XSS). Internet Explorer 8 – первый браузер, который имеет встроенную защиту от подобных угроз. Отметим, что его фильтр способен обучаться – любой пользователь может внести посильный вклад в повышение безопасности интернета, сообщая о подозрительных ресурсах. В качестве наиболее показательной иллюстрации можно привести проблемы, которые обнаружились в начале сентября 2009 года в фреймворке Ruby On Rails. В результате на сервере микроблоггинга Twitter (и некоторых других сайтах, например, Basecamp), написанных на его основе, у злоумышленника появилась возможность запустить на исполнение произвольный код в JavaScript. Уязвимость реализовывалась через межсайтовый скриптинг. Однако пользователи Internet Explorer 8 с включенным фильтром, благодаря встроенному XSS-фильтру, были защищены от подобных атак. Остальные браузеры также используют контентные фильтры для обеспечения безопасности, но лишь в Internet Explorer 8 сигнатурный фильтр дополнен «поведенческим» блоком, а также тесно интегрирован с другими защитными механизмами операционной системы. В частности, SmartScreen может работать в паре со «средством удаления вредоносного программного обеспечения» (Malicious Software Removal Tool) и «защитником Windows» (Windows Defender), которые можно загрузить с сайта Microsoft вместе с Internet Explorer 8 (в последних ОС от Microsoft

они уже предустановлены и требуют лишь периодического обновления БД).

Фильтр SmartScreen — это функция обозревателя Internet Explorer 8, помогающая избежать построенных с использованием социальной инженерии вредоносных фишинговых веб-сайтов и интернет-мошенничества при просмотре ресурсов Интернета.

Фильтр SmartScreen:

- проверяет веб-сайты по динамически обновляемому списку заявленных случаев фишинга и сайтов;
- проверяет загружаемые программы по динамически обновляемому списку заявленных сайтов с вредоносными программами;
- помогает предотвратить посещение фишинговых веб-сайтов и других содержащих вредоносные программы веб-сайтов, так как это может привести к краже идентификационных данных.

Если фильтр SmartScreen включен, то при попытке посетить веб-сайт, в отношении которого поступало сообщение, открывается приведенное ниже окно с рекомендацией не переходить на небезопасный веб-сайт.

Фильтр SmartScreen играет важнейшую роль в обеспечении вашей безопасности в сети. Авторы вредоносного ПО постоянно придумывают новые способы проникновения своего кода на компьютеры. Мы внесли ряд изменений, призванных защитить пользователей, сделав риски посещения вредоносных сайтов более ясными и воспрепятствовав бездумному игнорированию предупреждений. Посему настоятельно рекомендую включить фильтр SmartScreen и продолжить отсылать нам данные обратной связи.

Data Execution Prevention

Data Execution Prevention (DEP) (англ. Предотвращение выполнения данных) — функция безопасности, встроенная в семейство операционных систем Windows, которая не позволяет приложению исполнять код из области памяти, помеченной как «только для данных». Она позволит предотвратить некоторые атаки, которые, например, сохраняют код в такой области с помощью переполнения буфера. DEP работает в двух режимах: аппаратном, для процессоров, которые могут помечать страницы как «не для исполнения кода», и программном, для остальных процессоров.

В Internet Explorer 7 в Windows Vista была впервые представлена (выключенная по умолчанию) функция защиты памяти, которая помогала избегать атаки из Интернета. Эта функция также известна как Data Execution Prevention (DEP) или No-Execute (NX). В Internet Explorer 8 в Windows Server 2008 и Windows Vista SP1 данная функция будет по умолчанию включена.

DEP помогает избежать атак путем предотвращения запуска кода, размещенного в участке памяти, помеченном как неисполняемый. DEP в

комбинации с другими технологиями, какASLR, делает процесс использования взломщиками разнообразных уязвимостей, связанных с памятью (например, переполнение буфера) намного более сложным. Лучше всего данная технология работает для Internet Explorer и для загружаемых надстроек. Для обеспечения всех этих функций безопасности от пользователя не требуется никаких дополнительных действий и никаких запросов ему показано не будет.

В Internet Explorer 7 по причинам совместимости DEP по умолчанию был отключен. Несколько популярных надстроек были несовместимы с DEP и могли вызвать завершение работы Internet Explorer при включенном DEP. Чаще всего проблема состояла в том, что эти дополнения были скомпилированы с использованием старой библиотеки ATL. До версии 7.1 SP1 ATL полагалась на динамически сгенерированный код, который несовместим с DEP. И хотя большинство разработчиков популярных надстроек уже выпустили обновленные для DEP версии, некоторые могут быть не обновлены до выхода Internet Explorer 8.

К счастью новые DEP API были добавлены в Windows Server 2008 и Vista SP1, чтобы позволить использование DEP, сохраняя совместимость со старыми версиями ATL. Новые API позволяют Internet Explorer использовать DEP, при этом старые надстройки, использующие старые версии ATL, не станут причиной завершения работы Internet Explorer.

В редких случаях, когда дополнение несовместимо с DEP по какой-либо иной причине, отличной от использования старой версии ATL, опция в групповых политиках позволит организациям выключать DEP для Internet Explorer до тех пор, пока обновленная версия дополнения не будет развернута. Локальные администраторы могут контролировать использование DEP, запустив Internet Explorer как администраторы и выключив опцию защиты памяти.

Проверка состояния вашей безопасности

Увидеть, какие именно процессы в Windows Vista защищены DEP, вы можете во вкладке диспетчера задач. Для более ранних версий Windows вы можете использовать Process Explorer. В обоих случаях проверьте, чтобы была отмечена опция Data Execution Prevention в выборе отображаемых колонок.

HTTPS

HTTPS (аббр. от англ. HyperText Transfer Protocol Secure) — расширение протокола HTTP, для поддержки шифрования в целях повышения безопасности. Данные в протоколе HTTPS передаются поверх криптографических протоколов SSL или TLS. В отличие от HTTP с TCP-портом 80, для HTTPS по умолчанию используется TCP-порт 443.

HTTPS не является отдельным протоколом. Это обычный HTTP, работающий через шифрованные транспортные механизмы SSL и TLS. Он обеспечивает защиту от атак, основанных на прослушивании сетевого соединения — от снифферских атак и атак типа man-in-the-middle, при условии, что будут использоваться шифрующие средства и сертификат сервера проверен и ему доверяют.

По умолчанию HTTPS URL использует 443 TCP-порт (для незащищённого HTTP — 80). Чтобы подготовить веб-сервер для обработки https-соединений, администратор должен получить и установить в систему сертификат открытого ключа для этого веб-сервера. В TLS используется как асимметричная схема шифрования (для выработки общего секретного ключа), так и симметричная (для обмена данными, зашифрованными общим ключом). Сертификат открытого ключа подтверждает принадлежность данного открытого ключа владельцу сайта. Сертификат открытого ключа и сам открытый ключ посылаются клиенту при установлении соединения; закрытый ключ используется для расшифровки сообщений от клиента. Существует возможность создать такой сертификат, не обращаясь в ЦС. Подписываются такие сертификаты этим же сертификатом и называются самоподписанными (self-signed). Без проверки сертификата каким-то другим способом (например, звонок владельцу и проверка контрольной суммы сертификата) такое использование HTTPS подвержено атаке man-in-the-middle.

Эта система также может использоваться для аутентификации клиента, чтобы обеспечить доступ к серверу только авторизованным пользователям. Для этого администратор обычно создаёт сертификаты для каждого пользователя и загружает их в браузер каждого пользователя. Также будут приниматься все сертификаты, подписанные организациями, которым доверяет сервер. Такой сертификат обычно содержит имя и адрес электронной почты авторизованного пользователя, которые проверяются при каждом соединении, чтобы проверить личность пользователя без ввода пароля.

В HTTPS для шифрования используется длина ключа 40, 56, 128 или 256 бит. Некоторые старые версии браузеров используют длину ключа 40 бит (пример тому — IE версий до 4.0), что связано с экспортными ограничениями в США. Длина ключа 40 бит не является сколько-нибудь надёжной. Многие современные сайты требуют использования новых версий браузеров, поддерживающих шифрование с длиной ключа 128 бит, с целью обеспечить достаточный уровень безопасности. Такое шифрование значительно затрудняет злоумышленнику поиск паролей и другой личной информации.

Традиционно на одном IP-адресе может работать только один HTTPS сайт. Для работы нескольких HTTPS-сайтов с различными сертификатами применяется расширение TLS под названием Server Name Indication (SNI).

(Содержание)

1. 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?

(Содержание)

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.

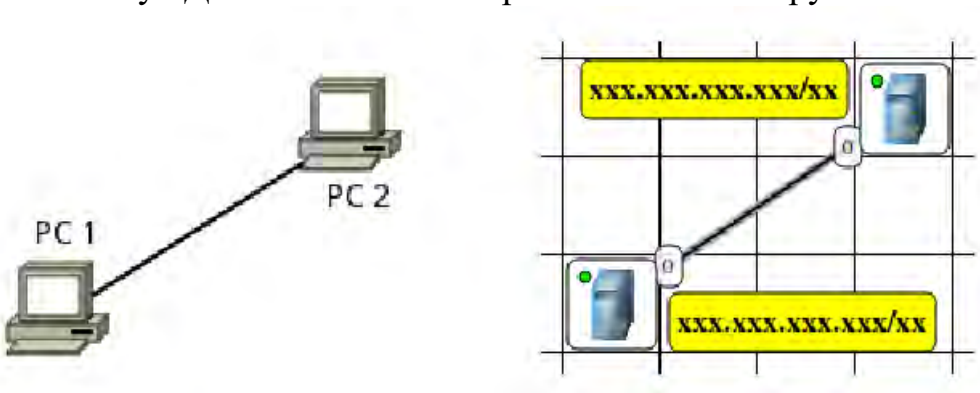

Рисунок 2.1. – Соединение двух ЭВМ напрямую.

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 сме-

нил 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.

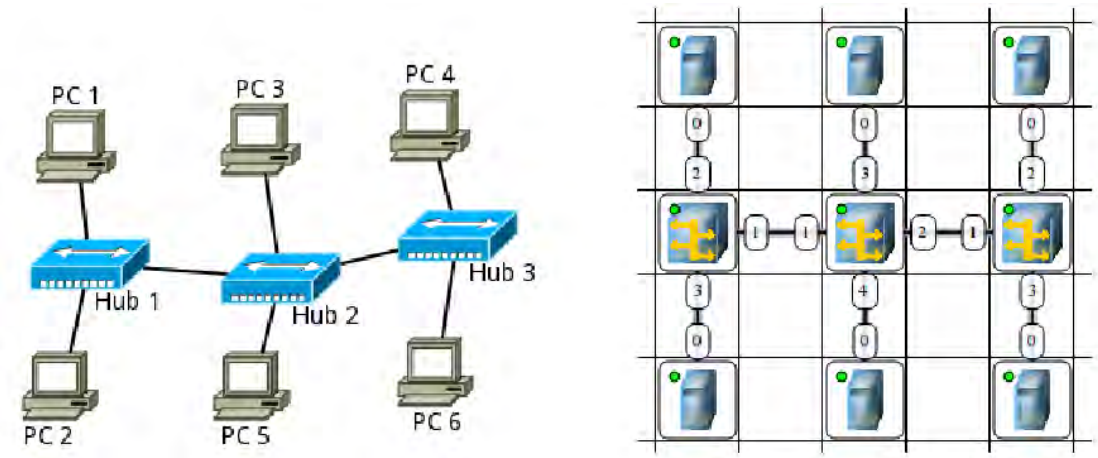

Рисунок 2.2. – Построение ЛВС на концентраторах

Построение ЛВС на коммутаторах

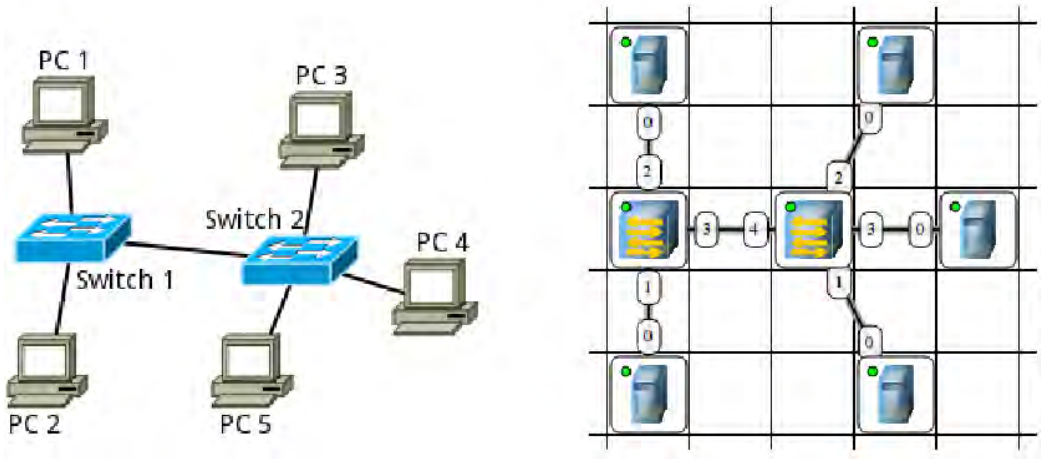

Рисунок 2.3. – Построение ЛВС на коммутаторах

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.

(Содержание)

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.

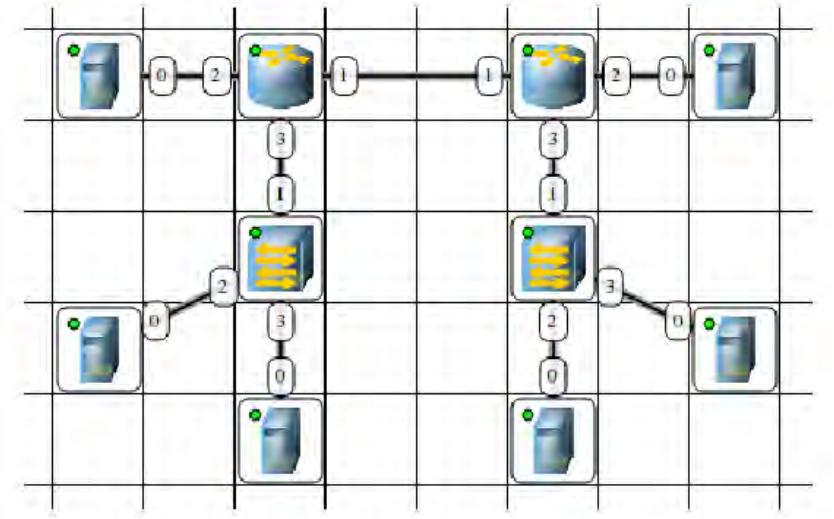

Рисунок 2.4. – Модель ЛВС

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.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. 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?

(Содержание)

2.3. 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.

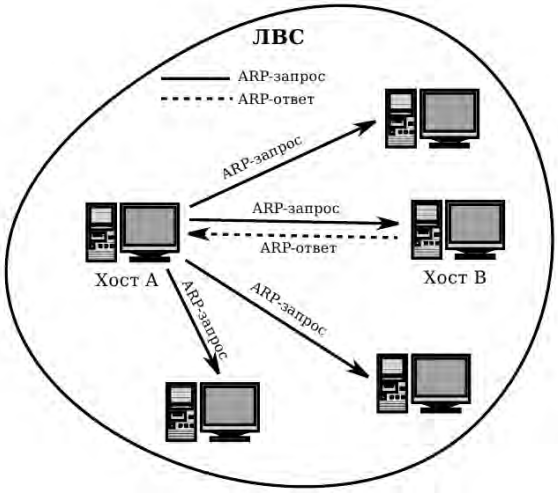

Рисунок 2.5 – Схема работы протокола ARP

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 по-

сылается широковещательный запрос.

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

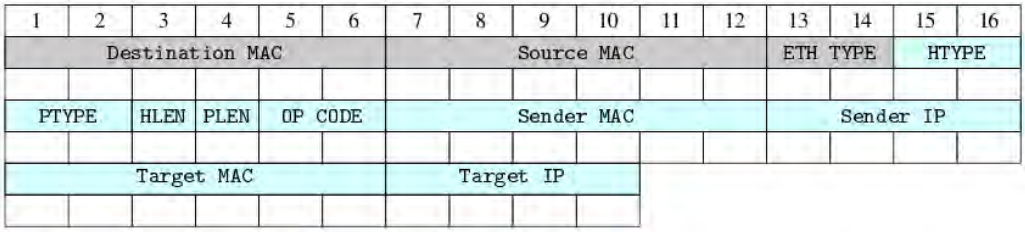

Рисунок 2.6 – Структура кадра ARP

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 се-

ти, 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.

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

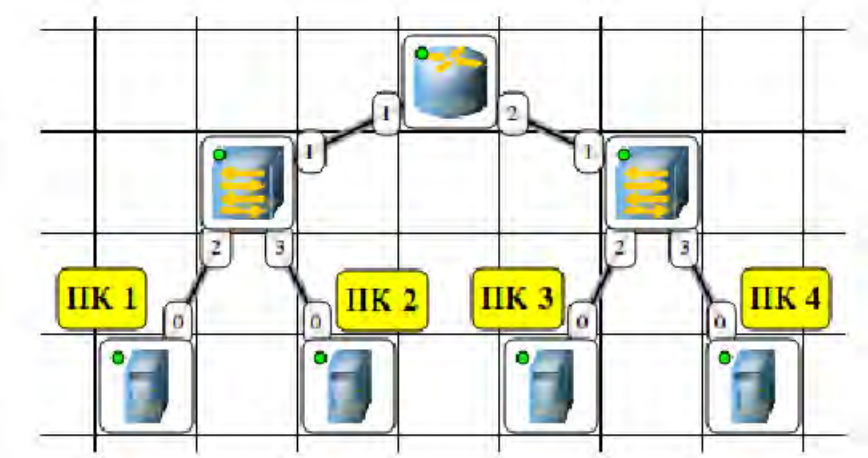

Рисунок 2.7 – Структура ЛВС для ознакомления с ARP протоколом

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.

(Содержание)

2.4. 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.

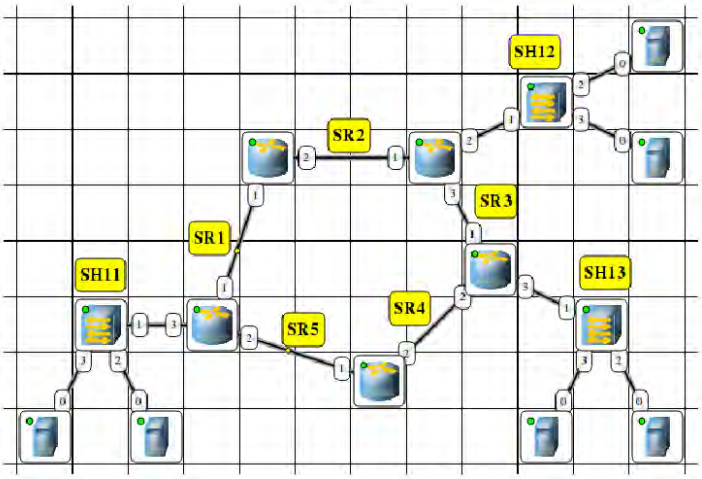

Рисунок 2.8. Структура сети для знакомства с протоколом RIP

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.

(Содержание)

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.

(Содержание)

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.

(Содержание)

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:

2. Титульный лист.
3. Цель работы.
4. Заполненные таблицы.
5. 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.

(Содержание)

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.

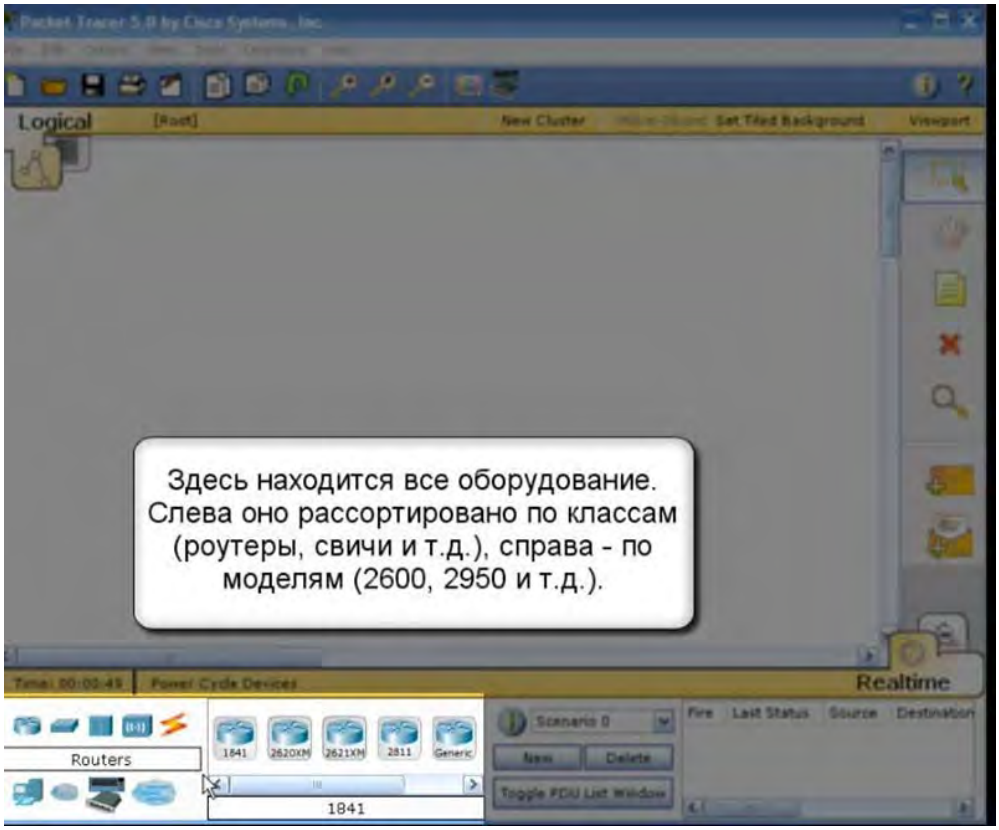

Рисунок 2.10. Главное окно

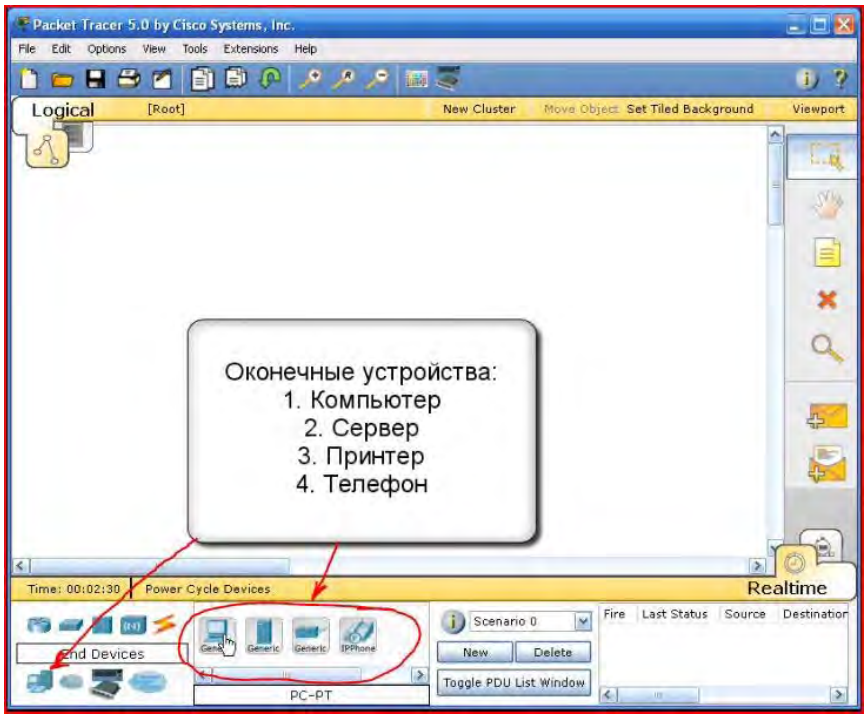

Рисунок 2.11. Доступное оборудование

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).

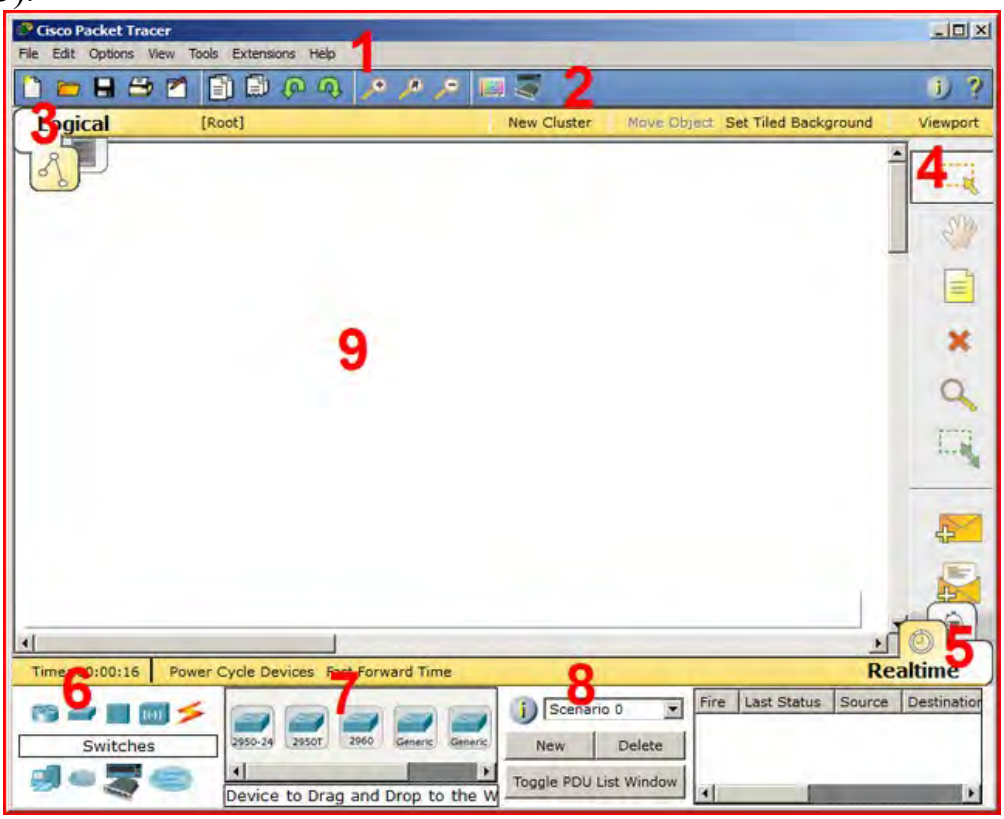

Рисунок 2.13. Главное меню программы

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).

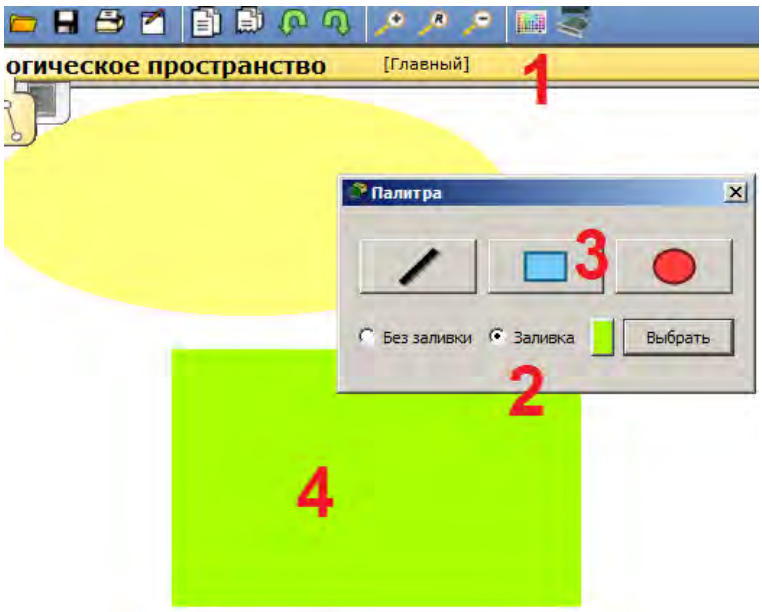

Рисунок 2.14. Палитра

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).

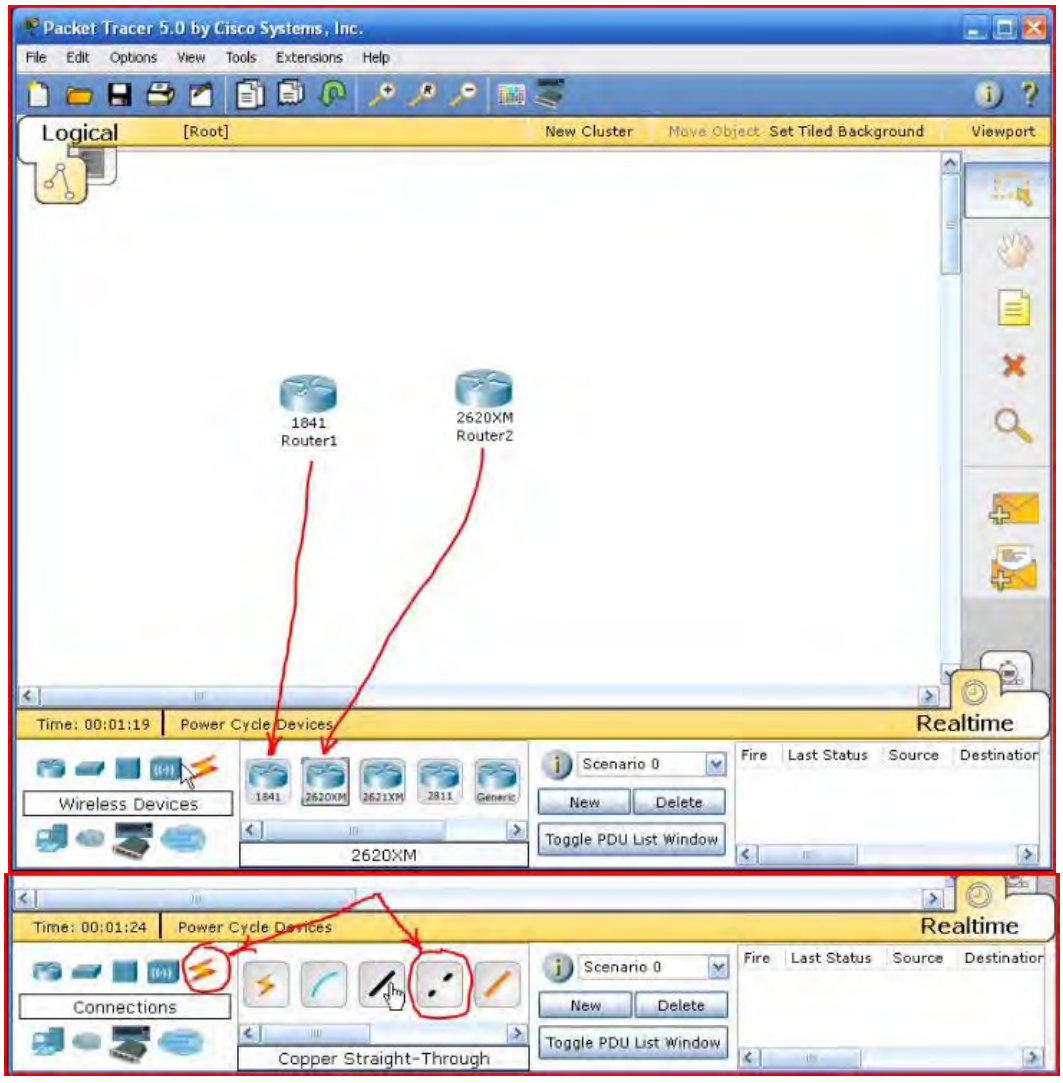

Рисунок 2.15. Моделирование сети

Соединить порты роутеров (рис 2.16).

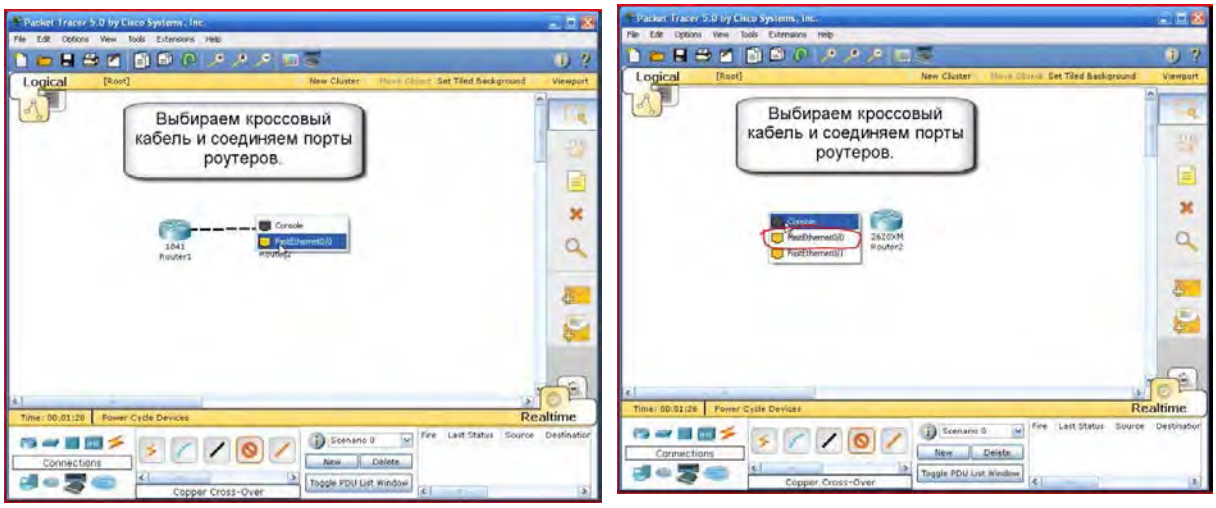

Рисунок 2.16. Соединение роутеров

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

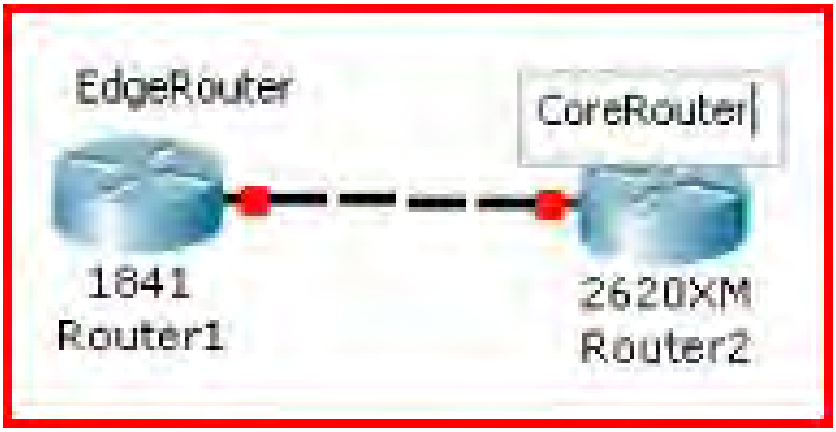

Рисунок 2.17. Подписи

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).

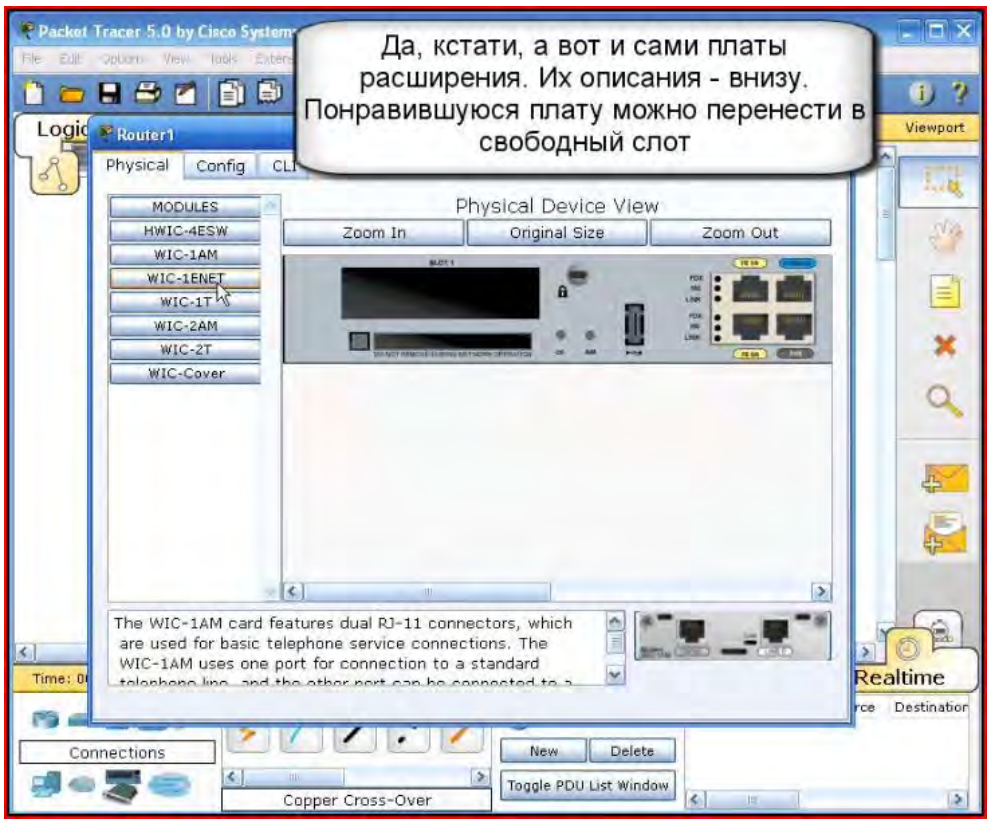

Рисунок 2.18. Физическая конфигурация

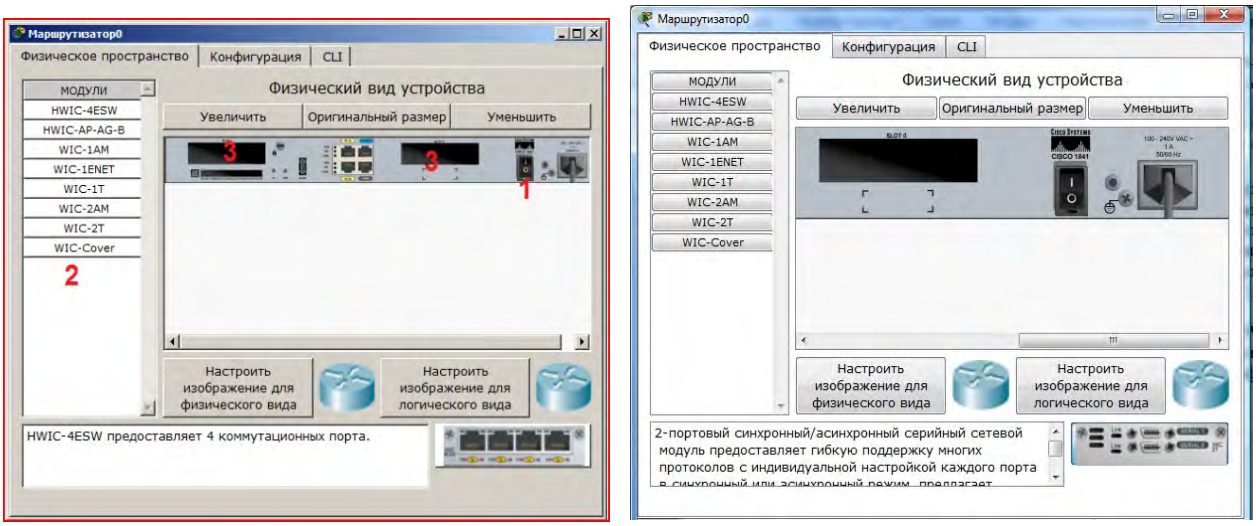

Рисунок .19. Изменение физической конфигурации

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

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.

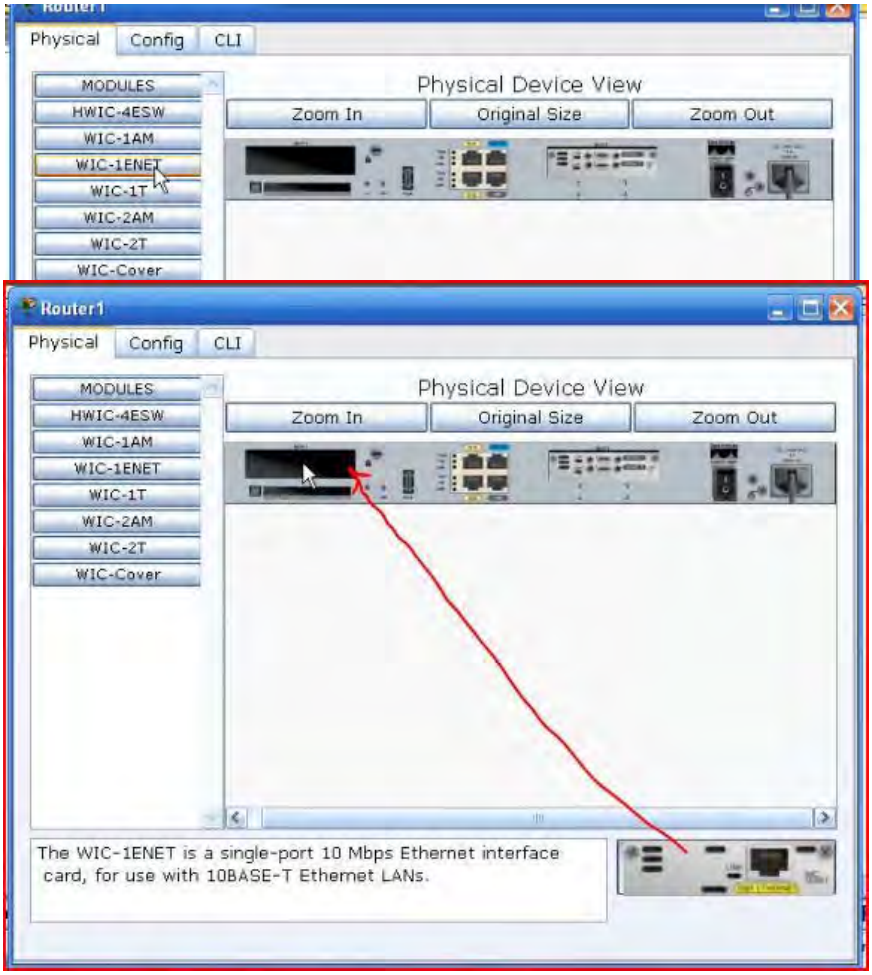

Рисунок 2.20 – Последующее изменение физической конфигурации

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).

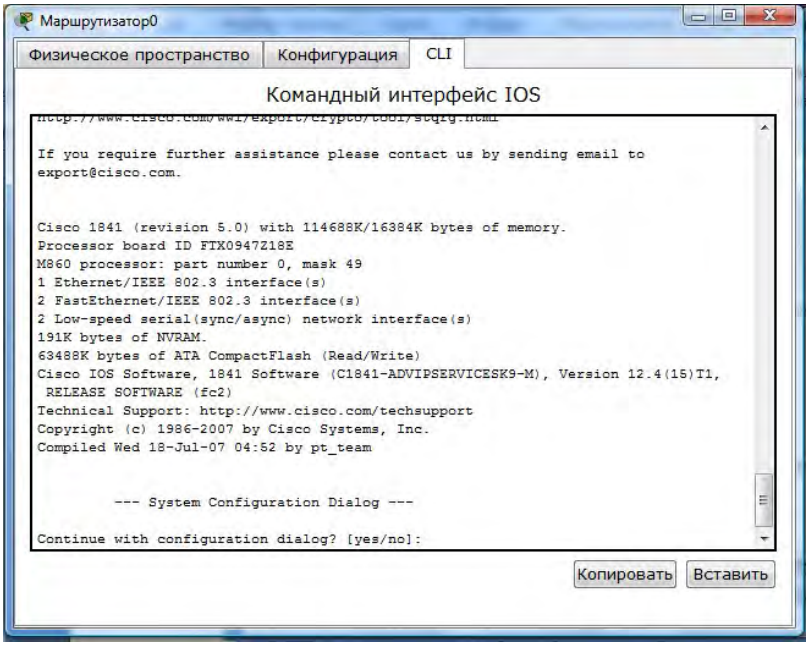

Рисунок 2.21. Командная строка

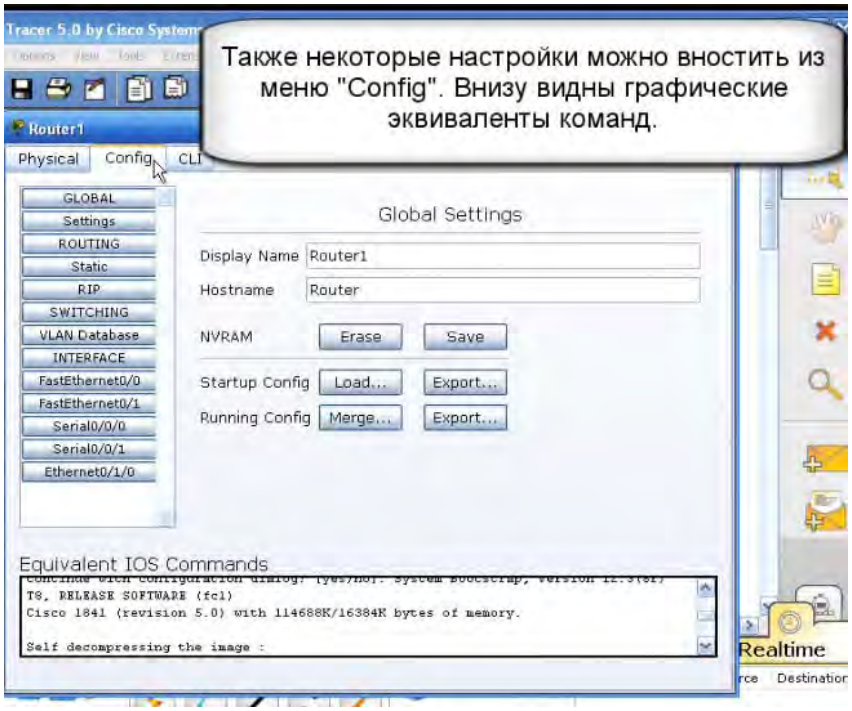

Рисунок 2.22. Настройка во вкладке Config

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

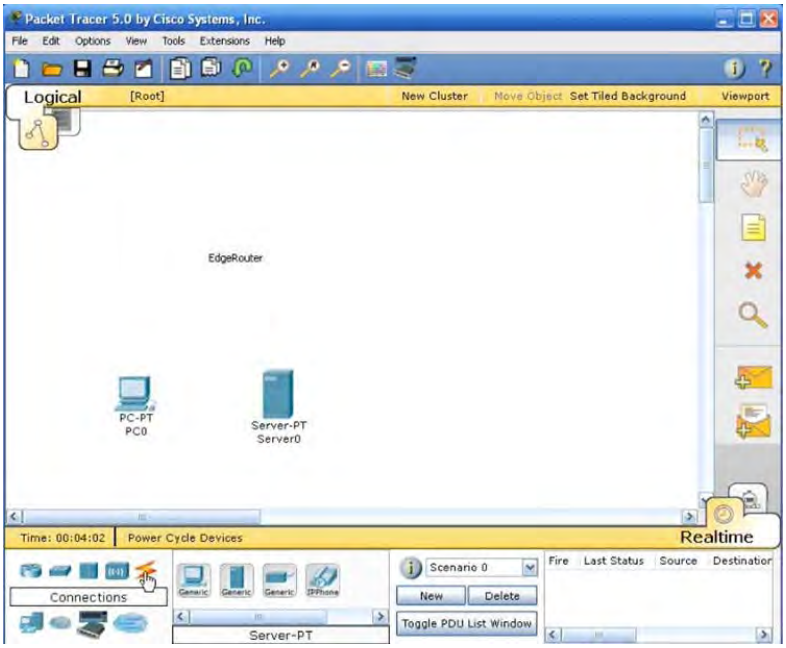

Рисунок 2.23. ПК и сервер

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).

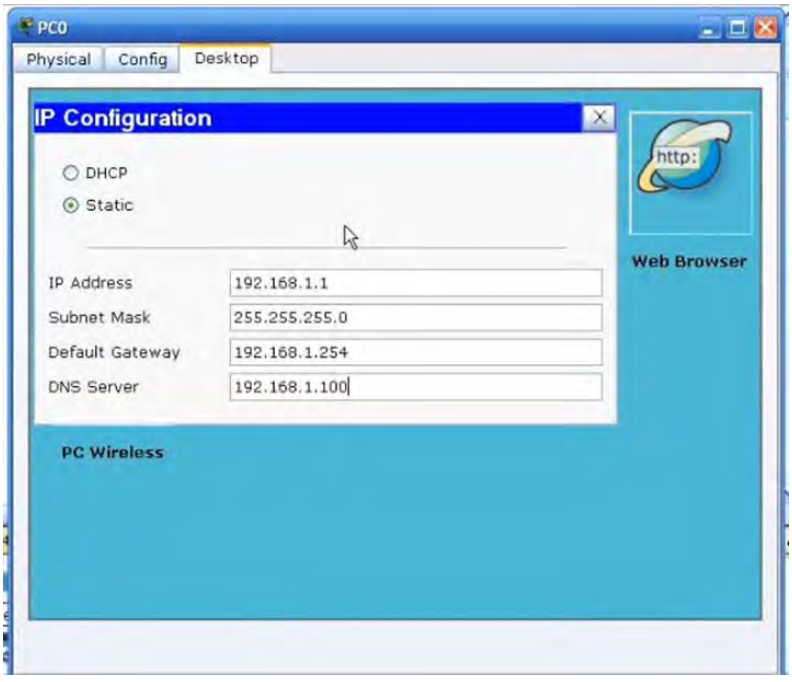

Рисунок 2.26. Настройка ПК

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

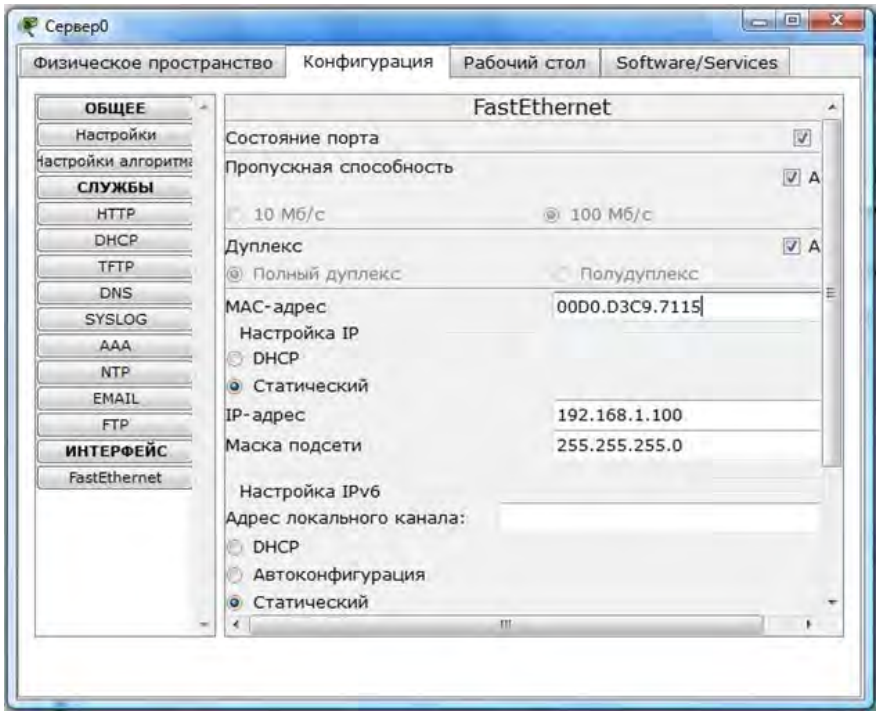

Рисунок 2.27. Настройки FastEthernet сервера

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

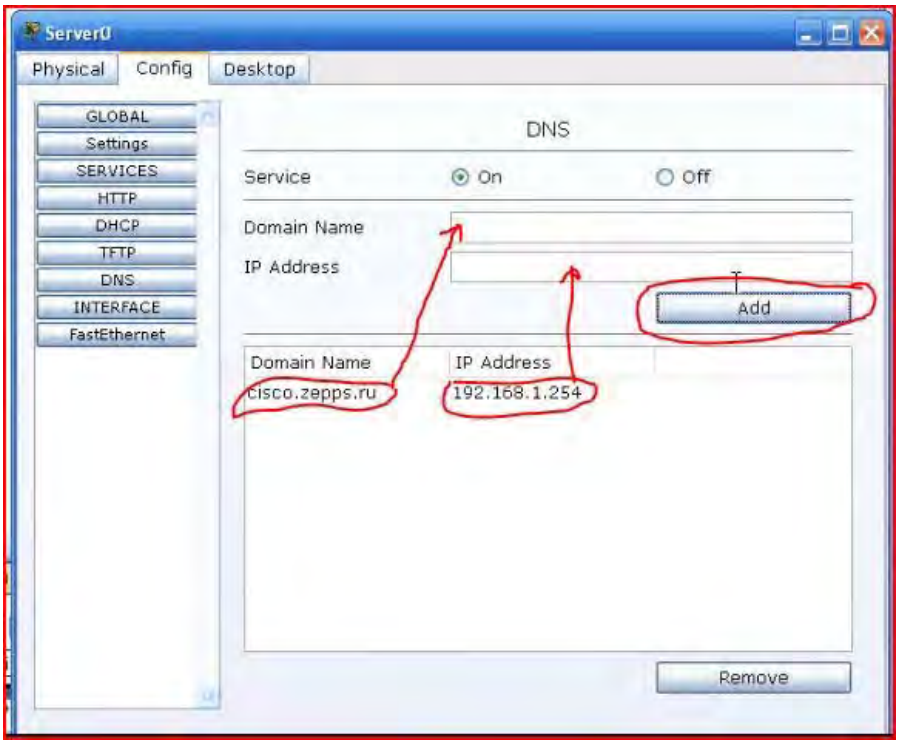

Рисунок 2.28. Настройки DNS

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

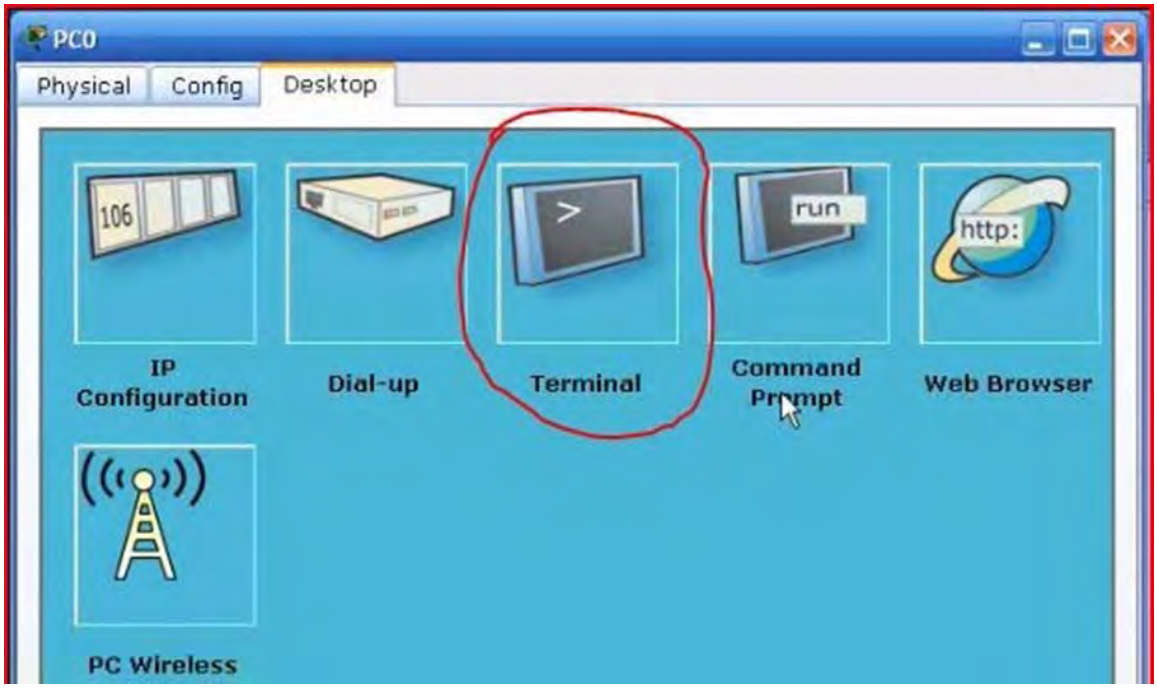

Рисунок 2.29. Кнопка вызова терминала

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

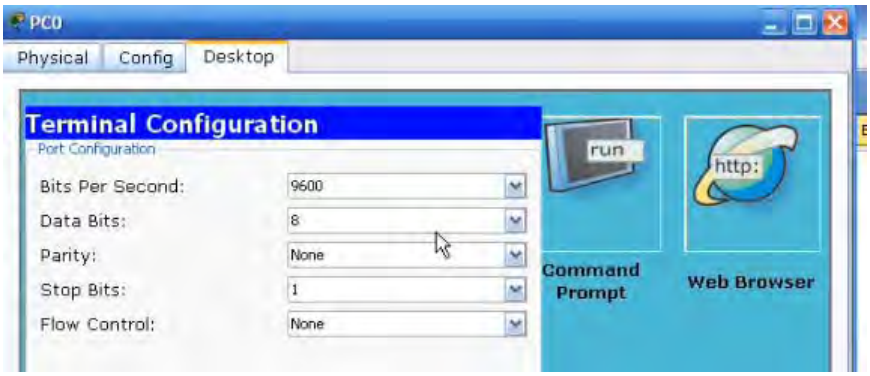

Рисунок 2.30. Настройки терминала

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

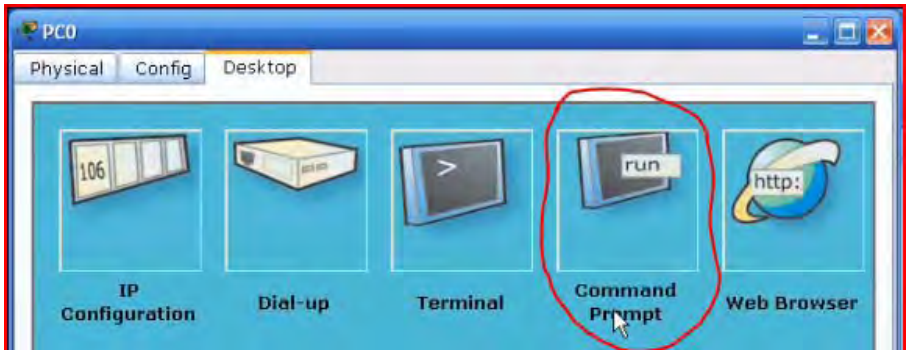

Рисунок 2.31. Кнопка вызова командной строки

Пропингуем сеть командой ping 192.168.1.100 (рис. 2.32).

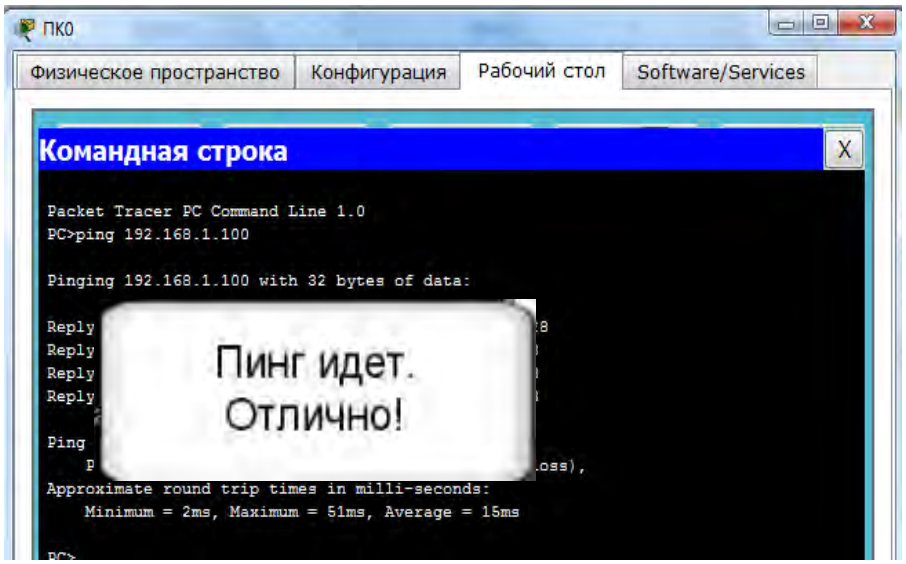

Рисунок 2.32. Результат выполнения команды ping

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).

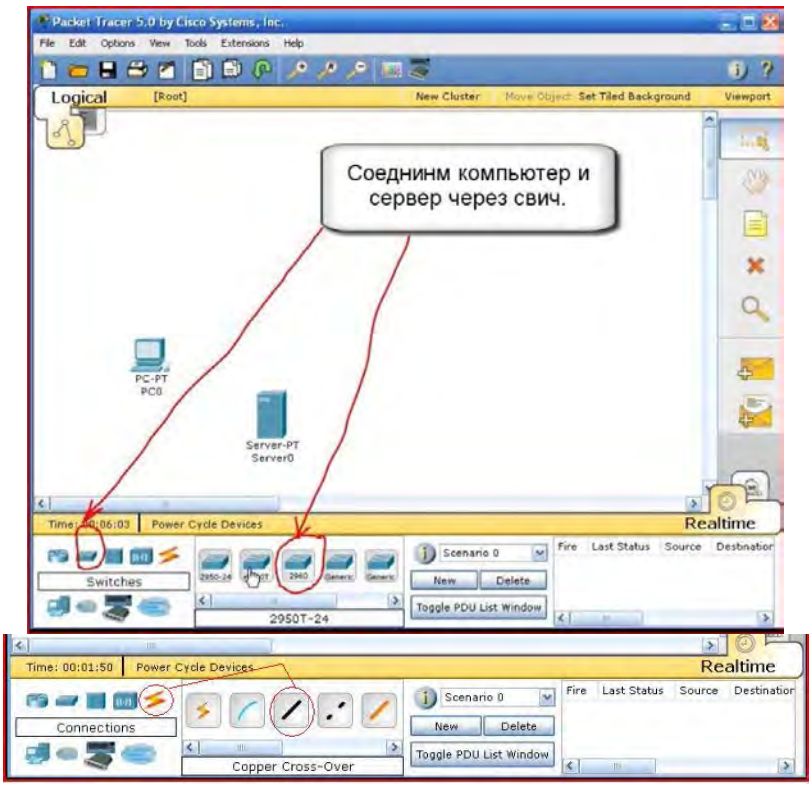

Рисунок 2.33. Соединение компьютера и сервера

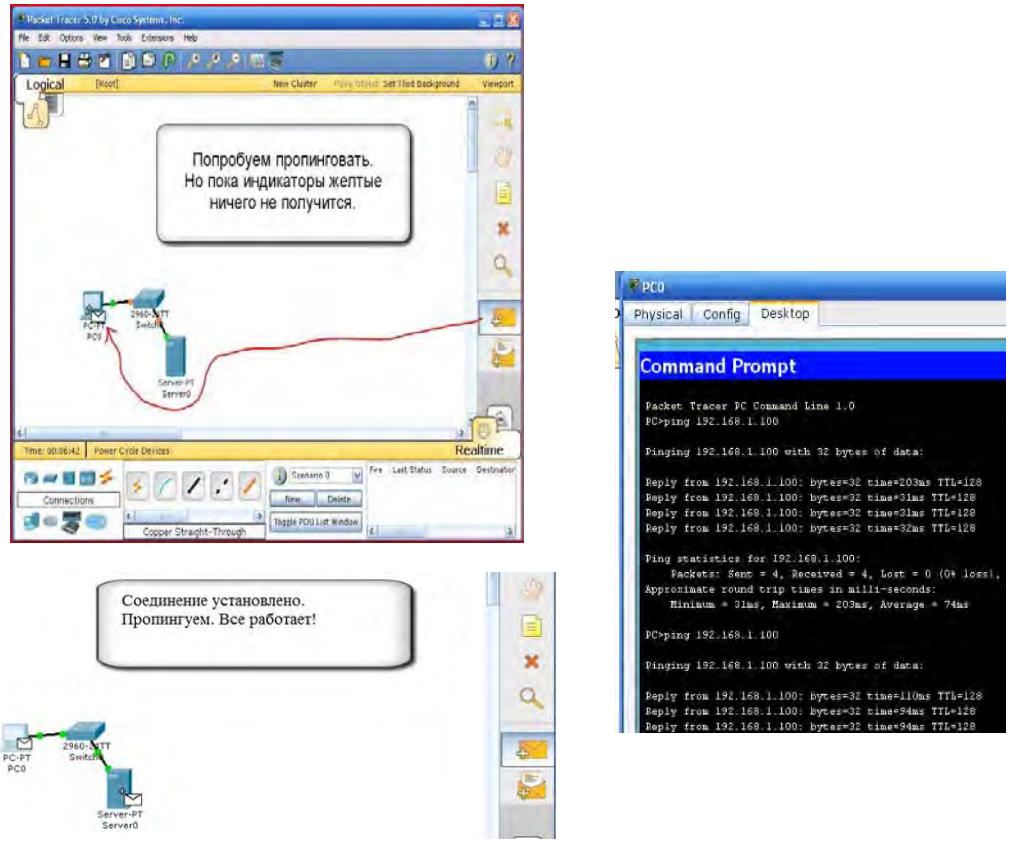

Рисунок 2.34. Успешное соединение при помощи свитча.

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

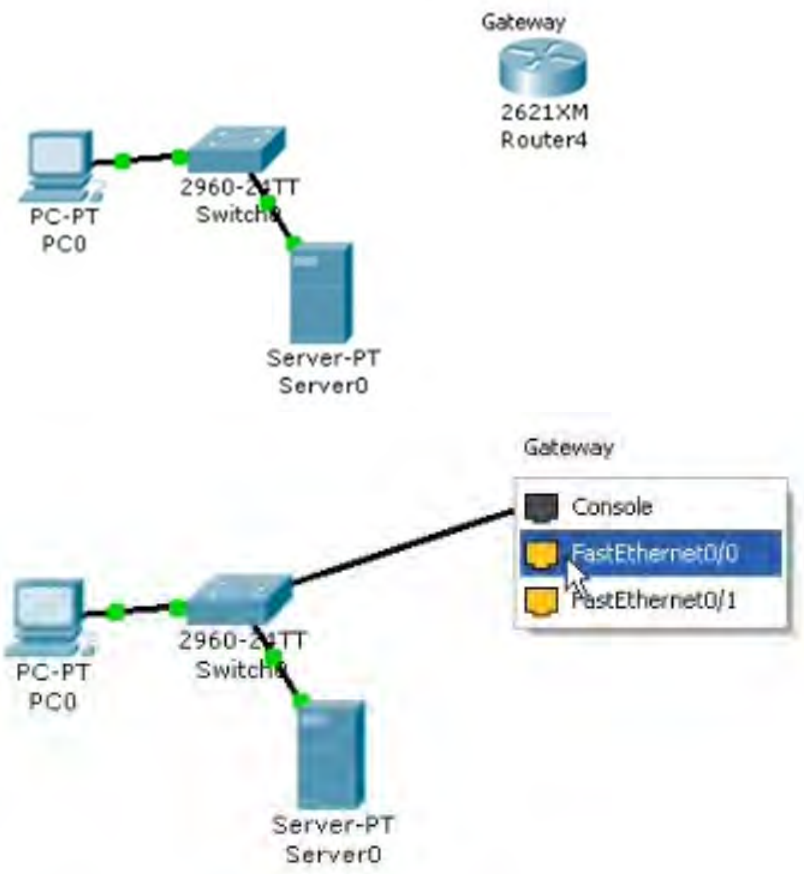

Рисунок 2.35. Подключение роутера

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

Рисунок 2.36. Добавление устройства

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

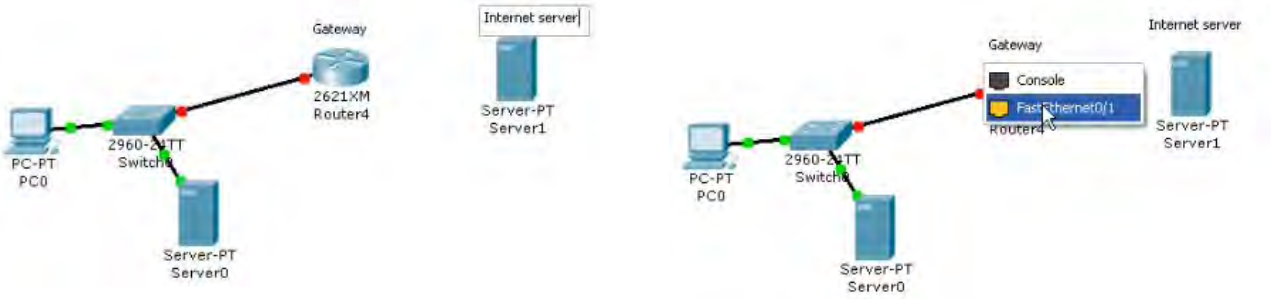

Рисунок 2.37. Добавление Internet-сервера

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

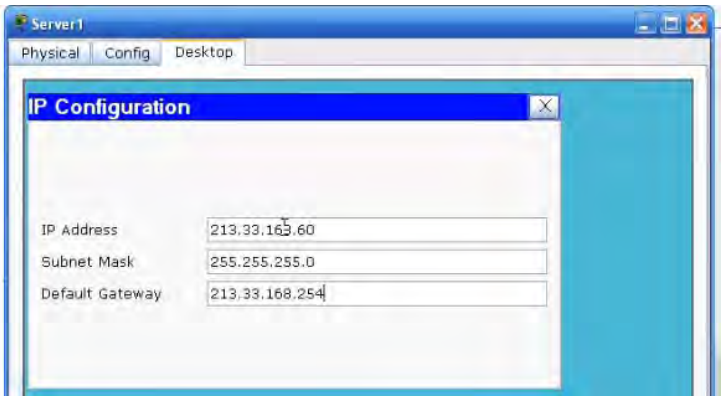

Рисунок 2.39. Настройки сервера 1

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

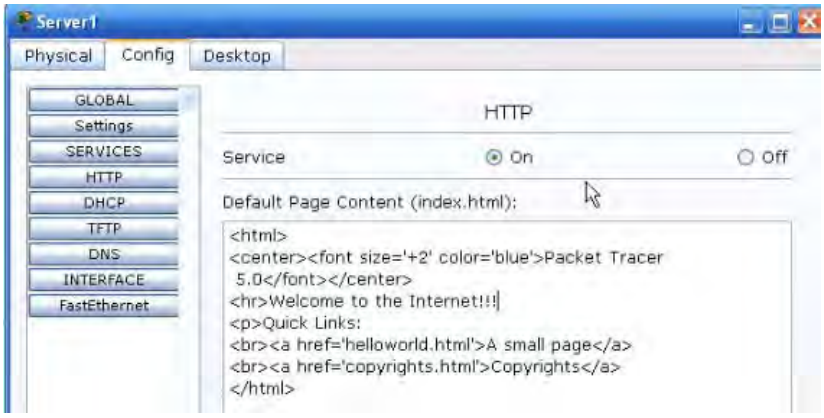

Рисунок 2.40. HTML-страницы

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):

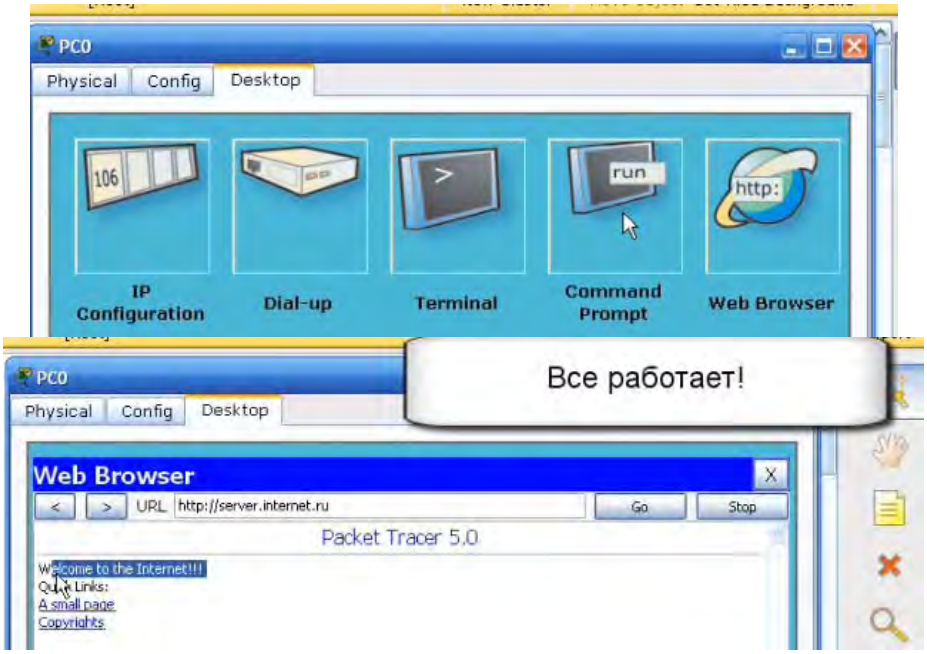

Рисунок 2.43. Настройки для WEB-браузера

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):

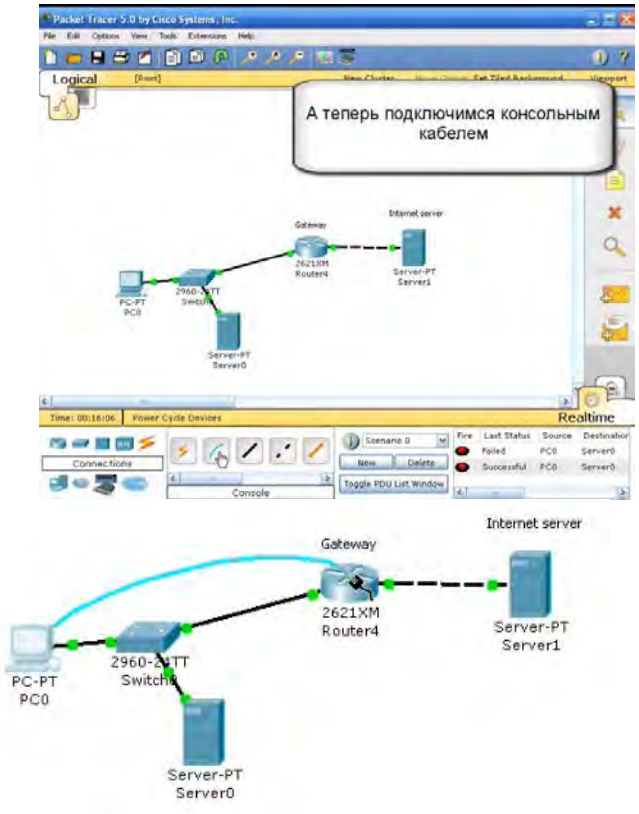

Рисунок 2.44 – Подключение консольным кабелем

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

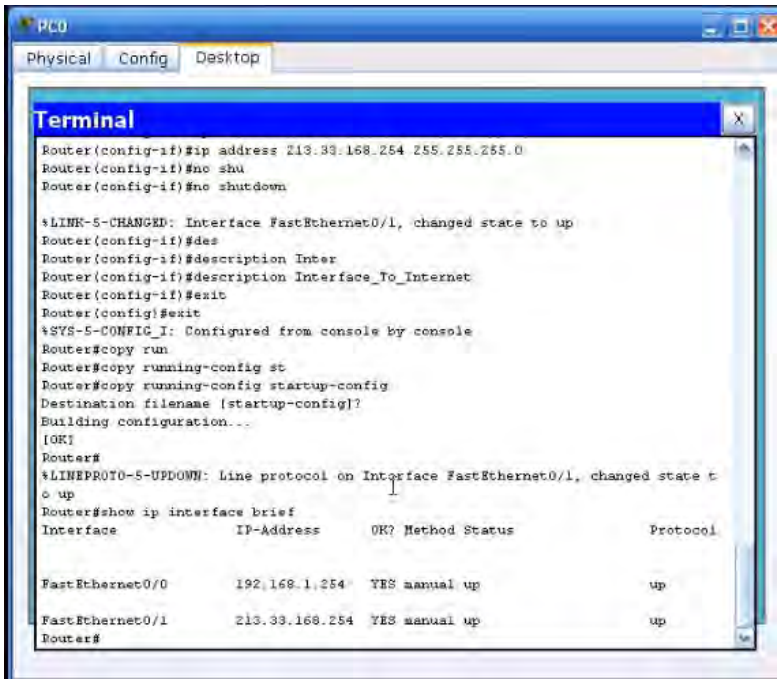

Рисунок 2.45. Системное сообщение роутера

Проверим канал связи (рис. 2.46).

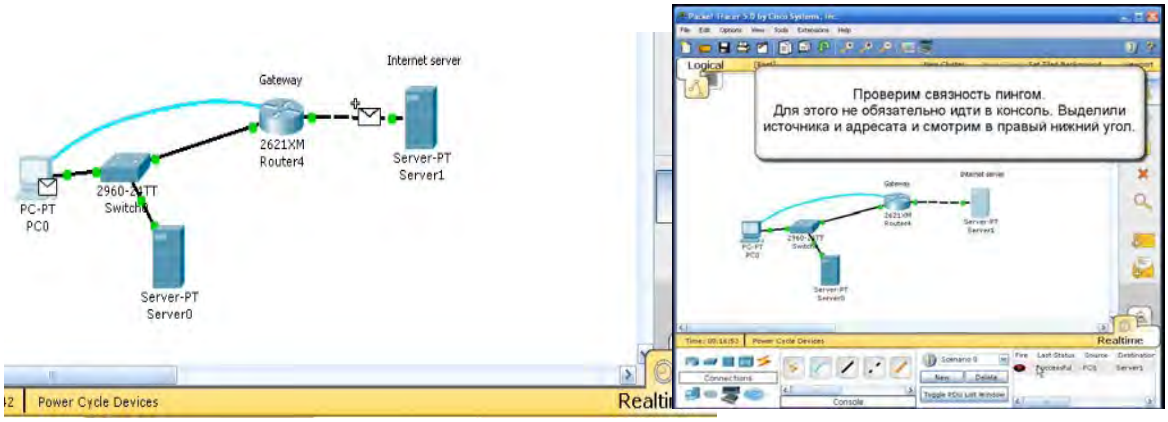

Рисунок 2.46. Проверка канала связи

Сформируем сложный запрос (рис. 2.47).

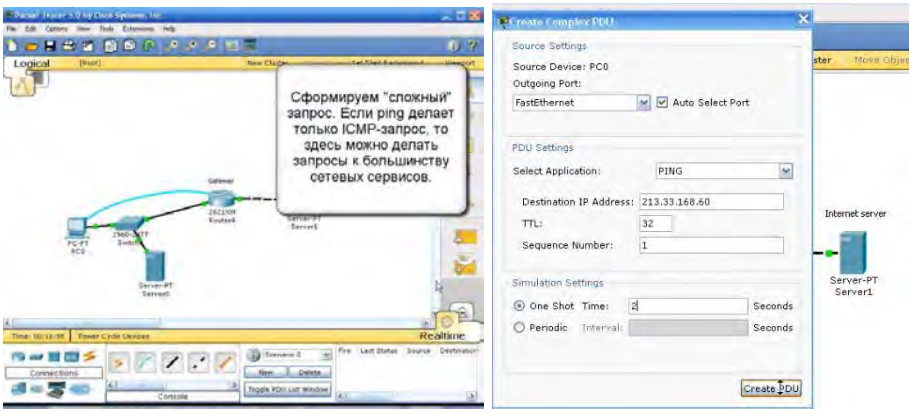

Рисунок 2.47 – Формирование сложного запроса

Отключим один интерфейс (рис. 2.48).

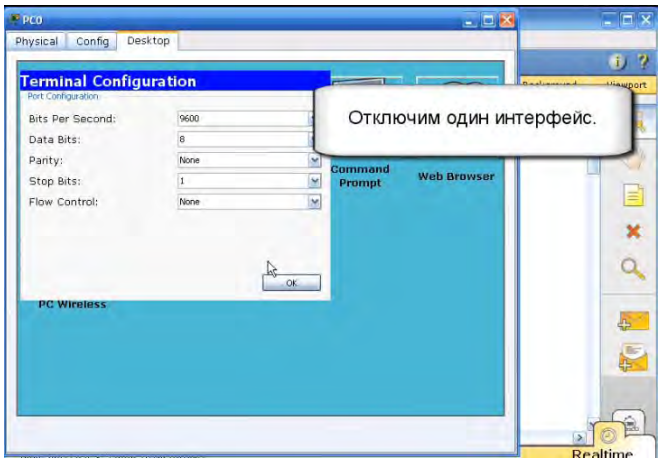

Рисунок 2.48. Отключение интерфейса

Пинг не пройдет (рис. 2.49).

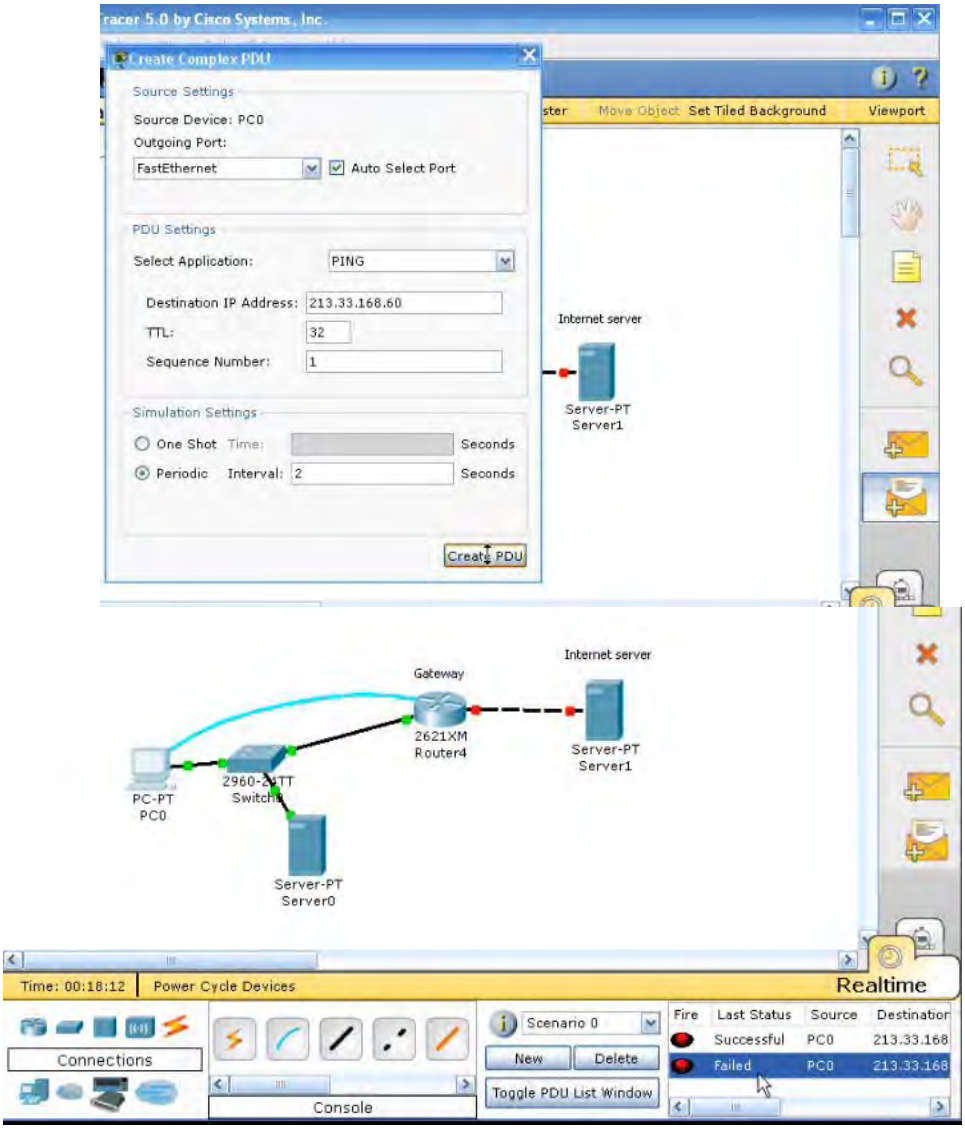

Рисунок 2.49. Формирование запроса

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.

(Содержание)

2.10. Лабораторная работа 10.
Соединение двух сетей.

Цель работы: Научиться соединять две сети в эмуляторе PT5

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.

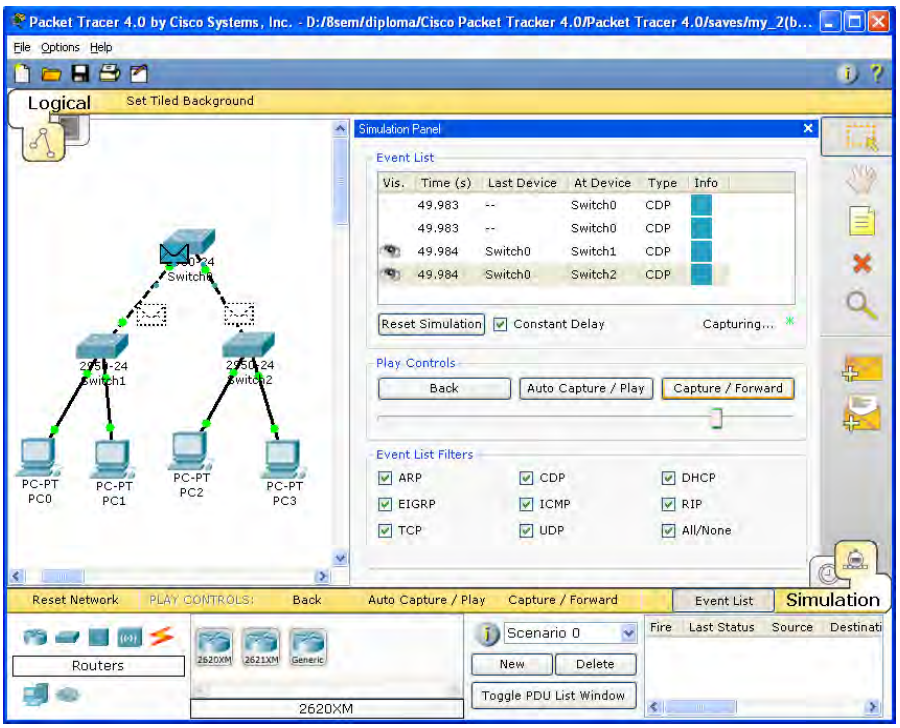

Рисунок 2.50. Режим «Симуляции» в Cisco Packet Tracer

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).

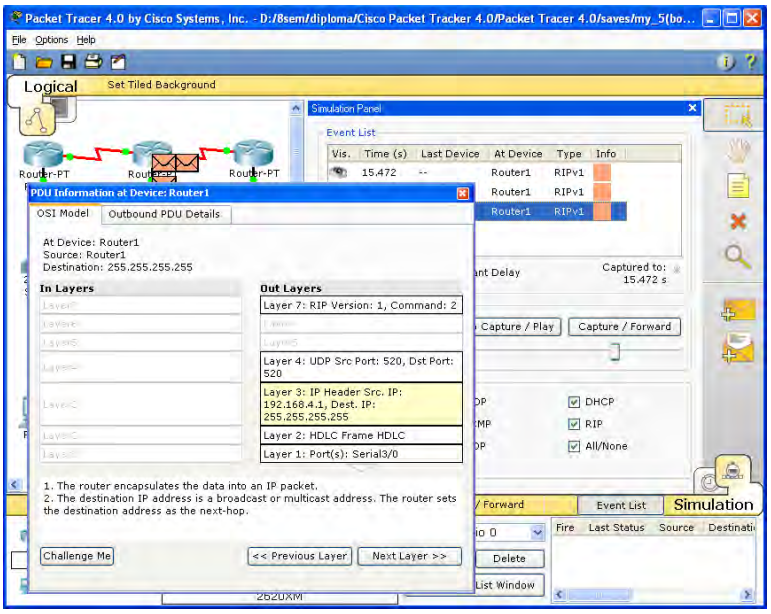

Рисунок 2.51. Анализ семиуровневой модели OSI в Cisco Packet Tracer

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/

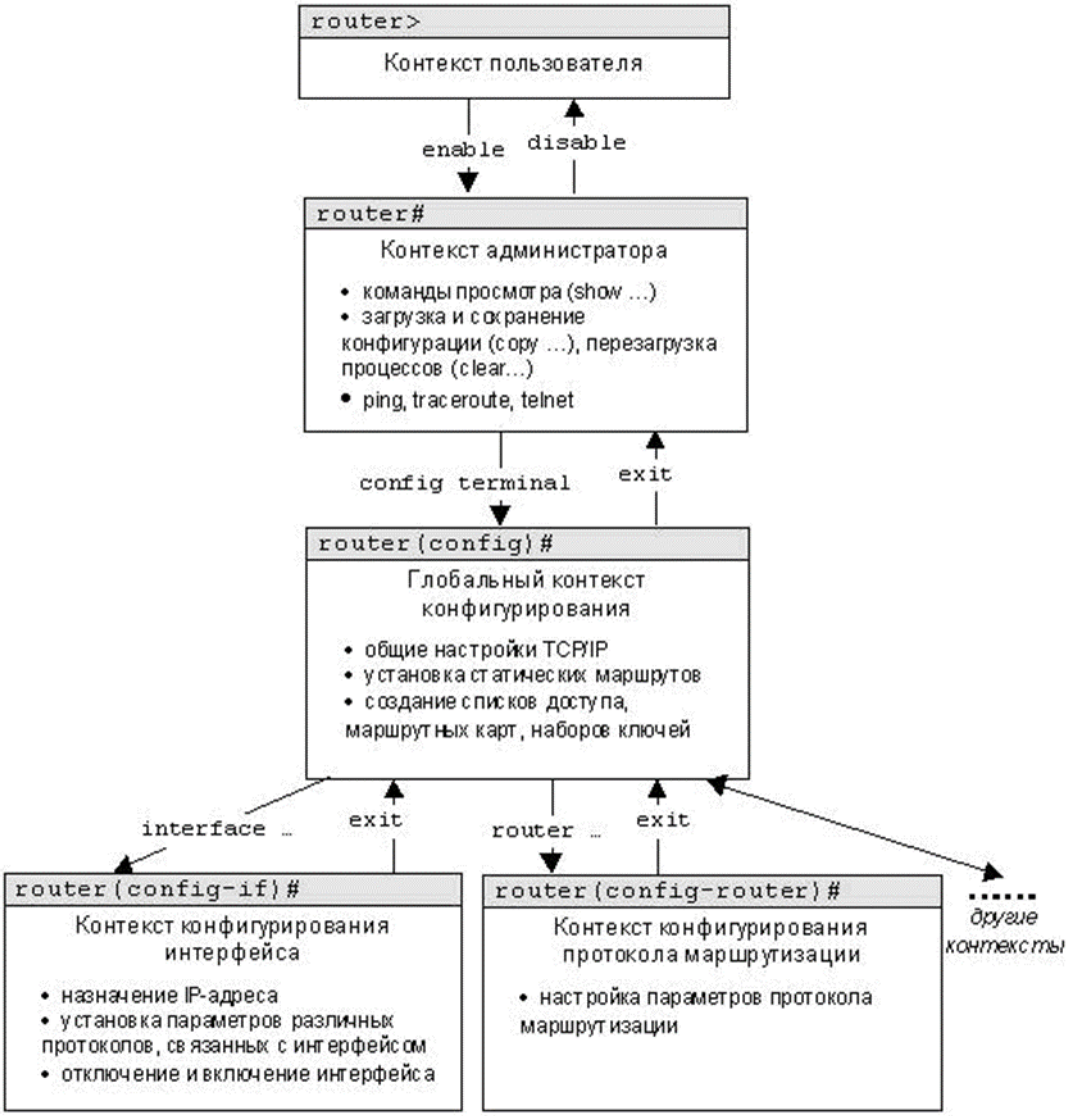

Рисунок 2.52. Схема контекстов Cisco IOS

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:

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

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

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 RE-
TURN to get started.
```

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)#inter
face vlan 1
Switch(config-if)#


Rout-
```

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 «Hostname»

Данная команда используется для изменения имени конфигурируемого устройства. Формат команды:

hostname *новое_имя*

Пример выполнения команды:

```
Rout-
er(config)#hostna
```

Как видно, маршрутизатор поменял своё имя с Router на R1.

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
Router(config-router)#
```

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 ***Rout-***
er(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:

```
Rout-
er(config)#interface se-
rial 2/0  Router(config-
```

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 ко-

манда 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#
```

Производится вывод таблицы маршрутизации.

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 "rip"
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-
chain
                   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

Пример выполнения команды:

```
Router#ping
77.134.25.133 Type
escape sequence to
abort.

Sending 5, 100-byte ICMP Echos to 77.134.25.133, timeout is 2 seconds:

..!!!

Success rate is 60 percent (3/5)
```

Каждый ICMP-пакет, на который был получен ответ, обозначается восклицательным знаком, каждый потерянный пакет – точкой.

Контекст пользователя

Команда «Enable».

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

Пример выполнения команды:

```
Router>enable

Router#
```

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 при-

вилегированный 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.

(Содержание)

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.

(Содержание)

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.

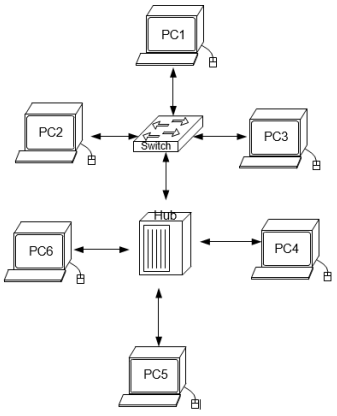

Рисунок 2.52. Проектируемая сеть

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), Approxi-
mate 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.

(Содержание)

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.

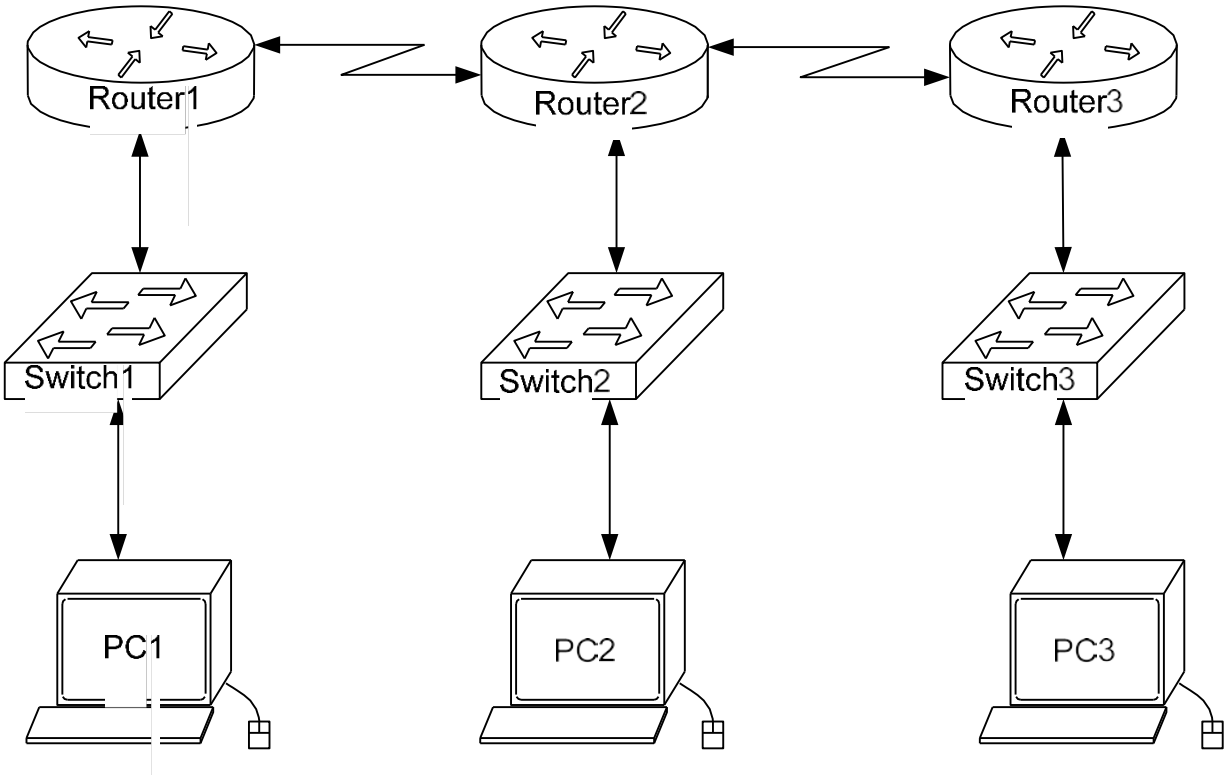

Рисунок 2.53. Схема сети

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.

(Содержание)

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.

ВСПОМОГАТЕЛЬНЫЙ РАЗДЕЛ

КОМПЬЮТЕРНЫЕ СЕТИ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дисциплина «Компьютерные сети» является одной из важнейших составных частей подготовки специалистов различных направлений, владеющих современными информационными технологиями, связанными с использованием средств вычислительной техники. С использованием компьютерных сетей строятся большинство автоматизированных систем, таких как: управления технологическими процессами, банковские системы, управления предприятиями, управления распределенными базами данных и др. Дисциплина связана с различными курсами по информатике, основам теории передачи данных, микропроцессорной технике.

Целью курса является обучение студентов базовым методам и средствам разработки, тестирования, эксплуатации, администрирования различных компьютерных сетей, в том числе и элементов компьютерной сети «Интернет».

В результате освоения дисциплины будущий специалист должен:

знать:

- теоретические основы построения и функционирования компьютерных сетей;
- технические и программные средства для создания наиболее распространенных типов компьютерных сетей;
- технические и программные средства тестирования, эксплуатации, администрирования различных компьютерных сетей;

приобрести: практические навыки по проектированию, созданию, конфигурированию, настройки и сопровождению функционирования основных типов компьютерных сетей.

Данная дисциплина базируется на следующих дисциплинах: «Каналы передачи данных», «Электроника и схемотехника», «Микропроцессорная техника», «Теория автоматического управления», обеспечивает базу для параллельного изучения дисциплины: «Технология и оборудование автоматизированного производства» и для выполнения соответствующего раздела дипломного проекта.

Методы (технологии) обучения

Основными методами обучения, отвечающими целям изучения учебной дисциплины, являются:

– элементы проблемного обучения (проблемное изложение, вариативное изложение, частично-поисковый метод), реализуемые на лекционных занятиях;

– элементы учебно-исследовательской деятельности, реализуемые на лабораторных занятиях и при самостоятельной работе;

– коммуникативные технологии (дискуссия, учебные дебаты, «мозговой штурм» и другие формы и методы), реализуемые на практических занятиях;

– проектные технологии, используемые при проектировании конкретного объекта, реализуемые при выполнении курсовой работы.

Организация самостоятельной работы студентов

При изучении учебной дисциплины рекомендуется использовать следующие формы самостоятельной работы:

– контролируемая самостоятельная работа в виде решения индивидуальных задач в аудитории во время проведения практических занятий под контролем преподавателя в соответствии с расписанием;

– управляемая самостоятельная работа, в том числе в виде выполнения индивидуальных расчетных заданий с консультациями преподавателя;

– подготовка рефератов по индивидуальным темам, в том числе с использованием патентных материалов;

– подготовка курсовой работы по индивидуальным заданиям, в том числе разноуровневым заданиям.

Согласно типовому учебному плану на изучение учебной дисциплины «Программное управление технологическим оборудованием» отведено 110 часов, в том числе 50 часов аудиторных занятий, из них лекции – 16 часов, лабораторные занятия – 34 часа.

Примерный тематический план

№	Название раздела, темы, занятия, вопросов	Количество аудиторных часов	
		Лекции	Лабораторные занятия
1	2	3	5
1.	Основные понятия и определения компьютерной сети. Введение. Основные понятия и определения. Классификация и характеристики компьютерных сетей. Основные свойства. Состав и назначения компонентов компьютерных сетей.	2	4
2.	Топологии сетей: звездообразная, кольцевая, шинная, древовидная, сотовая и полносвязная: принципы работы, области применения, достоинства и недостатки.	2	4
3.	Методы доступа к ресурсам компьютерной сети.	2	4
4.	Передающая среда: витая пара, коаксиальный кабель, волоконно-оптический кабель, радио и инфракрасный каналы.	2	4
5.	Логическое и физическое структурирование сетей. Маршрутизация и системы адресации компьютеров в ЛВС. Элементы промышленных сетей. Сетевые ПЛК.	2	6
6.	Основы администрирования и управления в компьютерных сетях.	2	4
7.	Методы обеспечения безопасности и сохранения данных.	2	4
8.	Мониторинг и анализ локальных сетей	2	4
	Всего	16	34

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Тема 1. Основные понятия и определения компьютерной сети.
Введение. Основные понятия и определения. Классификация и характеристики компьютерных сетей. Основные свойства. Состав и назначения компонентов компьютерных сетей.

Тема 2. Топологии сетей: звездообразная, кольцевая, шинная, древовидная, сотовая и полносвязная, принципы работы, области применения, достоинства и недостатки.

Тема 3. Методы доступа к ресурсам компьютерной сети.
Тема 4. Передающая среда: витая пара, коаксиальный кабель, волоконно-оптический кабель, радио и инфракрасный каналы.
Тема 5. Логическое и физическое структурирование сетей. Маршрутизация и системы адресации компьютеров в ЛВС. Элементы промышленных сетей. Сетевые ПЛК.
Тема 6. Основы администрирования и управления в компьютерных сетях.
Тема 7. Методы обеспечения безопасности и сохранения данных.
Тема 8. Мониторинг и анализ локальных сетей.

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Примерный перечень тем лабораторных занятий

1. Изучение программных средств тестирования параметров соединения в компьютерных сетях и проверки настройки протокола TCP/IP
2. Соединение ЭВМ в сеть
3. Маршрутизация в компьютерных сетях.
4. Разрешение адресов по протоколу ARP.
5. Динамическая маршрутизация по протоколу RIP. Получение сетевых настроек по DHCP.
6. Системы исчисления, применяемые в компьютерных сетях.
7. Способы адресации в компьютерных сетях.
8. Понятие масок подсетей.
9. Логическое моделирование сети на базе оборудования Cisco.
10. Объединение компьютерных сетей.
11. Служебные утилиты для работы в Интернет. Изучение протокола HTTP.
12. Настройка статической маршрутизации на оборудовании Cisco.
13. 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.

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.