МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
СТАРООСКОЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ
ИМ. А.А. УГАРОВА
(филиал) федерального государственного автономного образовательного учреждения
высшего образования
«Национальный исследовательский технологический университет «МИСиС»
ОСКОЛЬСКИЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ

А.В. Семенов

Компьютерные сети

Учебное пособие для студентов специальности
09.02.04 – Информационные системы (по отраслям)

Одобрено научно-методическим советом Оскольского политехнического колледжа
в качестве учебного пособия

Старый Оскол 2017

ББК

УДК

Рецензент:
преподаватель ОПК СТИ НИТУ «МИСиС» *Назарова О.И.*

Семенов А.В.
Компьютерные сети: учебное пособие. – Старый Оскол: СТИ НИТУ «МИСиС», 2017. – 129 с.

Данное учебное пособие составлено в соответствии с ФГОС3+ и уровню подготовки выпускников по специальности 09.02.04 – «Информационные системы» по ОП.03 Компьютерные сети. Целью пособия является практическое знакомство с различными подходами к описанию проектов компьютерных сетей, их моделированию и анализу с использованием современных средств проектирования.

Учебное издание предназначено для студентов системы СПО.

Содержание

ВВЕДЕНИЕ

Данное издание является частью учебно-методического комплекта по специальности 09.02.04 «Информационные системы (по отраслям)». Учебное пособие предназначено для изучения дисциплины «Компьютерные сети». Учебное пособие разработано на основании Федерального государственного образовательного стандарта среднего профессионального образования с учетом его профиля.

Компьютерные сети и сетевые технологии оказывают постоянно возрастающее влияние на все стороны нашей жизни. Их стремительное развитие требует широких и глубоких знаний, чему способствует введение дисциплины по компьютерным сетям в стандарты и учебные планы многих специальностей.

NetCracker – программный продукт, разработанный компанией NetCracker Technology, позволяет создавать проекты вычислительных сетей разной сложности/топологий и проводить их анализ, используя технологию имитационного моделирования.

Во время лабораторных занятий, описанных в данном учебном пособии, предполагается использование студентами программы NetCracker Professional.

Netcracker предлагает использовать простой и интуитивно понятный способ конструирования модели сети, основанный на применении готовых базовых блоков, соответствующих хорошо знакомым сетевым устройствам, таким как компьютеры, маршрутизаторы, коммутаторы, мультиплексоры и каналы связи.

Область применения пакета – создание проекта сетевого решения, тестирование этого решения и документирование окончательного варианта. База данных оборудования допускает, хотя и с некоторыми ограничениями, добавление нового оборудования с характеристиками, задаваемыми пользователем. Эта возможность, в частности, в достаточной мере компенсирует отсутствие оборудования Gigabit Ethernet, которое пользователь может создать самостоятельно.

Пользователь применяет технику drag-and-drop для графического изображения моделируемой сети из библиотечных элементов. Затем система Netcracker выполняет детальное моделирование полученной сети, отображая результаты динамически в виде наглядной мультипликации результирующего трафика. Другим вариантом задания топологии моделируемой сети является импорт топологической информации из систем управления и мониторинга сетей. После окончания моделирования пользователь получает в свое распоряжение следующие характеристики производительности сети:

- ✓ Прогнозируемые задержки между конечными и промежуточными узлами сети, пропускные способности каналов, коэффициенты использования сегментов, буферов и процессоров.
- ✓ Пики и спады трафика как функцию времени, а не как усредненные значения.
- ✓ Источники задержек и узких мест сети.

По опыту использования пакета точность анализа такова, что позволяет качественно оценивать возможность перегрузки оборудования и каналов передачи данных – находить «узкие места» сетевого проекта. Также необходимо учитывать, что требования пакета к производительности процессора растут по мере увеличения числа заданных потоков данных и на машинах, например, Celeron-500МГц симуляция проекта с числом потоков 15 уже может давать сбои, а для нормальной работы требует, по крайне мере, Celeron-800МГц.

Кроме того, пакет делает возможным познакомиться с практикой создания самых разнообразных сетевых решений почти «вживую» без дорогостоящей тестовой лаборатории.

Методические рекомендации по каждой практической работе имеют теоретическую часть, с необходимыми для выполнения работы пояснениями. Практические задания органично сочетаются с теоретическими знаниями.

В методическое пособие включены практические задания по разделам:

- ✓ Основные принципы построения компьютерных сетей;
- ✓ Протоколы. Пакетный принцип передачи данных;
- ✓ Глобальные сети. Технологии глобальных сетей;

Все практические занятия проводятся в компьютерной аудитории. На время выполнения практических занятий за студентом закрепляется персональный компьютер.

Практическая работа №1

Аппаратное и программное обеспечение сетей ЭВМ. Установка и первичная настройка сетевого ПО

Цель работы: Изучение состава аппаратного и программного обеспечения сетей ЭВМ. Получение практических навыков базовой настройки сетевой системы

В результате выполнения практической работы студент должен:

Знать:

- назначение, классификацию аппаратного и программного обеспечение сетей ЭВМ;
- коммуникационное оборудование, применяемое в сетевых технологиях.

Уметь:

- использовать некоторые сетевые прикладные программные пакеты для решения сетевых задач;
- производить базовые настройки сетевой системы.

Задание для практической работы:

1. Изучить теоретическую часть
2. Выполнить практическую часть
3. Ответить на контрольные вопросы
4. Оформить отчет.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Вычислительная сеть — это сложный комплекс взаимосвязанных и согласованно функционирующих программных и аппаратных компонентов. Изучение сети в целом предполагает знание принципов работы ее отдельных элементов:

- ✓ компьютеров;
- ✓ коммуникационного оборудования;
- ✓ операционных систем;
- ✓ сетевых приложений.

Оборудование сетей подразделяется на *активное* — интерфейсные карты компьютеров, повторители, концентраторы и т. п. и *пассивное* — кабели, соединительные разъемы, коммутационные панели и т. п. Кроме того, имеется вспомогательное оборудование - устройства бесперебойного питания, кондиционирования воздуха и аксессуары - монтажные стойки, шкафы, кабелепроводы различного вида. С точки зрения физики, активное оборудование — это устройства, которым необходима подача энергии для генерации сигналов, пассивное устройство подачи энергии не требует.

Компьютерной сетью называют совокупность узлов (компьютеров, терминалов, периферийных устройств), имеющих возможность информационного взаимодействия друг с другом с помощью специального коммуникационного оборудования и программного обеспечения.

Средства передачи и обработки информации ориентированы в ней на коллективное использование общесетевых ресурсов – информационных, программных, аппаратных.

Компьютерные сети могут работать в различных режимах: обмена данными между абонентами сети, запроса и выдачи информации, сбора информации пакетной обработки данных по запросам пользователей с удаленных терминалов, в диалоговых режимах.

Таким образом, с появлением сетей ЭВМ разрешены две очень важные проблемы:

- ✓ обеспечение в принципе неограниченного доступа к ЭВМ пользователей независимо от территориального расположения;
- ✓ возможность оперативного перемещений больших массивов информации на любые расстояния, позволяющий своевременно получать данные для принятия тех или иных решений.

Использование вычислительных сетей дает предприятию следующие возможности:

1. Разделение дорогостоящих ресурсов;
2. Улучшение доступа к информации;
3. Быстрое и качественное принятие решений;
4. Совершенствование коммуникаций;
5. Свобода в территориальном размещении компьютеров.

Программное обеспечение сетей ЭВМ в расширенном варианте составляют:

1. сетевые операционные системы;
2. сетевые драйвера, протоколы, службы и другое дополнительное программное обеспечение сетевых интерфейсов;
3. прикладное сетевое программное обеспечение.

Под *сетевыми операционными системами* понимают такие операционные системы, которые обеспечивают пользователям распределенный доступ к сетям ЭВМ.

Во вторую группу входит большой круг всевозможного программного обеспечения в основном изготовителя данного интерфейса (сетевой платы, модема и т.п.) для обеспечения правильной работы сетевого устройства.

При этом под *драйвером* понимается программа, непосредственно взаимодействующая с интерфейсом - сетевым адаптером и операционной системой (ОС). Драйвер сетевого адаптера взаимодействует с ОС через систему протоколов и служб, которые могут находится как в самих ОС, так и поставляться вместе с устройством.

При этом под *сетевым протоколом* понимается набор правил поведения сетевых узлов при передаче-приеме информации.

Под сетевыми службами понимается набор программного обеспечения сетевого обеспечения узкоспециального назначения, например:

- клиенты сетей - позволяют подключаться, обозревать и пользоваться сетевыми ресурсами соответствующих сетей,
- службы контроля трафика сетей,
- службы использования доступа к разделяемым ресурсам,
- доменные службы и др.

Круг *прикладного сетевого программного* обеспечения составляют всевозможные сетевые приложения.

Каждый компьютер работает под управлением собственной операционной системы. Взаимодействие между компьютерами сети происходит за счет передачи сообщений через сетевые адаптеры и каналы связи. С помощью этих сообщений один компьютер обычно запрашивает доступ к *локальным ресурсам* другого компьютера. Такими ресурсами могут быть как данные, хранящиеся на диске, так и разнообразные периферийные устройства - принтеры, модемы, факс-аппараты и т.д. Разделение локальных ресурсов каждого компьютера между всеми пользователями сети - основная цель создания вычислительной сети.

Каким же образом сказывается на пользователе тот факт, что его компьютер подключен к сети? Прежде всего, он может пользоваться не только файлами, дисками, принтерами и другими ресурсами своего компьютера, но и аналогичными ресурсами других компьютеров, подключенных к той же сети. Правда, для этого недостаточно снабдить компьютеры сетевыми адаптерами и соединить их кабельной системой. Необходимы еще некоторые добавления к операционным системам этих компьютеров. На тех компьютерах, ресурсы которых должны быть доступны всем пользователям сети, необходимо добавить модули, которые постоянно будут находиться в режиме ожидания запросов, поступающих по сети от других компьютеров. Обычно такие модули называются программными *серверами (server),* так как их главная задача - обслуживать (serve) запросы на доступ к ресурсам своего компьютера. На компьютерах, пользователи которых хотят получать доступ к ресурсам других компьютеров, также нужно добавить к операционной системе некоторые специальные программные модули, которые должны вырабатывать запросы на доступ к удаленным ресурсам и передавать их по сети на нужный компьютер. Такие модули обычно называют программными *клиентами (client).* Собственно, сетевые адаптеры и каналы связи решают в сети достаточно простую задачу — они передают сообщения с запросами и ответами от одного компьютера к другому, а основную работу по организации совместного использования ресурсов выполняют клиентские и серверные части операционных систем.

Пара модулей «клиент – сервер» обеспечивает совместный доступ пользователей к определенному типу ресурсов, например, к файлам. В этом случае говорит, что пользователь имеет дело с файловой *службой (service).* Обычно сетевая операционная система поддерживает несколько видов сетевых служб для своих пользователей - файловую службу, службу печати, службу электронной почты, службу удаленного доступа и т. п.

Термины «клиент» и «сервер» используются не только для обозначения программных модулей, но и компьютеров, подключенных к сети. Если компьютер предоставляет свои ресурсы другим компьютерам сети, то он называется сервером, а если он их потребляет - клиентом. Иногда один и тот же компьютер может одновременно играть роли и сервера, и клиента.

Сетевые службы всегда представляют собой *распределенные программы,* состоящие из нескольких взаимодействующих частей, причем каждая часть, как правило, выполняется на отдельном компьютере сети.

До сих пор речь шла о системных распределенных программах. Однако в сети могут выполняться и распределенные пользовательские программы - приложения. Распределенное приложение также состоит из нескольких частей, каждая из которых выполняет какую-то определенную законченную работу по решению прикладной задачи. Например, одна часть приложения, выполняющаяся на компьютере пользователя, может поддерживать специализированный графический интерфейс, вторая - работать на мощном выделенном компьютере и заниматься статистической обработкой введенных пользователем данных, а третья - заносить полученные результаты в базу данных на компьютере с установленной стандартной СУБД. Распределенные приложения в полной мере используют потенциальные возможности распределенной обработки, предоставляемые вычислительной сетью, и поэтому часто называются *сетевыми приложениями.*

Виртуальная машина - программная и/или аппаратная система, эмулирующая аппаратное обеспечение некоторой платформы и исполняющая программы для target-

платформы на host-платформе. Наиболее распространенные специализированные программы от известных производителей:

- Microsoft Virtual PC
- Oracle VM VirtualBox
- VMWare Workstation

При помощи данного программного обеспечения можно установить любую систему на любую платформу (Windows, Mac OS, Linux и др.). При этом получим полнофункциональную систему с доступом в локальную сеть и интернет, под которую можно даже не выделять самостоятельного раздела. Таких виртуальных компьютеров, при необходимости, можно установить десятки.

Виртуальная машина – это компьютер в компьютере, имеющий свой жесткий диск, процессор, выделенную оперативную память, графический адаптер и т.д. Всеми этими ресурсами делится с ней физическая машина, то есть компьютер, который стоит дома или в офисе. После настройки и создания виртуальной машины на нее устанавливается виртуальная операционная система, абсолютно ничем не отличающаяся от реальной. Возможности виртуальной машины ограничиваются только возможностями персонального компьютера.

Все виртуальные машины после их создания могут работать либо в оконном режиме, либо в полноэкранном режиме.

На 64-разрядную систему можно устанавливать 32 и 64-разрядные гостевые системы, в то время как на хост с 32-разрядной системой можно будет установить только 32-разрядную гостевую систему.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Описание сетевого оборудования, применяемого к учебной аудитории для поддержания учебного процесса

Например, витая пара (TP - Twisted Pair) – это кабель, выполненный в виде скрученной пары проводов (рис. 1). Он может быть экранированным и неэкранированным. Экранированный кабель более устойчив к электромагнитным помехам. Витая пара наилучшим образом подходит для малых учреждений. Недостатками данного кабеля является высокий коэффициент затухания сигнала и высокая чувствительность к электромагнитным помехам, поэтому максимальное расстояние между активными устройствами в ЛВС при использовании витой пары должно быть не более 100 метров.

Рис. 1.1 «Кабель на основе витой пары»

Задание на практическую работу

- ✓ Познакомиться с основным сетевым оборудованием, применяемым к учебной аудитории для поддержания учебного процесса.
- ✓ Охарактеризовать назначение, маркировку, функции и параметры следующего коммуникационного оборудования согласно вариантам:

ВАРИАНТЫ ЗАДАНИЙ

№ вар.	Наименование оборудования
1	Повторитель
2	Концентратор
3	Коммутатор
4	Кабельная система «Витая пара»
5	Оптоволоконный кабель
6	Маршрутизатор
7	Брандмауэр
8	Сетевая плата
9	Модем
10	Мост

- ✓ Охарактеризовать сетевые операционные системы согласно вариантам по следующей схеме:

1) платность,
2) доступ к исходному коду,
3) многоплатформенность,
4) мультизадачность,
5) количество пользователей,
6) функции управления сетью,
7) интерфейс работы,
8) потребляемые ресурсы.

№ вар.	Наименование операционной системы
1	MS-DOS
2	Microsoft Windows XP
3	Microsoft Windows 7
4	Microsoft Windows 8.1
5	Microsoft Windows Server 2003

- ✓ Вам предоставлены 2 виртуальные машины. В одной установлен Windows XP/7, в другой – MS-DOS.

1. Конфигурация виртуальной машины с установленным Windows XP/7.

- NetBIOS-имя машины – WINDOWS;
- установленные сетевые протоколы – NetBEUI;
- предоставлен в совместное использование дисковый ресурс, сетевое имя – PUBLIC;
- имеется учетная запись пользователя student, пароль пустой.

Данную виртуальную машину необходимо запустить в начале выполнения лабораторной работы, по окончании – завершить. Изменение ее конфигурации не требуется.

2. Конфигурация виртуальной машины с установленной MS-DOS.

- виртуальная машина имеет сетевую плату AMD PCNet PCI Adapter (Plug&Play, драйвер сетевой платы автоматически определяет аппаратные параметры NIC, такие как IRQ, BASEIO и DMA; не следует указывать данные параметры в конфигурационных файлах);
- установленная операционная система – MS DOS;
- каталог C:\Drivers содержит NDIS-драйвер сетевой платы AMD PCNet PCI Adapter:
    - *.dos – файл драйвера
    - *.ini – значения некоторых параметров секции драйвера сетевой платы в файле настроек сетевого клиента (регистрируемое в сетевой подсистеме имя драйвера);
- каталог MSClient содержит инсталляционный пакет Microsoft Network Client.

Требуется установить и настроить сетевой клиент, выполнив следующие условия:

- имя компьютера должно быть уникально;
- из сетевых протоколов должен быть установлен только NetBEUI;
- из сетевых карт должна быть установлена только AMD PCNet PCI Adapter;
- не должно быть сохраненных паролей пользователей.

Также, требуется скопировать на локальный диск файл, расположенный на общем ресурсе \\WINNT\PUBLIC.

Дополнительная информация

- Для перезагрузки DOS следует использовать сочетание клавиш Ctrl+Atl+Del
- Для проверки доступности сетевых ресурсов можно использовать команду net.exe
    - net view – выводит список доступных серверов
    - net view \\SERVERNAME – выводит список ресурсов на сервере SERVERNAME
- Для подключения сетевого диска можно использовать команду вида
  net use z: \\SERVERNAME\RESOURCENAME
- Для отключения сетевого диска можно использовать команду вида
  net use z: /delete

3. Подготовка виртуальной машины:

3.1. Установить операционную систему – какую-либо DOS (можно извлечь из старых версий Windows, либо скачать бесплатную DOS)

3.2. На диск C: скопировать инсталляционный пакет MS Network Client (обычно присутствует на инсталляционных дисках серверных версий Windows).

3.3. На диск C: скопировать драйвер для AMD PCNet PCI Adapter (pcntnd.dos, protocol.ini – можно найти в Интернет).

Виртуальная машина 2

Аппаратная конфигурация (примерная):

- Оперативная память – 64 или более Мб
- Жесткий диск – 1 или более Гб
- Сетевой адаптер – подключен к виртуальной сети VMnet2

4. Подготовка виртуальной машины

4.1. Установить операционную систему – какую-либо Windows.

- задать имя машины WINDOWS;
- задать имя рабочей группы WORKGROUP;
- установить драйвер NIC AMD PCNet PCI Adapter;
- установить сетевой протокол NetBEUI.

4.2. Создать в Windows пользователя student с пустым паролем.

4.3. Создать каталог C:\Public и предоставить его в совместное пользование с именем PUBLIC. Разрешить чтение сетевого ресурса для всех пользователей. Скопировать в каталог какой-либо файл.

Контрольные вопросы:

1. Что такое компьютерная сеть?
2. Что входит в аппаратное обеспечение сетей?
3. Функции и характеристики коммуникационного оборудования?
4. Что такое активное оборудование сетей?
5. Что такое пассивное оборудование сетей?
6. Что такое вспомогательное оборудование сетей?
7. Что называют операционной системой?
8. Что входит в группу прикладного программного обеспечения?
9. По каким критериям можно охарактеризовать сетевую операционную систему?
10. Что такое технология «клиент-сервер»?
11. Что такое виртуальная машина? Ее назначение?

Практическая работа № 2

Сетевое обеспечение под управлением оперативное системы Windows 7

Цель работы: экспериментальное исследование сетевого конфигурирования в операционной системе Windows 7

В результате выполнения практических заданий обучающийся должен:

Знать:

- ✓ основные сетевые компоненты операционной системы, необходимые для подключения компьютера к локальной или внешней сети.

Уметь:

- ✓ работать с компонентами «Центр управления сетями и общим доступом», «Сетевое расположение», «Карта сети», уметь подключать компьютер к локальной или внешней сети.

Задание для практической работы:

1. Изучить теоретическую часть
2. Выполнить практическую часть
3. Ответить на контрольные вопросы
4. Оформить отчет.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Зачастую, настройка локальной сети в операционных системах Windows Vista, Windows 7, Windows Server 2008/2008 R2 начинается с такой области конфигурирования сетевых свойств, как компонент «Центр управления сетями и общим доступом». При помощи данного средства конфигурирования сетей можно выбирать сетевое размещение, просматривать карту сети, настраивать сетевое обнаружение, общий доступ к файлам и принтерам, а также настраивать и просматривать состояние ваших текущих сетевых подключений.

Открытие компонента «Центр управления сетями и общим доступом»

Для того чтобы воспользоваться функционалом средства конфигурирования сетей, нужно для начала его открыть. Чтобы открыть окно «Центр управления сетями и общим доступом», выполните одно из следующих действий:

- В области уведомлений нажмите правой кнопкой мыши на значке «Сеть» и из контекстного меню выберите команду «Центр управления сетями и общим доступом»;
- Нажмите на кнопку «Пуск» для открытия меню, выделите элемент «Сеть» и нажмите на нем правой кнопкой мыши. Из контекстного меню выберите команду «Свойства»;
- Нажмите на кнопку «Пуск» для открытия меню, откройте «Панель управления», из списка компонентов панели управления выберите категорию «Сеть и Интернет», а затем перейдите по ссылке «Центр управления сетями и общим доступом»;
- Нажмите на кнопку «Пуск» для открытия меню, в поле поиска введите Центр управления и в найденных результатах откройте приложение «Центр управления сетями и общим доступом»;
- Воспользуйтесь комбинацией клавиш +R для открытия диалога «Выполнить». В диалоговом окне «Выполнить», в поле «Открыть» введите

%windir%\system32\control.exe/name Microsoft.NetworkAndSharingCenter и нажмите на кнопку «ОК».

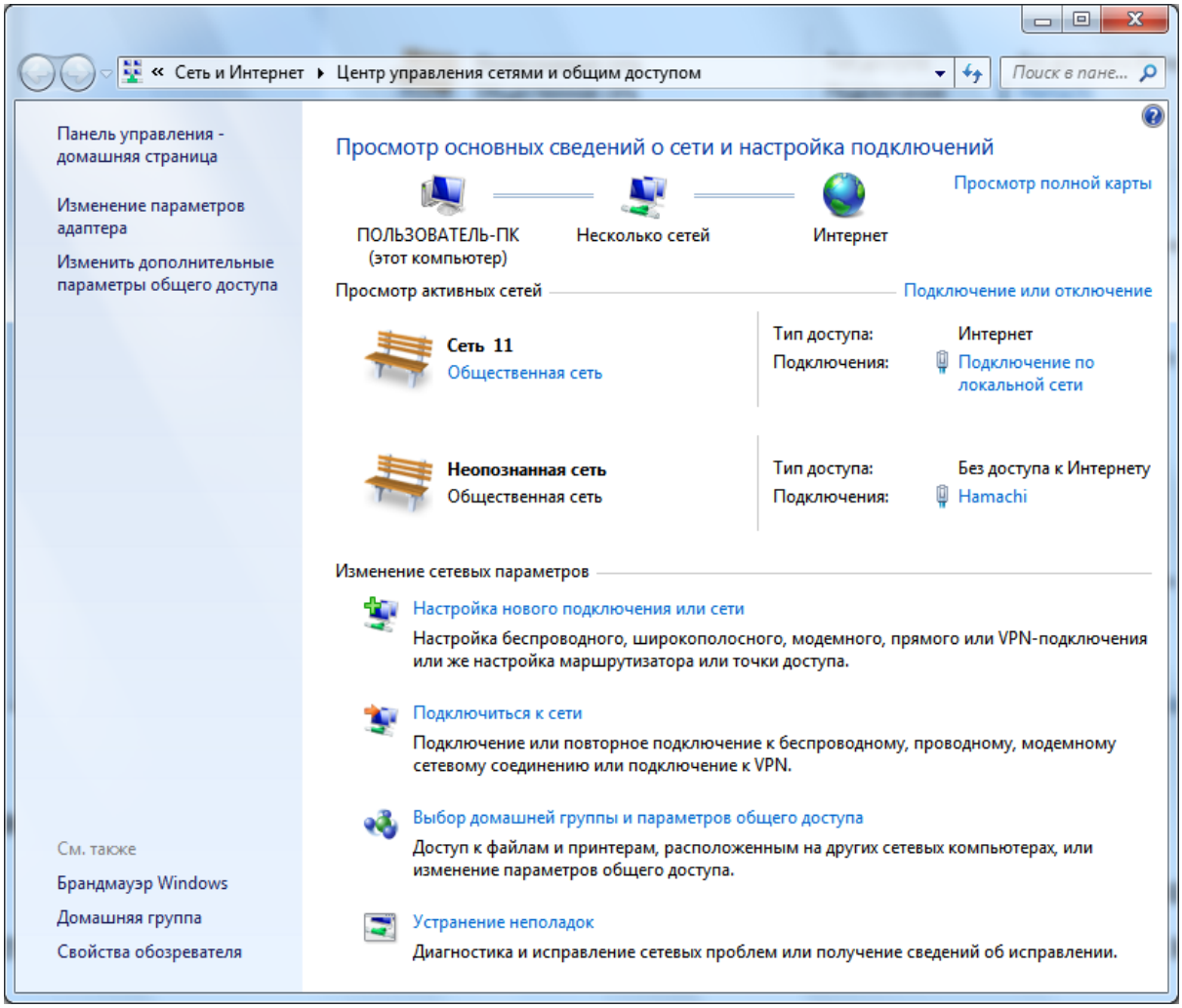


Рис. 2.1 «Центр управления сетями и общим доступом»

Понятие сетевого расположения

Перед началом работы с данным компонентом, следует разобраться с таким понятием как сетевое расположение. Этот параметр задается для компьютеров при первом подключении к сети и во время подключения автоматически настраивается брандмауэр и параметры безопасности для того типа сети, к которому производится подключение. В отличие от операционной системы Windows Vista, где для всех сетевых подключений используется самый строгий профиль брандмауэра для сетевого размещения, операционная система Windows 7 поддерживает несколько активных профилей, что позволяет наиболее безопасно использовать несколько сетевых адаптеров, подключенных к различным сетям. Существует четыре типа сетевого расположения (рис.2.2).

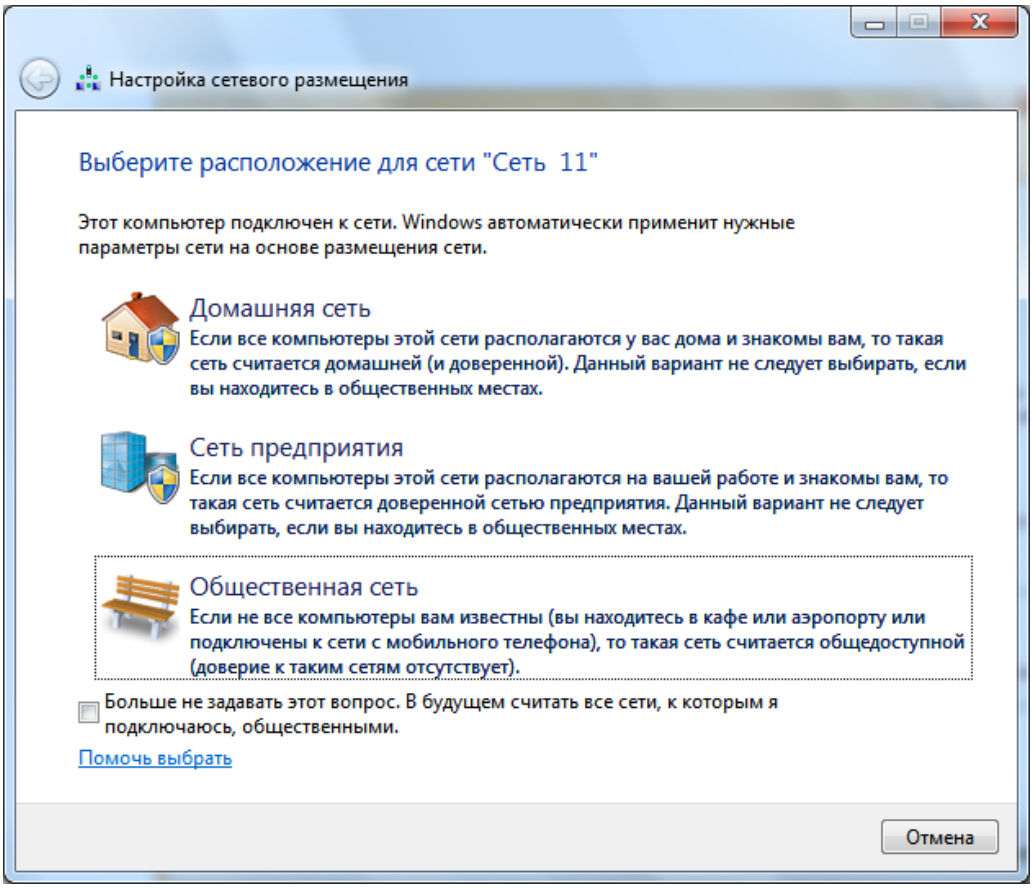


Рис. 2.2 «Выбор сетевого расположения»

Домашняя сеть. Данное сетевое расположение предназначено для использования компьютера в домашних условиях или в таких сетях, где пользователи очень хорошо знают друг друга. Такие компьютеры могут создавать и присоединяться к домашним группам. Для домашних сетей автоматически включается обнаружение сети.

Сеть предприятия. Такое сетевое расположение используется в сети малого офиса (SOHO). Для этого сетевого расположения также включено обнаружение сети, но вы не можете ни создавать, ни присоединять компьютер к домашней группе.

Общественная сеть. Это сетевое расположение предназначено для использования компьютера в таких общественных местах, как кафе или аэропорты. Это наиболее строгое размещение, у которого по умолчанию отключены возможности присоединения к домашней группе и сетевое обнаружение.

Доменная сеть. Если компьютер присоединён к домену Active Directory, то существующей сети будет автоматически назначен тип сетевого размещения «Домен». Доменный тип сетевого расположения аналогичен рабочей сети, за исключением того, что в домене конфигурация брандмауэра Windows, сетевого обнаружения, а также сетевой карты определяется групповой политикой.

Каким образом связаны компьютеры в сети, можно просматривать с помощью карты сети. Однако этот компонент доступен не для всех типов сетевого расположения**.**

Карта сети

Карта сети – это графическое представление расположения компьютеров и устройств, которое позволяет увидеть все устройства вашей локальной сети, а также схему их подключения друг к другу. В окне «Центр управления сетями и общим доступом» отображается только локальная часть сетевой карты, компоновка которой зависит от

имеющихся сетевых подключений. Компьютер, на котором выполняется создание карты, отображается в левом верхнем углу. Другие компьютеры подсети отображаются слева. Такие устройства инфраструктуры, как коммутаторы, концентраторы и шлюзы в другие сети отображаются справа. Сетевое сопоставление работает в проводных и беспроводных сетях, однако, только в частных и доменных сетях. Просмотреть карту публичной сети невозможно. Протокол LLTD обеспечивает сопоставление только компьютеров в одной подсети, которая является обычной установкой в домашних или малых офисах.

Можно заметить, что некоторые компьютеры и устройства отображаются отдельно в нижней части окна «Карта сети» либо могут вообще отсутствовать. Например, если сервер печати беспроводной сети поддерживает технологию UPnP, а не LLTD, то он будет располагаться в нижней части окна «Карта сети». Подобная ситуация возникает, поскольку не все операционные системы и устройства предполагают поддержку протокола LLTD или вследствие возможной неправильной настройки устройств. Пример карты сети вы можете увидеть на рис.2.3.

Рис. 2.3 «Пример карты сети»

За работу карты сети в операционных системах отвечают два компонента:

- ✓ Обнаружение топологии связи Link Layer (Link Layer Topology Discover Mapper – LLTD Mapper) – компонент, который запрашивает в сети устройства для включения их в карту;
- ✓ Отвечающее устройство LLTD (Link Layer Topology Discover Responder – LLTD Responder) – компонент, который отвечает за запросы компонента LLTD Mapper.

По умолчанию, карту сети можно просматривать только для расположений «Домашняя сеть» или «Сеть предприятия». При попытке просмотра сетевой карты для расположений «Доменная сеть» или «Общественная сеть» вы увидите сообщение о невозможности отображения карты.

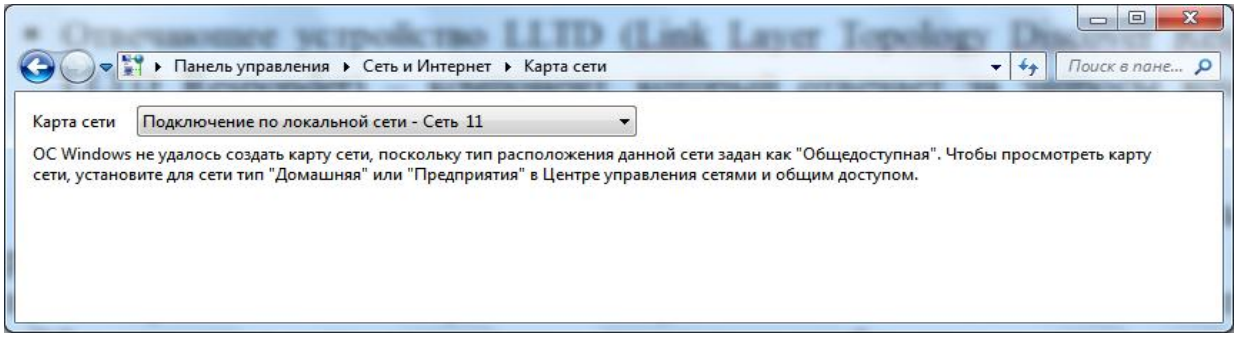


Рис. 2.4 «Попытка просмотра карты сети»

Для того чтобы включить сетевое сопоставление в доменной сети, вам нужно на контроллере домена выполнить следующие действия:

1. Откройте оснастку «Управление групповой политики»;
2. Выберите объект групповой политики (например, Default Domain Policy, область действия – весь домен), который будет распространяться на компьютер, расположенный в доменной сети, нажмите на нем правой кнопкой мыши и из контекстного меню выберите команду «Изменить»;
3. В оснастке «Редактор управления групповыми политиками» разверните узел Конфигурация компьютера/Политики/Административные шаблоны/Сеть/Обнаружение топологии связи (Link Layer) и выберите политику «Включает драйвер отображения ввода/вывода (LLTDIO)»;
4. В свойствах параметра политики установите переключатель на опцию «Включить» и установите флажок «Разрешить операцию для домена»;
5. Повторите аналогичные действия для параметра политики «Включить драйвер «Ответчика» (RSPNDR)»;
6. Обновите параметры политики на клиентской машине, используя команду gpupdate /force /boot;
7. Обновите карту сети.

Сетевые подключения

После установки драйвера для каждого сетевого адаптера, операционная система Windows пытается автоматически сконфигурировать сетевые подключения на локальном компьютере. Все доступные сетевые подключения отображаются в окне «Сетевые подключения». Сетевое подключение представляет собой набор данных, необходимых для подключения компьютера к Интернету, локальной сети или любому другому компьютеру.

Открыть окно «Сетевые подключения» вы можете любым из следующих способов:

- ✓ Откройте окно «Центр управления сетями и общим доступом» и перейдите по ссылке «Изменение параметров адаптера»;
- ✓ Нажмите на кнопку «Пуск» для открытия меню, в поле поиска введите Просмотр сетевых и в найденных результатах откройте приложение «Просмотр сетевых подключений»;
- ✓ Воспользуйтесь комбинацией клавиш +R для открытия диалога «Выполнить». В диалоговом окне «Выполнить», в поле «Открыть» введите ncpa.cpl или control netconnection и нажмите на кнопку «ОК».

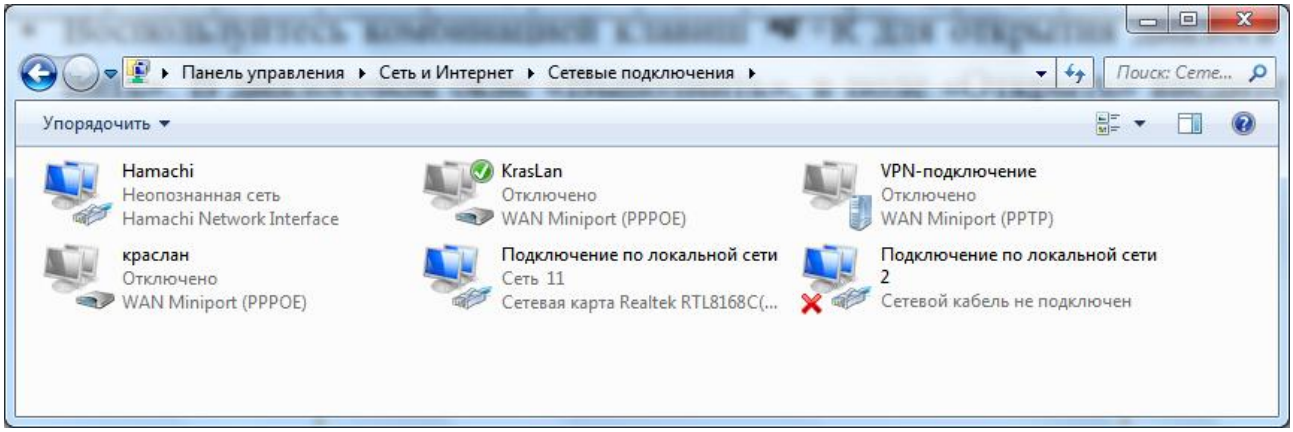


Рис. 2.5 Окно «Сетевые подключения»

При выборе любого сетевого подключения вы можете выполнить с ним следующие действия:

Переименование подключения. Операционная система по умолчанию назначает всем сетевым подключениям имена «Подключение по локальной сети» или «Подключение к беспроводной сети» и номер подключения в том случае, если у вас существует более одного сетевого подключения. При желании, вы можете переименовать любое сетевое подключение одним из трех следующих способов:

- Нажмите на клавишу F2, введите новое имя сетевого подключения, после чего нажмите на клавишу Enter;
- Нажмите правой кнопкой мыши на переименовываемом сетевом подключении и из контекстного меню выберите команду «Переименовать». Введите новое имя сетевого подключения, после чего нажмите на клавишу Enter;
- Выберите сетевое подключение и нажмите на кнопку «Переименование подключения», которая расположена на панели инструментов. После чего введите новое имя сетевого подключения и нажмите на клавишу Enter.

Состояние сети. Используя данное окно, вы можете просмотреть любые данные о состоянии сетевого подключения и такие детали, как IP-адрес, MAC-адрес и прочее. Чтобы открыть диалоговое окно сведений о сетевом подключении, выполните следующие действия:

1. Откройте диалоговое окно «Состояние» одним из следующих способов:

- Нажмите правой кнопкой мыши на сетевом подключении и из контекстного меню выберите команду «Состояние»;
- Выберите сетевое подключение и нажмите на кнопку «Просмотр состояния подключения», которая расположена на панели инструментов;
- Выберите сетевое подключение и нажмите на клавишу Enter.

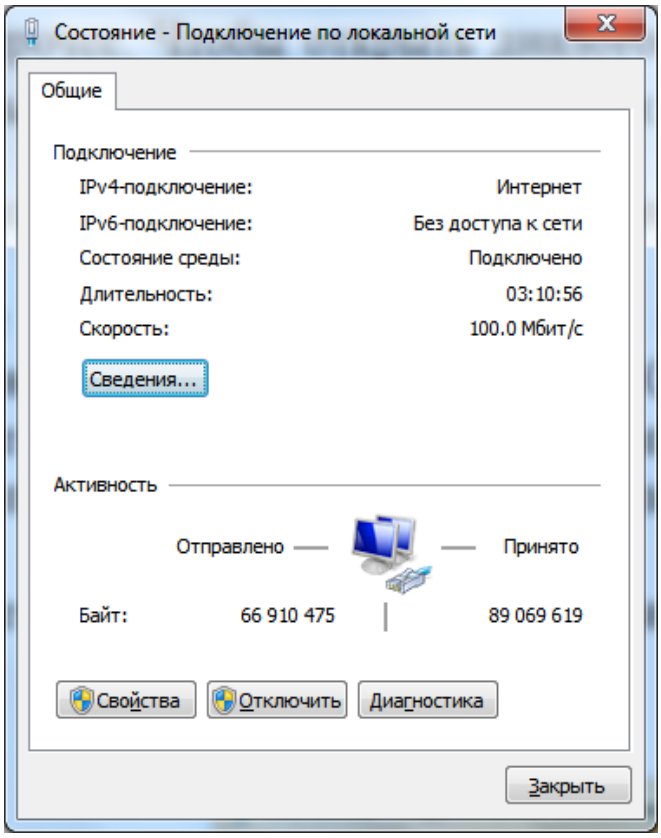


Рис. 2.6 «Диалоговое окно состояния подключения по локальной сети»

2. В окне «Состояние – подключение по локальной сети» нажмите на кнопку «Сведения». В диалоговом окне «Сведения о сетевом подключении», отображенном ниже, вы можете просмотреть подробные сведения о текущем сетевом подключении.

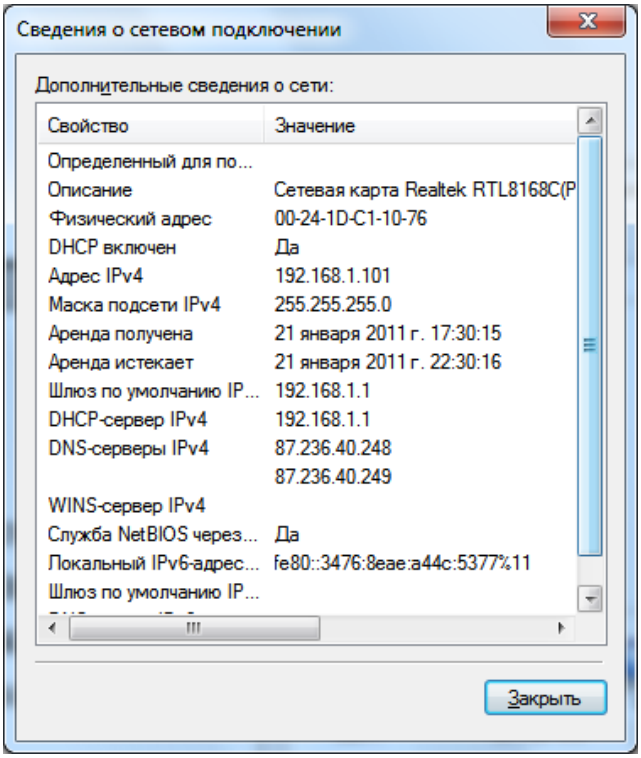


Рис. 2.7. Сведения о сетевом подключении

Диагностика подключения. В случае обнаружения проблем в работе вашего сетевого подключения, окно «Сетевые подключения» предлагает средство диагностики «Устранение неполадок», которое содержит возможность решения при помощи анализа подключения. Для того чтобы воспользоваться данным средством выполните любое из следующих действий:

Нажмите правой кнопкой мыши на сетевом подключении и из контекстного меню выберите команду «Диагностика».

- Выберите сетевое подключение и нажмите на кнопку «Диагностика подключения», которая расположена на панели инструментов.

В открывшемся диалоговом окне «Диагностика сетей Windows» для устранения неполадок следуйте действиям мастера.

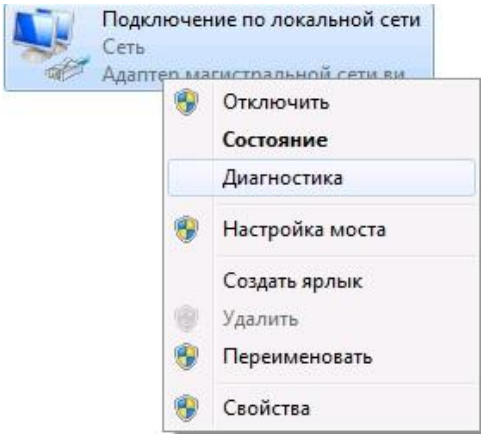


Рис. 2.8 «Открытие мастера устранения неполадок подключения по локальной сети»

Отключение сетевого устройства. Иногда проблемы с сетевыми подключениями решаются посредством отключения сетевого адаптера компьютера от сети. Для того чтобы отключить сетевой адаптер выполните одно из следующих действий:

- Нажмите правой кнопкой мыши на сетевом подключении и из контекстного меню выберите команду «Отключить»;
- Выберите сетевое подключение и нажмите на кнопку «Отключение сетевого устройства», которая расположена на панели инструментов.

Настройка параметров подключения. Как таковые, сетевые подключения не позволяют осуществлять коммуникации. Осуществление коммуникаций обеспечивают сетевые клиенты, службы и протоколы, которые привязаны к созданным сетевым подключениям (рис.2.9).

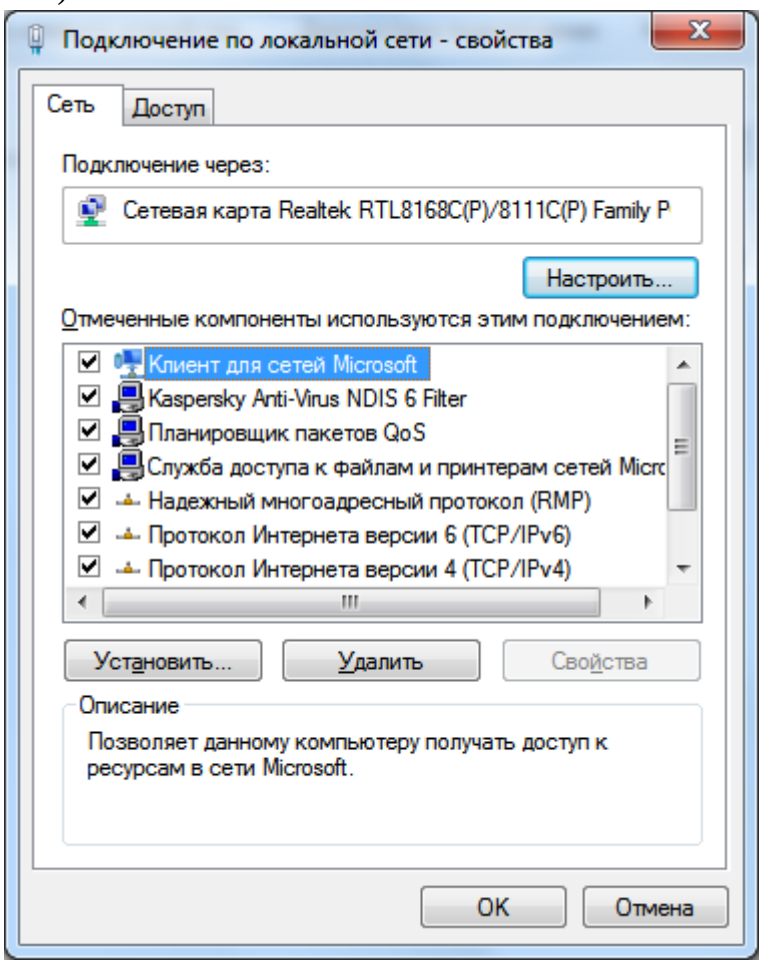


Рис. 2.9 «Диалоговое окно свойств сетевого подключения»

Для того чтобы изменить настройки вашего сетевого подключения, вы можете воспользоваться средствами настройки параметров подключения. Для изменения компонентов и настроек сетевого подключения, выполните следующие действия:

- Нажмите правой кнопкой мыши на сетевом подключении и из контекстного меню выберите команду «Свойства»;
- Выберите сетевое подключение и нажмите на кнопку «Настройка параметров подключения», которая расположена на панели инструментов;
- Выберите сетевое подключение и воспользуйтесь комбинацией клавиш Alt + Enter.

Установленные возле компонентов флажки указывают, что эти компоненты привязаны к подключению.

ПРАКТИЧЕСКАЯ ЧАСТЬ

1. Выполнить все этапы настройки сети в операционной системе Windows 7, указанные выше.

Контрольные вопросы

1. Какое назначение у компонента «Центр управления сетями и общим доступом»?
2. Продемонстрируйте, какие существуют способы открытия компонента «Центр управления сетями и общим доступом».
3. Охарактеризуйте типы сетевого расположения.
4. Что представляет собой карта сети, по сути, и по виду?
5. Какие протоколы отвечают за построение карты сети.
6. В каких ситуациях просмотр карты сети будет не возможен?
7. С каким аппаратным сетевым компонентом связываются свойства доступных сетевых подключений?
8. Какие существуют способы открытия окна «Сетевые подключения»?
9. Какие действия пользователь обычно может выполнить по отношению к любому сетевому подключению?
10. Охарактеризуйте свойства сведений о сетевом подключении для компьютера, на котором вы выполняете лабораторную работу. Объясните значения данных свойств.
11. Какими способами можно вызвать средства диагностики подключения?
12. Охарактеризуйте компоненты подключения к сети компьютера, за которым вы выполняете лабораторную работу.

Практическая работа №3

Семейство протоколов TCP/IP. Использование утилит стека протоколов

Цель работы: познакомиться со средствами диагностики сети и поиска неисправностей стека TCP/IP.

В результате выполнения практических заданий обучающийся должен:

Знать:

- ✓ принципы работы протоколов TCP/IP;
- ✓ основные утилиты для исследования сети и поиска неисправностей в настройке. **TCP/IP.**

Уметь:

- ✓ применять диагностические утилиты для исследования сети и поиска неисправностей в настройке TCP/IP.

Задание для практической работы:

1. Изучить теоретическую часть
2. Выполнить практическую часть
3. Ответить на контрольные вопросы
4. Оформить отчет.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Диагностические утилиты TCP/IP.

Стек TCP/IP предоставляет пользователям две основные службы, которые используют прикладные программы:

1. Дейтаграммное средство доставки пакетов. Это означает, что протоколы стека TCP/IP определяют маршрут передачи небольшого сообщения, основываясь только на адресной информации, находящейся в этом сообщении. Доставка осуществляется без установки логического соединения. Такой тип доставки делает протоколы TCP/IP адаптируемыми к широкому диапазону сетевого оборудования;

2. Надежное потоковое транспортное средство. Большинство приложений требуют, от коммуникационного программного обеспечения автоматического восстановления при ошибках передачи, потери пакетов или сбоях в промежуточных маршрутизаторах. Надежное транспортное средство позволяет устанавливать логическое соединение между приложениями, а затем посылать большие объемы данных по этому соединению.

В его структуру входят протоколы IP, ARP, ICMP, TCP, UDP, TELNET, FTP, HTTP, SDH и другие.

Целью устранения неисправностей в настройке TCP/IP является восстановление нормальной работы сети. Для поиска неисправностей можно использовать специальные диагностические утилиты, предназначенные для проверки конфигурации стека TCP/IP и тестирования сетевого соединения. Список некоторых утилит приведен в таблице 3.1.

Таблица 3.1 «Диагностические утилиты TCP/IP»

Утилита	Применение
Arp	Выводит для просмотра и изменения таблицу трансляции адресов, используемую протоколом разрешения адресов ARP (Address Resolution Protocol - определяет локальный адрес по IP-адресу).
Hostname	Выводит имя локального хоста. Используется без параметров.
Ipconfig	Выводит значения для текущей конфигурации стека TCP/IP: IP-адрес,

	маску подсети, адрес шлюза по умолчанию, адреса WINS (Windows Internet Naming Service) и DNS (Domain Name System)
Nbtstat	Выводит статистику и текущую информацию по NetBIOS, установленному поверх TCP/IP. Используется для проверки состояния текущих соединений NetBIOS.
Netstat	Выводит статистику и текущую информацию по соединению TCP/IP.
Nslookup	Осуществляет проверку записей и доменных псевдонимов хостов, доменных сервисов хостов, а также информации операционной системы, путем запросов к серверам DNS.
Ping	Осуществляет проверку правильности конфигурирования TCP/IP и проверку связи с удаленным хостом.
Route	Модифицирует таблицы маршрутизации IP. Отображает содержимое таблицы, добавляет и удаляет маршруты IP.
Tracert	Осуществляет проверку маршрута к удаленному компьютеру путем отправки эхо-пакетов протокола ICMP (Internet Control Message Protocol). Выводит маршрут прохождения пакетов на удаленный компьютер.

Проверка правильности конфигурации TCP/IP.

При устранении неисправностей и проблем в сети TCP/IP следует сначала проверить правильность конфигурации TCP/IP. Для этого используется утилита ipconfig.

Эта команда полезна на компьютерах, работающих с DHCP (Dynamic Host Configuration Protocol), так как дает пользователям возможность определить, какая конфигурация сети TCP/IP и какие величины были установлены с помощью DHCP.

Синтаксис:

ipconfig [/all | /renew[adapter] | /release]

Параметры:

all - выдает весь список параметров. Без этого ключа отображается только IP-адрес, маска и шлюз по умолчанию;

renew[adapter] - обновляет параметры конфигурации DHCP для указанного сетевого адаптера;

release[adapter] - освобождает выделенный DHCP IP-адрес;

adapter – имя сетевого адаптера;

displaydns - выводит информацию о содержимом локального кэша клиента DNS, используемого для разрешения доменных имен.

Таким образом, утилита ipconfig позволяет выяснить, инициализирована ли конфигурация и не дублируются ли IP-адреса:

- если конфигурация инициализирована, то появляется IP-адрес, маска, шлюз;
- если IP-адреса дублируются, то маска сети будет 0.0.0.0;
- если при использовании DHCP компьютер не смог получить IP-адрес, то он будет равен 0.0.0.0 .

Тестирование связи с использованием утилиты ping.

Утилита ping (Packet Internet Grouper) используется для проверки конфигурирования TCP/IP и диагностики ошибок соединения. Она определяет доступность и функционирование указанного узла и позволяет измерить время прохождения пакетов от данного узла до любого другого узла сети. Использование ping лучший способ проверки того, что между локальным компьютером и сетевым хостом существует маршрут. (Хостом

называется любое сетевое устройство (компьютер, маршрутизатор), обменивающееся информацией с другими сетевыми устройствами по TCP/IP.)

Команда ping проверяет соединение с удаленным хостом, посылая к этому хосту несколько IP-пакетов и ожидая ответы на них. При этом она измеряет интервал времени, в течение которого пакет вернулся, а также показывает соотношение количества отосланных пакетов к количеству принятых, то может служить субъективной оценкой «качества связи» между узлами. Если связь между хостами плохая, из сообщений ping станет ясно, сколько пакетов потеряно.

Утилита использует протокол ICMP. Посылаемые и получаемые IP-пакеты – это эхо-запросы и эхо-ответы протокола ICMP.

По умолчанию передается 4 эхо-пакета длиной 32 байта (периодическая последовательность символов алфавита в верхнем регистре). Ping позволяет изменить размер и количество пакетов, указать, следует ли записывать маршрут, который она использует, какую величину времени жизни (ttl) устанавливать, можно ли фрагментировать пакет и т.д.. При получении ответа в поле time указывается, за какое время (в миллисекундах) посланный пакет доходит до удаленного хоста и возвращается назад. Так как значение по умолчанию для ожидания отклика равно 1 секунде, то все значения данного поля будут меньше 1000 миллисекунд. Если вы получаете сообщение «Request time out» (Превышен интервал ожидания), то, возможно, если увеличить время ожидания отклика, пакет дойдет до удаленного хоста. Это можно сделать с помощью ключа –w.

Ping можно использовать для тестирования как имени хоста (DNS или NetBIOS), так и его IP-адреса. Если ping с IP-адресом выполнилась успешно, а с именем – неудачно, это значит, что проблема заключается в распознавании соответствия адреса и имени, а не в сетевом соединении.

Утилита ping используется следующими способами:

1) Для проверки того, что TCP/IP установлен и правильно сконфигурирован на локальном компьютере, в команде ping задается адрес петли обратной связи (loopback address): ping 127.0.0.1

Если тест успешно пройден, то вы получите следующий ответ:

```
C:\Windows\system32>ping 127.0.0.1

Обмен пакетами с 127.0.0.1 по с 32 байтами данных:
Ответ от 127.0.0.1: число байт=32 время<1мс TTL=128
Ответ от 127.0.0.1: число байт=32 время<1мс TTL=128
Ответ от 127.0.0.1: число байт=32 время<1мс TTL=128
Ответ от 127.0.0.1: число байт=32 время<1мс TTL=128

Статистика Ping для 127.0.0.1:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 0мсек, Максимальное = 0 мсек, Среднее = 0 мсек

C:\Windows\system32>
```

Рис. 3.1 «Результат работы утилиты ping»

2) Чтобы убедиться в том, что компьютер правильно добавлен в сеть и IP-адрес не дублируется, используется IP-адрес локального компьютера:

ping IP-адрес локального хоста

3) Чтобы проверить, что шлюз по умолчанию функционирует и что можно установить соединение с любым локальным хостом в локальной сети, задается IP-адрес шлюза по умолчанию:

ping IP-адрес шлюза

4) Для проверки возможности установления соединения через маршрутизатор в команде ping задается IP-адрес удаленного хоста:

ping IP-адрес удаленного хоста

Синтаксис утилиты ping:

```
C:\Windows\system32>ping

Использование:
    ping [-t] [-a] [-n <число>] [-l <размер>] [-f] [-i <TTL>] [-v <TOS>]
         [-r <число>] [-s <число>] [[-j <список узлов>] | [-k <список узлов>]]
         [-w <тайм-аут>] [-R] [-S <адрес источника>] [-4] [-6] конечный_узел

Параметры
    -t                      Проверка связи с указанным узлом до прекращения.
                            Для отображения статистики и продолжения проверки
                            нажмите сочетание клавиш CTRL+BREAK;
                            для прекращения нажмите CTRL+C.
    -a                      Определение имен узлов по адресам.
    -n <число>              Число отправляемых запросов эха.
    -l <размер>             Размер буфера отправки.
    -f                      Установка в пакете флага, запрещающего
                            фрагментацию (только IPv4).
    -i <TTL>                Задание срока жизни пакетов.
    -v <TOS>                Задание типа службы (только IPv4. Этот параметр
                            недоступен и не влияет на поле TOS в заголовке IP).
    -r <число>              Запись маршрута для указанного числа прыжков
                            (только IPv4).
    -s <число>              Отметка времени для указанного числа прыжков
                            (только IPv4).
    -j <список_узлов>       Свободный выбор маршрута по списку узлов
                            (только IPv4).
    -k <список_узлов>       Жесткий выбор маршрута по списку узлов
                            (только IPv4).
    -w <тайм-аут>           Тайм-аут для каждого ответа (в миллисекундах).
    -R                      Использование заголовка для проверки также и
                            обратного маршрута (только IPv6).
    -S <адрес источника>    Используемый адрес источника.
    -4                      Принудительное использование протокола IPv4.
    -6                      Принудительное использование протокола IPv6.
```

```
C:\Windows\system32>ping -n 10 www.yandex.ru

Обмен пакетами с www.YANDEX.ru [5.255.255.5] с 32 байтами данных:
Ответ от 5.255.255.5: число байт=32 время=115мс TTL=49
Ответ от 5.255.255.5: число байт=32 время=113мс TTL=49
Ответ от 5.255.255.5: число байт=32 время=105мс TTL=49
Ответ от 5.255.255.5: число байт=32 время=107мс TTL=49
Ответ от 5.255.255.5: число байт=32 время=93мс TTL=49
Ответ от 5.255.255.5: число байт=32 время=89мс TTL=49
Ответ от 5.255.255.5: число байт=32 время=101мс TTL=49
Ответ от 5.255.255.5: число байт=32 время=92мс TTL=49
Ответ от 5.255.255.5: число байт=32 время=108мс TTL=49
Ответ от 5.255.255.5: число байт=32 время=94мс TTL=49

Статистика Ping для 5.255.255.5:
    Пакетов: отправлено = 10, получено = 10, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 89мсек, Максимальное = 115 мсек, Среднее = 101 мсек
```

Рис.3.2 «Пример использования утилиты ping»

Изучение маршрута между сетевыми соединениями с помощью утилиты tracert.

Tracert - это утилита трассировки маршрута. Она позволяет проследить путь от данного узла до любого другого узла сети Internet. Хост за хостом показывается прохождение IP-пакетов, при этом выводится название и IP-адрес каждого пройденного хота, а также значение интервала времени, в течение которого был получен ответ.

Утилита использует поле TTL (time-to-live, время жизни) из заголовка IP-пакета и сообщения об ошибках протокола ICMP для определения маршрута от одного хоста до другого.

Утилита tracert может быть более содержательной и удобной, чем ping, особенно в тех случаях, когда удаленный хост недостижим. С помощью нее можно определить район проблем со связью (у Internet-провайдера, в опорной сети, в сети удаленного хоста) по тому, насколько далеко будет отслежен маршрут. Если возникли проблемы, то утилита выводит на экран звездочки (*), либо сообщения типа «Destination net unreachable», «Destination host unreachable», «Request time out», «Time Exeeded».

Утилита tracert работает следующим образом: посылается по 3 пробных эхо-пакета на каждый хост, через который проходит маршрут до удаленного хоста. На экран при этом выводится время ожидания ответа на каждый пакет (Его можно изменить с помощью параметра -w). Пакеты посылаются с различными величинами времени жизни. Каждый маршрутизатор, встречающийся по пути, перед перенаправлением пакета уменьшает величину TTL на единицу. Таким образом, время жизни является счетчиком точек промежуточной доставки (хопов). Когда время жизни пакета достигнет нуля, предполагается, что маршрутизатор пошлет в компьютер-источник сообщение ICMP “Time Exeeded” (Время истекло). Маршрут исследуется путем посылки первого эхо-пакета с TTL=1. Затем TTL увеличивается на 1 в каждом последующем пакете до тех пор, пока пакет не достигнет удаленного хоста, либо будет достигнута максимально возможная величина TTL (по умолчанию 30, задается с помощью параметра -h).

Маршрут определяется путем изучения сообщений ICMP, которые присылаются обратно промежуточными маршрутизаторами.

Примечание: некоторые маршрутизаторы просто молча уничтожают пакеты с истекшим TTL и не будут видны утилите tracert.

Синтаксис:

```
Использование: tracert [-d] [-h максЧисло] [-j списокУзлов] [-w таймаут]
                       [-R] [-S адресИсточника] [-4] [-6] конечноеИмя

Параметры:
    -d                  Без разрешения в имена узлов.
    -h максЧисло        Максимальное число прыжков при поиске узла.
    -j списокУзлов      Свободный выбор маршрута по списку узлов (только IPv4).
    -w таймаут          Таймаут каждого ответа в миллисекундах.
    -R                  Трассировка пути (только IPv6).
    -S адресИсточника   Используемый адрес источника (только IPv6).
    -4                  Принудительное использование IPv4.
    -6                  Принудительное использование IPv6.
```

Рис.3.3 «Синтаксис утилиты tracert»

Утилита ARP.

Основная задача протокола ARP – трансляция IP-адресов в соответствующие локальные адреса. Для этого ARP-протокол использует информацию из ARP-таблицы (ARP-кэша). Если необходимая запись в таблице не найдена, то протокол ARP отправляет широковещательный запрос ко всем компьютерам локальной подсети, пытаясь найти владельца данного IP-адреса. В кэше могут содержаться два типа записей: статические и динамические. Статические записи вводятся вручную и хранятся в кэше постоянно. Динамические записи помещаются в кэш в результате выполнения широковещательных запросов. Для них существует понятие времени жизни. Если в течение определенного времени (по умолчанию 2 мин.) запись не была востребована, то она удаляется из кэша.

Утилита arp выводит для просмотра и изменения таблицу трансляции адресов.

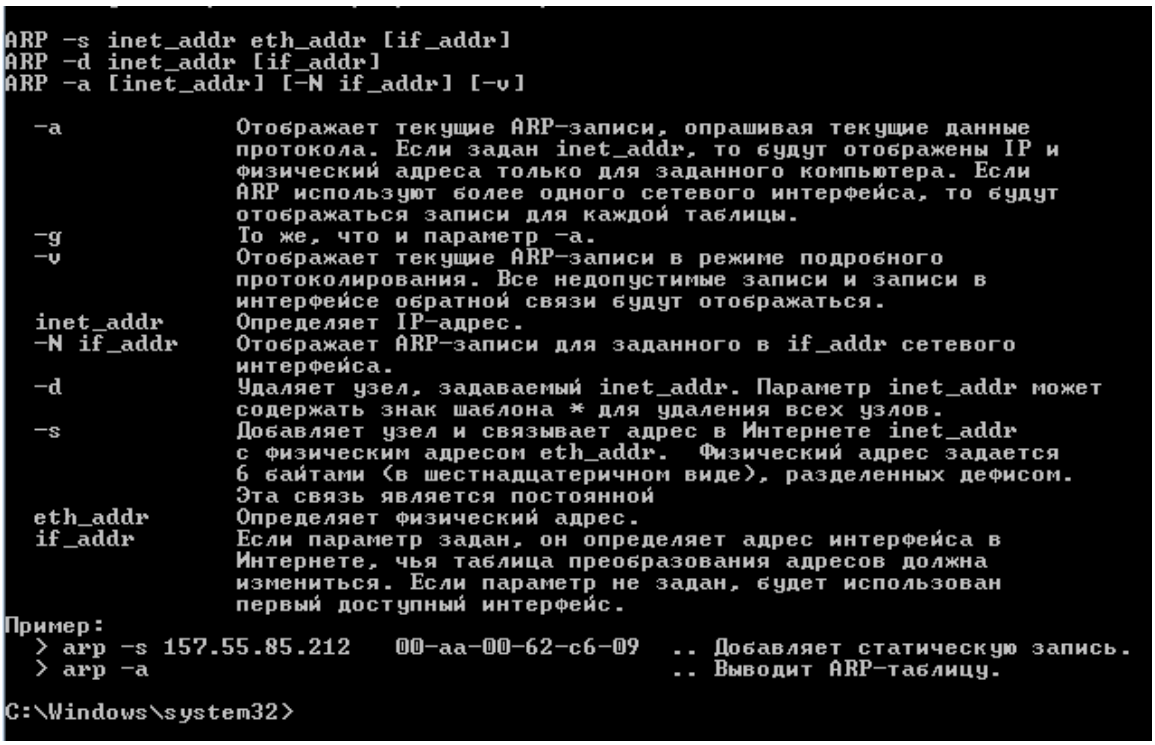


Рис.3.4 «Работа утилиты arp»

Утилита netstat.

Утилита netstat позволяет получить статическую информацию по некоторым из протоколов стека (TCP, UDP, IP, ICMP), а также выводит сведения о текущих сетевых соединениях. Особенно она полезна на брандмауэрах, с ее помощью можно обнаружить нарушения безопасности периметра сети.

Синтаксис:

netstat [-a] [-e] [-n] [-s] [-p protocol] [-r]

Параметры:

-a - выводит перечень всех сетевых соединений и прослушивающихся портов локального компьютера;

-e - выводит статистику для Ethernet-интерфейсов (например, количество полученных и отправленных байт);

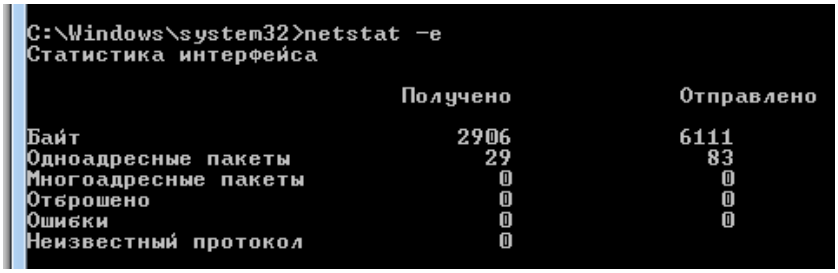


Рис. 3.5 «Работа утилиты netstat»

-n - выводит информацию по всем активным соединениям (например, TCP) для всех сетевых интерфейсов локального компьютера. Для каждого соединения выводится информация об IP-адресах локального и удаленного интерфейсов вместе с номерами используемых портов;

-s - выводит статистическую информацию для протоколов UDP, TCP, ICMP, IP. Ключ «/more» позволяет просмотреть информацию постранично;

-r - выводит содержимое таблицы маршрутизации.

Рис.3.6 «Результат работы команды netstat -s»

ПРАКТИЧЕСКАЯ ЧАСТЬ

1. Получение справочной информации по командам

Вывести на экран справочную информацию по утилитам ipconfig, ping, tracert, hostname. Для этого в командной строке ввести имя утилиты без параметров или с /?. Изучить ключи, используемые при запуске утилит.

Например, ipconfig

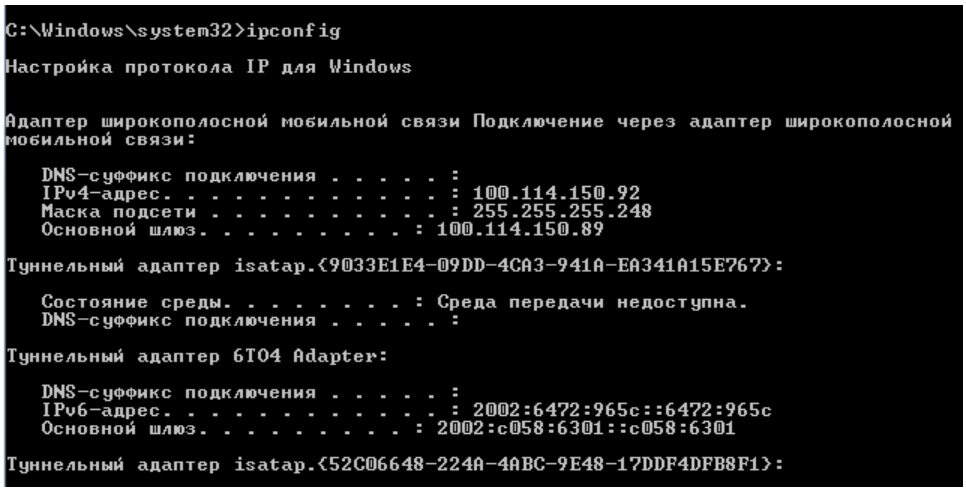


Рис3.7 «Результат работы утилиты ipconfig»

2. Получение имени хоста

Вывести на экран имя локального хоста с помощью команды hostname.

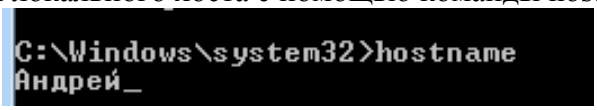


Рис. 3.7 «Результат работы утилиты ipconfig»

3. Изучение утилиты ipconfig

Проверить конфигурацию TCP/IP локального хоста с помощью утилиты ipconfig. Заполнить таблицу:

Имя хоста	
IP-адрес	
Маска подсети	
Основной шлюз	
Используется ли DHCP (адрес DHCP-сервера)	
Описание адаптера	
Физический адрес сетевого адаптера	
Адрес DNS-сервера	
Адрес WINS-сервера	

4. Тестирование связи с помощью утилиты ping

1. Проверить правильность установки и конфигурирования TCP/IP на локальном компьютере.
2. Проверить, правильно ли добавлен в сеть локальный компьютер и не дублируется ли IP-адрес.
3. Проверить функционирование шлюза по умолчанию, послав 5 эхо-пакетов длиной 64 байта.
4. Проверить с помощью ping, можете ли вы обратиться к компьютерам в своей локальной сети. Сравнить результаты выполнения программы ping с указанием адреса компьютера, который отключен, и несуществующего адреса. Отличаются ли эти результаты?
5. Проверить возможность установления соединения с различными удаленными хостами, используя DNS-имена. Определите IP-адреса этих узлов. Отметить время отклика (время кругового обращения пакета). Попробовать увеличить время отклика. Как влияет размер пакета на время кругового обращения?

5. Определение пути IP-пакета

1. Воспользоваться командой tracert для определения числа участков маршрута от вашего компьютера к различным хостам (локальному хосту, шлюзу по умолчанию, удаленному хосту). Отметьте, через какие промежуточные узлы проходят эхо-пакеты.

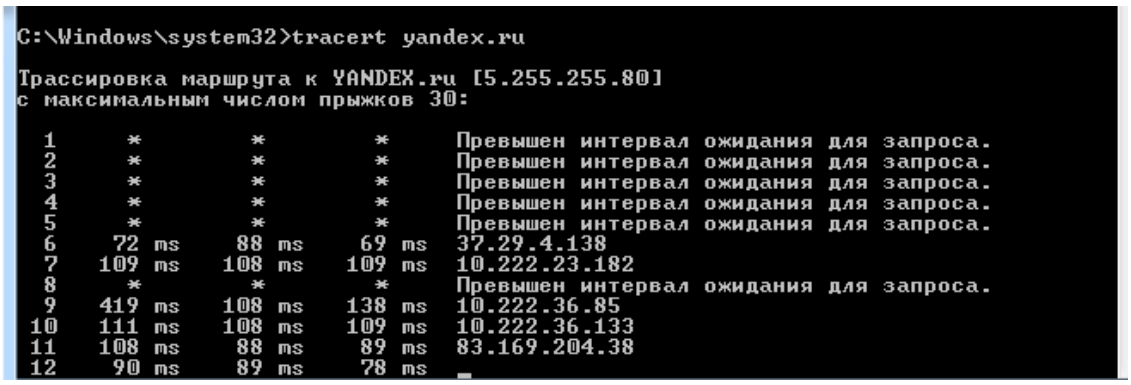


Рис. 3.8 «Результат работы утилиты tracert»

2. Сравнить значения времени кругового обращения, полученные при выполнении программы ping, с числом участков маршрута, полученным при выполнении программы tracert, для ряда адресов назначения. Существует ли зависимость между продолжительностью задержки и числом участков маршрута?

6. Просмотр ARP-кэша

С помощью утилиты arp просмотреть ARP-таблицу локального узла.

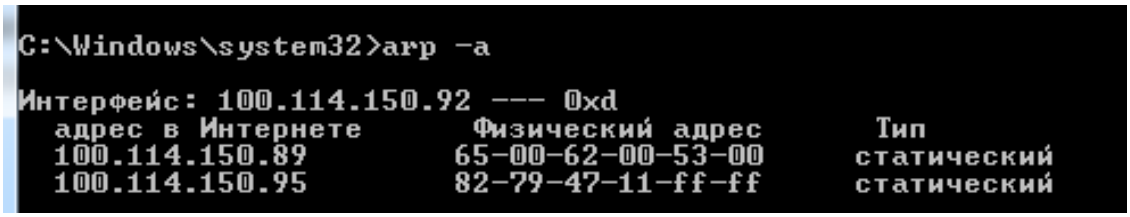


Рис. 3.9 «Результат работы утилиты arp»

7. Получение информации о текущих сетевых соединениях и протоколах стека TCP/IP.

1. С помощью утилиты netstat вывести перечень сетевых соединений и прослушиваемых портов локального узла.

2. Получить статистическую информацию для протоколов UDP, TCP, ICMP, IP.

```
C:\Windows\system32>netstat -a

Активные подключения

  Имя     Локальный адрес         Внешний адрес          Состояние
  TCP     0.0.0.0:135             ╚эфЁхщ_:0              LISTENING
  TCP     0.0.0.0:445             ╚эфЁхщ_:0              LISTENING
  TCP     0.0.0.0:49152           ╚эфЁхщ_:0              LISTENING
  TCP     0.0.0.0:49153           ╚эфЁхщ_:0              LISTENING
  TCP     0.0.0.0:49154           ╚эфЁхщ_:0              LISTENING
  TCP     0.0.0.0:49155           ╚эфЁхщ_:0              LISTENING
  TCP     0.0.0.0:49156           ╚эфЁхщ_:0              LISTENING
  TCP     100.114.150.92:139      ╚эфЁхщ_:0              LISTENING
  TCP     100.114.150.92:49277    cmc:https              CLOSE_WAIT
  TCP     100.114.150.92:49289    95.213.11.147:https    ESTABLISHED
  TCP     100.114.150.92:49366    95.213.4.195:https     ESTABLISHED
  TCP     100.114.150.92:49538    95.213.4.211:https     ESTABLISHED
  TCP     100.114.150.92:49542    95.213.143.223:http    ESTABLISHED
  TCP     100.114.150.92:49602    173.194.44.90:https    ESTABLISHED
  TCP     100.114.150.92:49605    api:https              CLOSE_WAIT
  TCP     100.114.150.92:49606    api:https              CLOSE_WAIT
  TCP     100.114.150.92:49607    api:https              CLOSE_WAIT
  TCP     100.114.150.92:49608    api:https              CLOSE_WAIT
  TCP     100.114.150.92:49609    api:https              CLOSE_WAIT
  TCP     100.114.150.92:49610    api:https              CLOSE_WAIT
  TCP     100.114.150.92:49611    api:https              ESTABLISHED
  TCP     100.114.150.92:49612    api:https              ESTABLISHED
  TCP     100.114.150.92:49613    api:https              ESTABLISHED
  TCP     100.114.150.92:49614    api:https              ESTABLISHED
  TCP     100.114.150.92:49615    api:https              ESTABLISHED
  TCP     100.114.150.92:49616    api:https              ESTABLISHED
  TCP     127.0.0.1:2559          ╚эфЁхщ_:0              LISTENING
  TCP     [::]:135                ╚эфЁхщ_:0              LISTENING
  TCP     [::]:445                ╚эфЁхщ_:0              LISTENING
  TCP     [::]:49152              ╚эфЁхщ_:0              LISTENING
  TCP     [::]:49153              ╚эфЁхщ_:0              LISTENING
  TCP     [::]:49154              ╚эфЁхщ_:0              LISTENING
  TCP     [::]:49155              ╚эфЁхщ_:0              LISTENING
  TCP     [::]:49156              ╚эфЁхщ_:0              LISTENING
  UDP     100.114.150.92:137      *:*
  UDP     100.114.150.92:138      *:*
  UDP     127.0.0.1:48000         *:*
  UDP     127.0.0.1:48001         *:*
```

Рис. 3.10 «Результат работы команды netstat -a»

3. Вывести на экран локальную таблицу маршрутизации. Изучить ее содержимое.

```
C:\Windows\system32>netstat -s

Статистика IPv4

  Получено пакетов                    = 12642
  Получено ошибок в заголовках        = 0
  Получено ошибок в адресах           = 0
  Направлено датаграмм                = 0
  Получено неизвестных протоколов     = 0
  Отброшено полученных пакетов        = 24
  Доставлено полученных пакетов       = 13051
  Запросов на вывод                   = 9783
  Отброшено маршрутов                 = 0
  Отброшено выходных пакетов          = 0
  Выходных пакетов без маршрута       = 0
  Требуется сборка                    = 0
  Успешная сборка                     = 0
  Сбоев при сборке                    = 0
  Успешно фрагментировано датаграмм   = 0
  Сбоев при фрагментации датаграмм    = 0
  Создано фрагментов                  = 0
```

Рис. 3.11 «Результат работы команды netstat -s»

Контрольные вопросы:

1. Какие утилиты можно использовать для проверки правильности конфигурирования TCP/IP? Каковы их возможности?
2. Какова стуктура протокола TCP/IP?
3. Каково место протокола TCP/IP в ЭМВОС (OSI)?
4. Каким образом каманда ping проверяет соединение с узлом сети? Отметьте возможные причины, по которым ping не может связаться с удаленным хостом.
5. Что такое хост?
6. Что такое петля обратной связи?

7. Каков порядок совместного применения утилит ipconfig и ping для диагностики неисправностей в настройке TCP/IP?
8. Что такое статический адрес?
9. Что такое динамический адрес?
10. Сколько промежуточных маршрутизаторов сможет пройти IP-пакет, если его время жизни равно 30?
11. Для чего предназначена и как работает утилита tracert?
12. Каково назначение утилиты arp, протокола ARP? Что такое ARP-кэш?
13. Как просмотреть перечень всех используемых в данный момент портов?
14. Для чего используется команда route? Какую информацию содержит таблица маршрутизации?

Практическая работа №4

Знакомство с ПО для моделирования компьютерных сетей. NetCracker. Создание нового проекта в NetCracker.

Цель работы: Освоение графического интерфейса NetCracker, знакомство с главными приложениями данной программы и общими принципами моделирования сети в ней.

В результате выполнения практических заданий обучающийся должен**:**

Знать:

- ✓ основные этапы создания компьютерных сетей в среде моделирования NetCracker

Уметь:

- ✓ создавать новый проект в NetCracker для моделирования работы спроектированной сети

Задание для практической работы:

1. Изучить теоретическую часть
2. Выполнить практическую часть
3. Ответить на контрольные вопросы
4. Оформить отчет.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Под имитационным моделированием понимают создание компьютерной модели реальной или предполагаемой системы (физической, технологической, финансовой и т. п.) и проведение на построенной модели экспериментов с целью изучения наблюдаемых результатов и/или предсказания будущих результатов.

Системы имитационного моделирования обычно включают также набор средств для подготовки исходных данных об исследуемой сети - предварительной обработки данных о топологии сети и измеренном трафике. Эти средства могут быть полезны, если моделируемая сеть представляет собой вариант существующей сети и имеется возможность провести в ней измерения трафика и других параметров, нужных для моделирования. Кроме того, система снабжается средствами для статистической обработки полученных результатов моделирования.

Можно провести классификацию систем по двум связанным критериям: цена и функциональные возможности. Нужно отметить, что функциональные возможности систем моделирования жестко связаны с их ценой. Анализ предлагаемых на рынке систем показывает, что динамическое моделирование вычислительных систем - дело весьма дорогостоящее. Все системы динамического моделирования могут быть разбиты на две ценовые категории:

- Дешевые (сотни и тысячи долларов).
- High-end (десятки тысяч долларов, в полном варианте - сто и более тысяч долларов).

Дешевые системы отличаются от дорогих тем, насколько подробно удается в них описать характеристики отдельных частей моделируемой системы. Они позволяет получить лишь "прикидочные " результаты, не дают статистических характеристик и не предоставляют возможности проведения подробного анализа системы.

Системы класса high-end позволяют собирать исчерпывающую статистику по каждому из компонентов сети при передаче данных по каналам связи и проводить статистическую оценку полученных результатов. По функциональности системы

моделирования, используемые при исследовании вычислительных систем, могут быть разбиты на два основных класса:

- Системы, моделирующие отдельные элементы (компоненты) системы.
- Системы, моделирующие вычислительную систему целиком.

NetCracker Professional – программный пакет, разработанный фирмой NetCracker Technology (http://www.netcracker.com), позволяет создавать проекты вычислительных сетей разной сложности и топологий, используя технологию имитационного моделирования работы сети.

NetCracker - это система, которая представляет собой CASE-средства автоматизированного проектирования, моделирования и анализа компьютерных сетей. Позволяет провести эксперименты, результаты которых могут быть использованы для обоснования выбора типа сети, сред передачи, сетевых компонент оборудования и программно-математического обеспечения. Программные средства NetCracker позволяют выполнить сбор соответствующих данных о существующей сети без останова ее работы, создать проект этой сети и выполнить необходимые эксперименты для определения предельных характеристик, возможности расширения, изменения топологии и модификации сетевого оборудования с целью дальнейшего ее совершенствования и развития.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Запустите из стартового меню программу NetCracker Professional. Нажмите на кнопку **Ok** в ответ на возможное сообщение о том, что база данных находится в режиме чтения «2read-only mode». Этот режим обычно связан с запретом на запись, установленным системным администратором для файлов пакета и не позволит создавать свои устройства и сохранять их в библиотеках пакета. В остальном поведение программы будет обычным.

1. Главное окно приложения NetCracker Professional показано на рисунке 4.1. Оно состоит из браузера оборудования слева, рабочего окна справа и главного меню вверху.

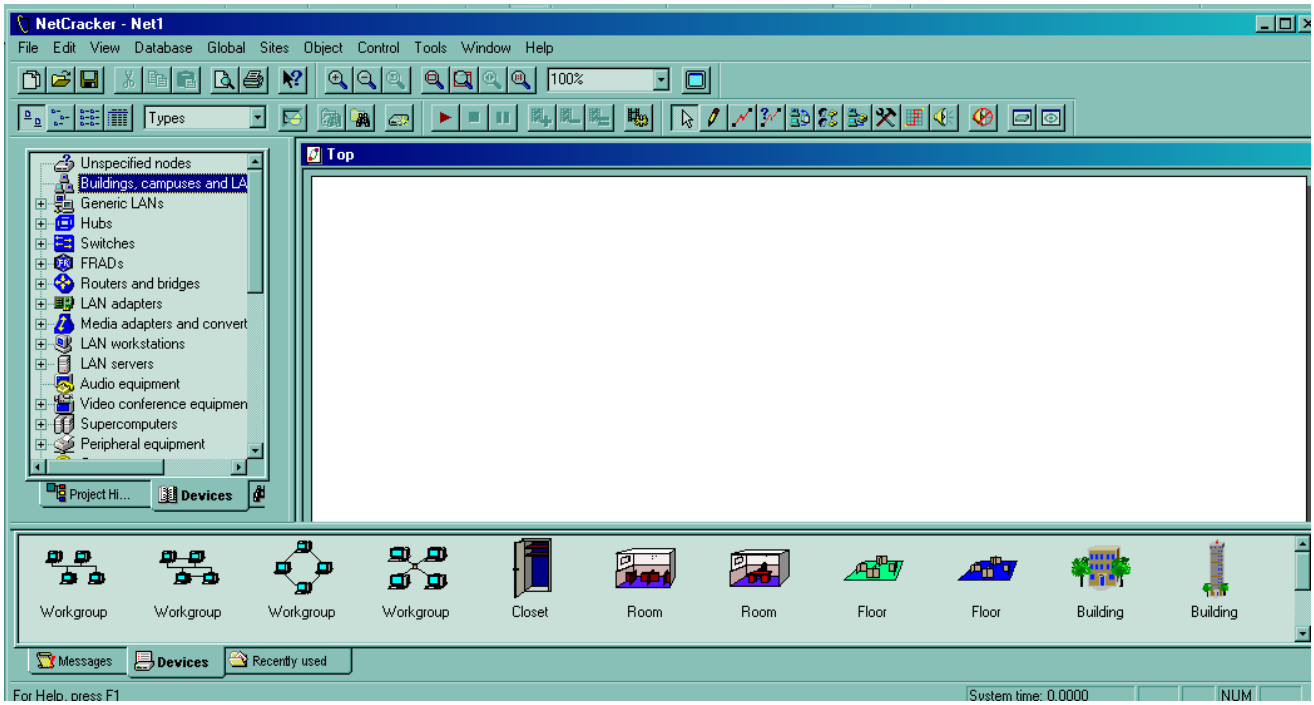


Рис.4.1 «Главное окно приложения NetCracker Professional»

2. Познакомьтесь с содержимым главного меню программы, выбирая основные пункты: *File, Edit, View, Database и др.*3. Откройте файл-пример проекта сети *NetCracker Professional* из подкаталога *Samples* каталога установки программы: **File** → **Open**. Выберете файл *Techno.net* (рис.4.2), нажав кнопку **Open** или двойным щелчком левой кнопки мыши. Проект сети загрузится в рабочее окно (рис.4.3.)

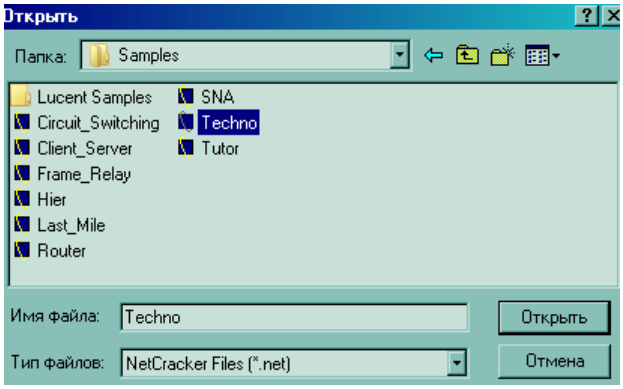


Рис.4.2 «Открытие файла-примера»

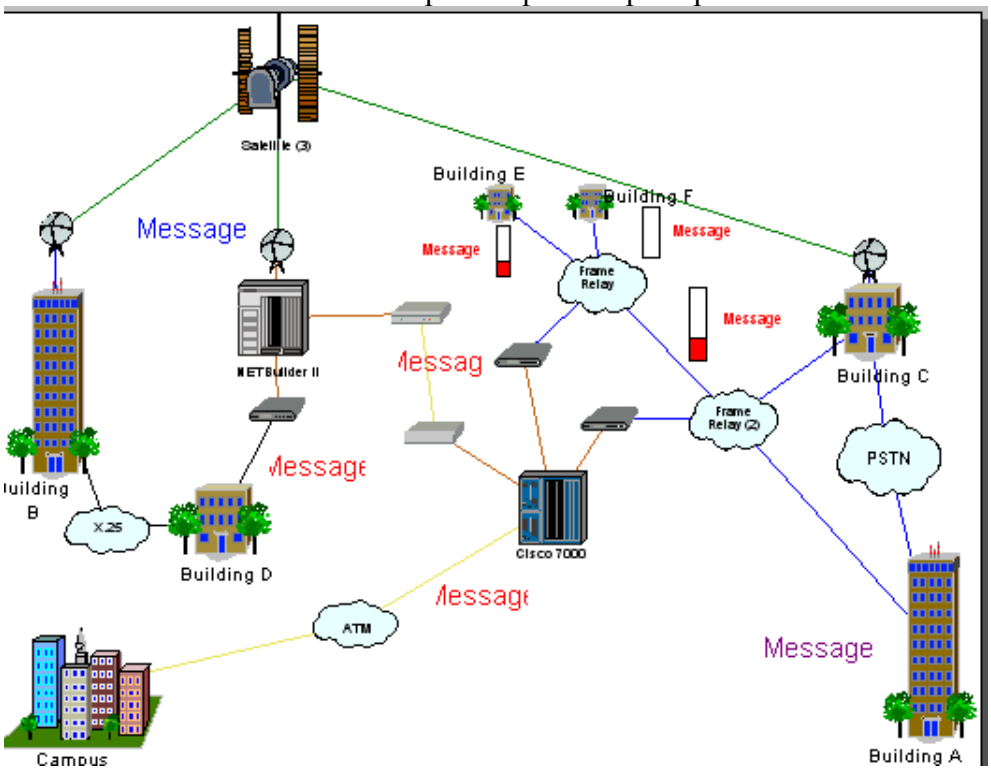
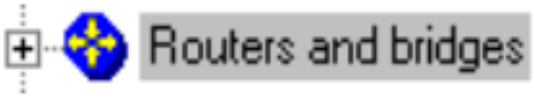


Рис 4.3 «Рабочее окно проекта сети»

3. Масштаб просмотра можно регулировать семейством кнопок Zoom 

4. С помощью линейки прокрутки ознакомьтесь с содержанием браузера оборудования (закладка Devices). Группы устройств, помеченные в узлах знаком “+”, раскрываются на составляющие (рис.4.4)



5. Сортировку оборудования, содержащегося в БД NetCracker, можно производить разными способами:

Database → **Hierarchy** → **Types** (сортировка по типам оборудования)

Database → Hierarchy → Vendors (сортировка по фирмам-производителям)

Например, Вам необходим в сетевом проекте сервер компании *Cray Research C916.* Для этого в разделе *Supercomputers* выберем группу с оборудованием компании *Cray Research*, а в нижнем окне *Devices - сервер C916.* Двойной щелчок левой кнопкой мыши вызовет страницу свойств сервера и вы увидите полный набор его технических характеристик, в т . ч. и *Price/Support*. Пройдите по закладкам и ознакомьтесь с содержащейся информацией о выбранном устройстве (рис.4.5)

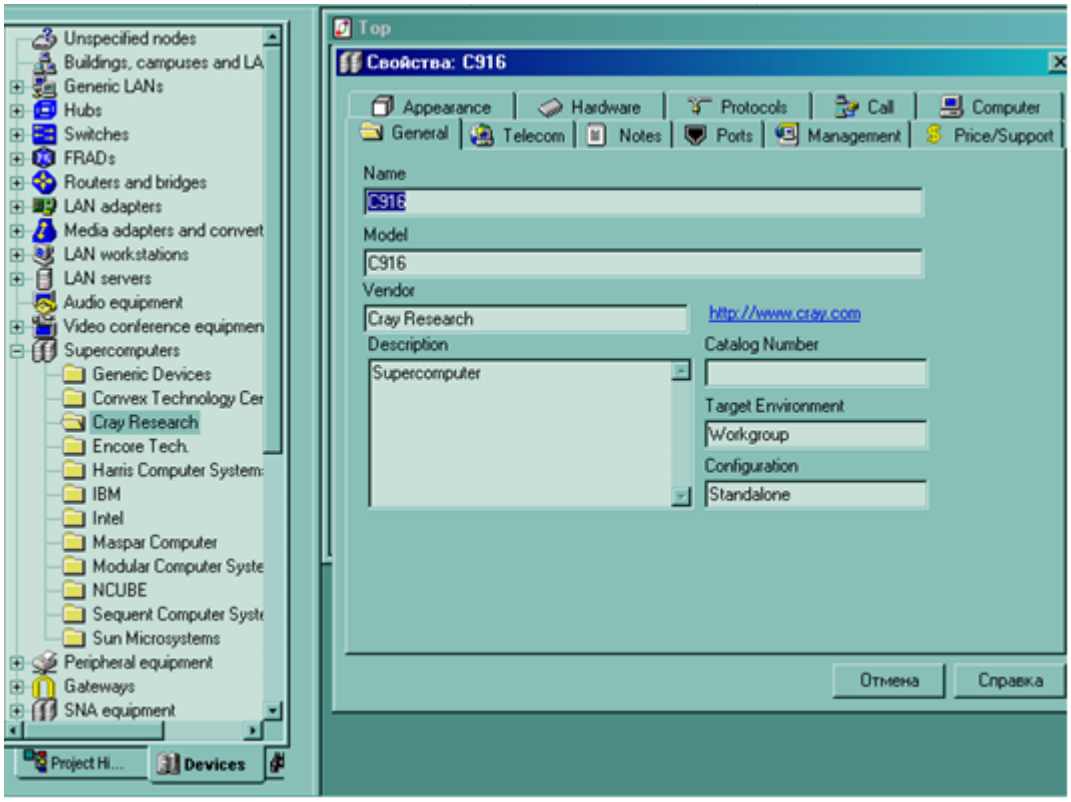


Рис.4.5 «Пример поиска оборудования в NetCracker Professional»

Используя **Database toolbar,** можно осуществлять просмотр состава группы, поиск и создание нового оборудования:

Поиск оборудования производится также из раздела меню *Database*, например, так: **Database → Find → Condition=Description → includes → Frame Relay.** Результаты поиска будут отображаться на закладке браузера оборудования *«Compatible Devices»*. Перейти к обычному режиму браузера можно выбрав закладку *«Devices»*. Часто не требуется использовать в проекте оборудование конкретных производителей, тогда можно воспользоваться «обобщенными» устройствами из раздела **Database → Hierarchy → Vendors → Generic Devices.**

В открытом файле-проекте сети Вы можете посмотреть и изменить характеристики оборудования, включенного в проект. Например, у Вас открыт в данный момент файл Techno.net. Дважды щелкните мышкой по маршрутизатору Cisco 7000, в результате появится окно конфигурации Cisco 7000 *(Рис.* 4.6).

При нажатии кнопки **Device Setup** появляется окно с описанием свойств Cisco 7000.

Если требуется информация об устройствах, которыми укомплектован маршрутизатор Cisco 7000 из проекта Techno.net, нужно выбрать название устройства и нажать кнопку **Plug-in Setup.** Такого же эффекта можно достичь, выбрав название устройства и нажав правую

кнопку мыши, затем в контекстном меню выбрать *Properties* (здесь можно также и прослушать название устройства по-английски **Say description).** Например, посмотрим свойства ATM Interface Processor TAXI multi-mode (рис.4.7)

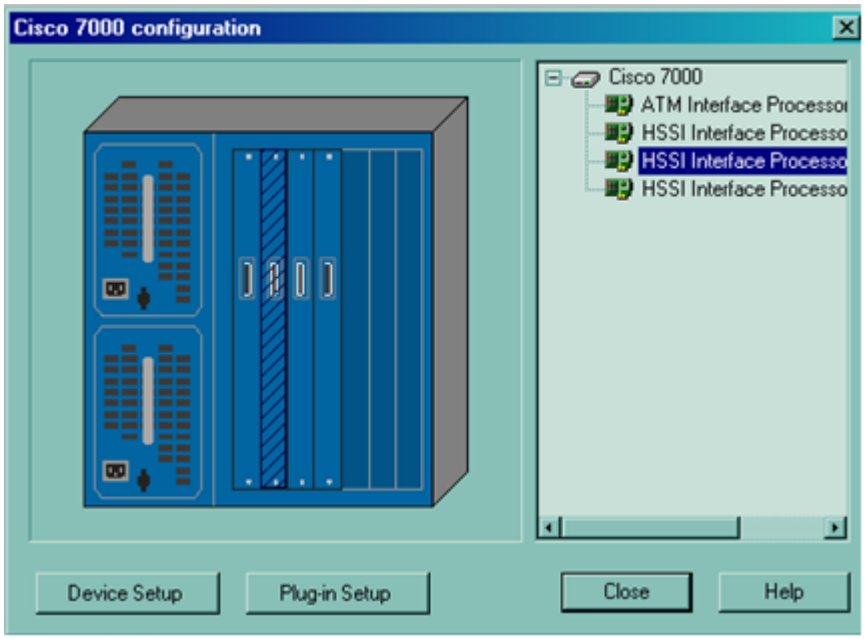


Рис.4.6 «Пример просмотра характеристик оборудования»

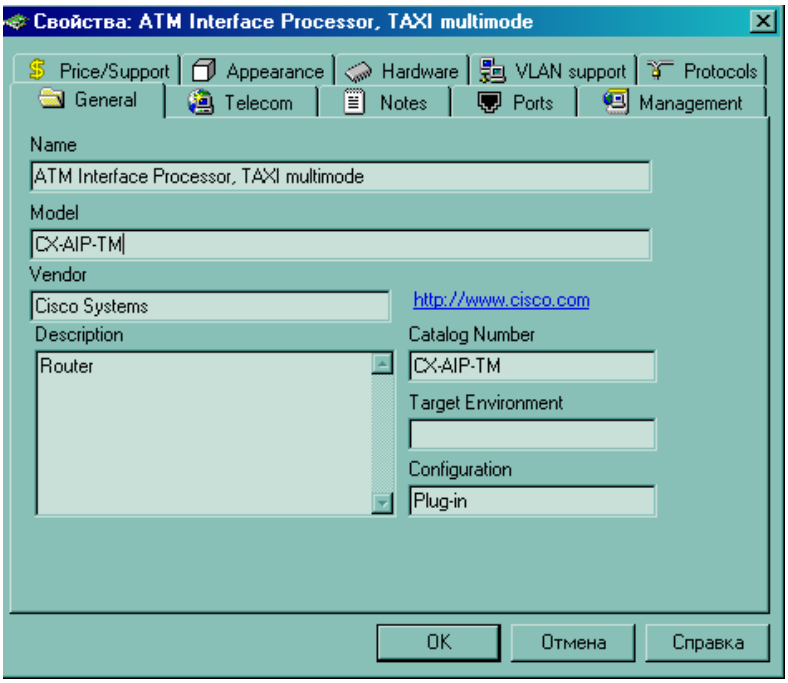


Рис.4.7 «Свойства ATM Interface Processor TAXI multi-mode»

Пройдите по закладкам и ознакомьтесь с содержащейся информацией о выбранном устройстве.

Все устройства, имеющиеся в базе данных NetCracker, из браузера оборудования (страница Devices) можно перетаскивать в рабочее поле своего проекта, удерживая левую кнопку мыши. Устройства, размещенные в проекте, должны быть соединены линиями связи. NetCracker позволяет установить цвет линий в зависимости от используемого в проекте

конкретного типа канала связи. В главном меню **View → Media Colors** установите свои цвета для каждого типа канала связи (Рис.4.8).

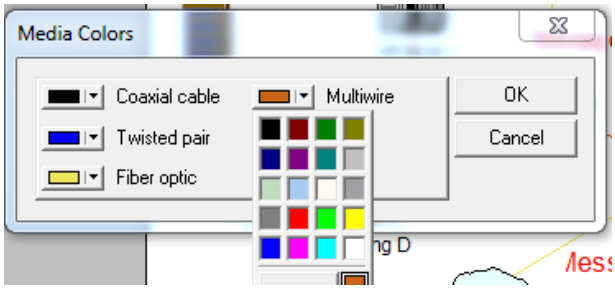


Рис.4.8 «Установление цвета типа канала связи»

Устройства соединяются с помощью мастера соединений **«Link Assistant».**

Среда NetCracker проверяет тип интерфейсов устройств и соединяет только совместимые. Например, в персональных компьютерах **(LanWorkstations → PCs → GenericDevices → PC)** в исходном состоянии есть только последовательные COM-порты, поэтому для соединения их с сетевым оборудованием потребуется установить сетевую карту.

Создайте новый проект **File → New**. Найдите компьютер в БД оборудования **(LanWorkstations → PCs → GenericDevices → PC)** и перенесите методом Drag-and-Drop иконку PC в основное окно проекта TOP. Затем найдите сетевую карту в БД оборудования **(LANadapters → Ethernet → GenericDevices → FastEthernet).** Перенесите иконку «FastEthernetAdapter» методом Drag-and-Drop на компьютер PC. Для сетей Ethernet можно выбрать и готовый «сетевой компьютер» EthernetWorkstation **(LANworkstations → Workstations → GenericDevices → EthernetWorkstation).**

Добавьте в основное окно еще один такой компьютер и коммутатор FastEthernet **(Switches → Workgroup → Ethernet → GenericDevices → EthernetSwich)** и приступайте к соединению двух компьютеров через коммутатор:

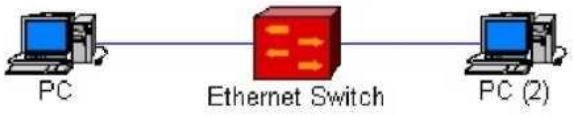


Рис.4.9 «Соединение двух компьютеров через коммутатор»

Порядок соединения таков:

1. Выбрать в панели инструментов инструмент *«Link devices»*: 

2. Убедиться, что модули (компьютеры, коммутаторы, хабы), которые вы планируете соединить, имеют совместимые сетевые порты, например, 100BASE-T.

3. Щелкнуть левой кнопкой мыши сначала по модулю-источнику трафика, затем по модулю-приемнику трафика

4. Нажать в диалоге **«Link Assistant»** на кнопку **«Link»**, а также задать тип, длину и прочие характеристики среды.

5. Закрыть диалог нажав на кнопку **«Close»**

Создание новых устройств (Device Factory)

Несмотря на обилие устройств в базе данных среды NetCracker, иногда требуемое оборудование отсутствует. При наличии доступа по записи к файлам баз данных программы NetCracker (обычный путь **C:\Program Files\NetCracker\DDB**) можно создать новое оборудование. Мастер **Device Factory** запускается из меню **Database**. Новое оборудование

создается на основе существующих шаблонов. На *рис.4.10* показан выбор шаблона для Gigabit Ethernet коммутатора:

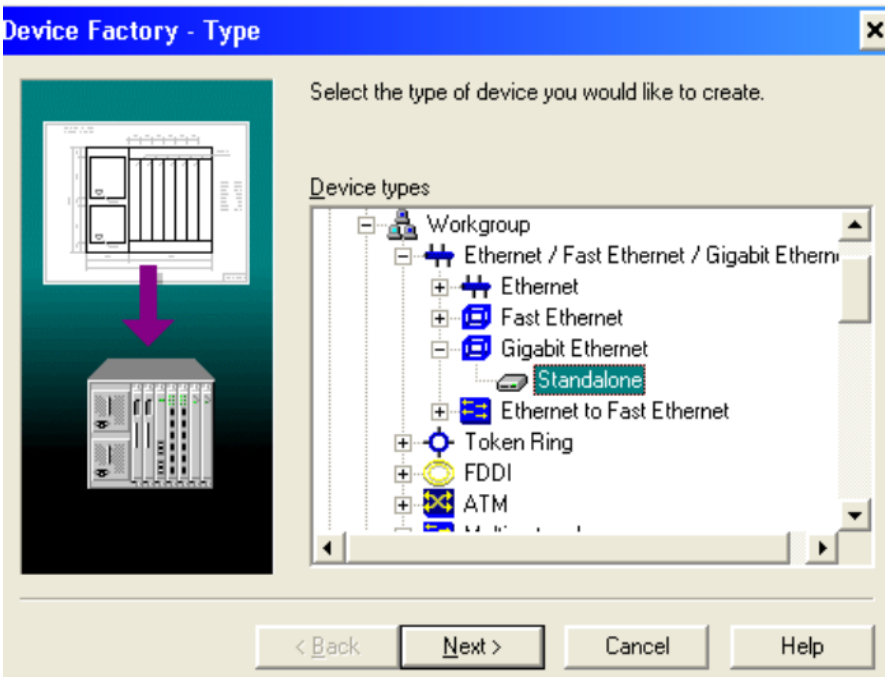


Рис.4.10 «Выбор шаблона для Gigabit Ethernet коммутатора»

Затем последовательно выбираются дополнительные свойства, такие как (например, для коммутаторов): название нового устройства, группы/количество портов (на рисунках выбрано название Gigabit Switch, добавлена одна группа из 24 портов), сигнальные стандарты (100Base-TX, 1000Base-T) для них (рис.4.11)

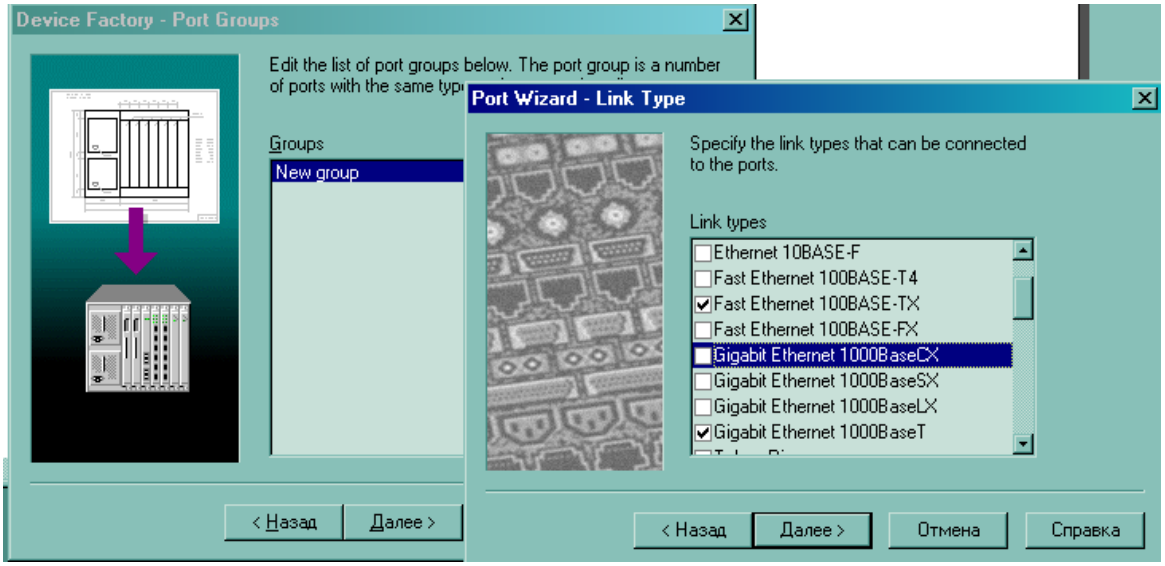


Рис.4.11 «Создание нового устройства»

и тип физической среды:

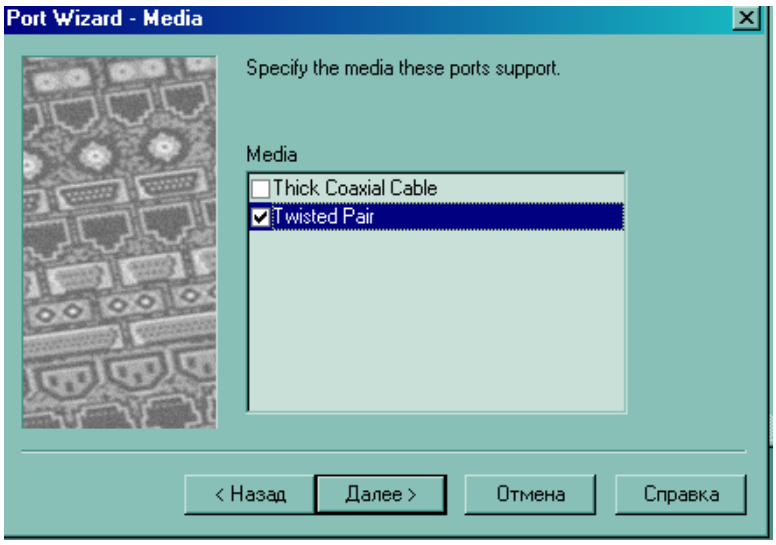


Рис.4.12 «Выбор физической среды для создаваемого оборудования»

В результате получено новое пользовательское устройство «Gigabit Switch» с 24 портами, поддерживающими стандарты 100Base-TX, 1000Base-T. Новое устройство будет доступно при выборе в тулбаре базы данных «User». Параметры устройства, по умолчанию определенные шаблоном (в данном случае Gigabit Ethernet Standalone), требуют проверки. Например, в шаблоне для гигабитного коммутатора задано аномально большое значение задержки (Telecom → Latency) – 0,1 с. С такой задержкой будут 100 %-ные потери данных, проходящих через это устройство. Типовое значение задержки для данного вида оборудования около 0,1 мкс, т. е. на 6 порядков меньше.

Задание трафика

Прежде всего, при задании трафика нужно учитывать процессорные возможности компьютера. Так, при 15 потоках трафика и включенной анимации для устойчивой работы программы требуется процессор не ниже Celeron-800. Проверьте конфигурацию своего компьютера: **My Computer → Properties.** Немного облегчить задачу для компьютера можно отменив визуализацию передаваемых данных: **Global → Data Flow → Uncheck All → Close.** При этом сохраняется возможность наблюдать результаты моделирования, получаемые через индикаторы статистики.

Трафик в моделируемой сети задается с помощью мастера, вызываемого кнопкой панели инструментов «Set traffic». Порядок задания трафика таков:

1. Выбрать в панели инструментов инструмент «Set traffic».
2. Щелкнуть левой кнопкой мыши сначала по модулю-источнику трафика, затем по модулю-приемнику трафика.
3. Наведите указатель мыши на один из стандартных профилей трафиков, например, «InterLAN traffic». Затем щелкните правой кнопкой мыши и в контекстном меню выберите данный профиль трафика (пункт Select). При выборе профиля можно изменять характеристики профиля (кнопка Edit), задавая статистику размеров дейтаграмм «Transaction size», статистику моментов прихода дейтаграмм, пауз «Time between transactions», а также протокол уровня приложения «Application Layer Protocol». Нажав на кнопку Add, можно создать свой профиль трафика с определенными Вами характеристиками. Трафик получит имя Traffic (номер), которое можно изменить, выбрав в контекстном меню трафика пункт Rename (попробуйте это сделать).

4. Посмотрите на определенные Вами потоки данных в сети **Global → Data Flow.** Здесь же можно отредактировать (в том числе и удалить) свойства потоков и профилей трафиков. Учитывайте максимальные пропускные способности каналов передачи данных и не перегружайте их чрезмерно. Замечено, что при перегрузке на порядок индикаторы статистики среды NetCracker дают неверные (произвольные) данные.
5. При выборе трафика клиент-сервер, например, профиля трафика почтового клиента «E-mail (POP)», установите серверное приложение (в данном примере - почтовый сервер). Для этого в браузере оборудования (закладка Devices) найдите группу «Network and Enterprise software». Затем перенесите иконку «E-mail server» методом Drag-and-Drop на компьютер-сервер.

После такой установки программного обеспечения будет возможно назначать клиент-серверные трафики. Назначать такие трафики нужно **от клиента к серверу**: сначала выбирать компьютер-клиент, затем - сервер. Добавить другие виды серверного трафика можно в свойствах программного обеспечения сервера: **Контекстное меню компьютера-сервера Configuration → Контекстное меню серверного программного обеспечения Properties → Закладка Traffic**

При назначении клиент-серверного трафика можно изменять характеристики ответов сервера, задавая статистику размеров дейтаграмм «transaction size», статистику моментов прихода дейтаграмм/пауз «Time between transactions», а также протокол уровня приложения «Application Layer Protocol».

Отчеты

В процессе разработки текущего варианта проекта сети можно получить в NetCracker отчеты о составе проекта. Например:

Tools → Reports → Bill of Material

Можно получить отчет о номенклатуре оборудования, входящего в проект сети, ценах каждой единицы оборудования, общей цены проекта:

Tools → Reports → Device Summary

или спецификацию всех единиц оборудования. Подобные спецификации можно сгенерировать и по отдельным классам оборудования (например, Workstations, Servers, Hubs, и т. д.). Полученные таким образом отчеты можно распечатать или сохранить в файл, воспользовавшись панелью меню по работе с отчетами.

Задание на практическую работу

1. Ознакомиться с системой NetCracker и получить начальные навыки работы с программой.
2. Используя имеющийся проект TUTOR.NET, научиться добавлять устройства и устанавливать связи между ними при создании проекта, настраивать протоколы, определять интенсивность трафика, анализировать загруженность сети и т.п.
3. Составить произвольную схему сети. Примерами могут стать типовые варианты: «Домашняя сеть» - соединение двух или более ПК посредством интерфейса Ethernet. «Малая АТС» - в состав ее могут входить несколько ТА и РВХ (учрежденческая) станция. «Выход ПК в сеть Интернет» - для моделирования такой сети потребуются Dial-up модем, ПК, а также модуль сети WAN. По окончанию моделирования необходимо задать трафик между оконечными устройствами и продемонстрировать работоспособность созданной модели.

Контрольные вопросы:

1. Что представляет собой компьютерная система NetCracker Technology?

2. Опишите главное окно NetCracker.
3. Что представлено в проекте TUTOR.NET?
4. Какими способами достигается запуск анимации?
5. Как добавить выбранное устройство в рабочую зону?
6. Как получить информацию об имеющемся в рабочей зоне конкретном устройстве?
7. Что позволяет делать кнопка Device Factory?
8. Как установить связь между устройствами?
9. Что устанавливает кнопка Set Traffic?
10. Как указать вид статистической информации о потоках данных?

Практическая работа №5

Моделирование передачи данных в сети

Цель работы: Знакомство с возможностями NetCracker в отношении анализа трафиков в сети посредством моделирования процессов передачи данных.

В результате выполнения практических заданий обучающийся должен:

Знать:

- ✓ Основные возможности NetCracer;
- ✓ принципы моделирования сети в NetCracer**.**

Уметь:

- ✓ собирать статистику, характеризующую работу сети;
- ✓ анализировать трафик в сети посредством моделирования процессов передачи данных.

Задание для практической работы:

1. Изучить теоретическую часть
2. Выполнить практическую часть
3. Ответить на контрольные вопросы
4. Оформить отчет.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Можно выделить две характеристики трафика — единица данных и способ упаковки этих единиц. Единицей данных может быть: бит, байт, октет, сообщение, блок. Они упаковываются в файлы, пакеты, кадры, ячейки. Они могут также передаваться без упаковки.

Скорость измеряется в единицах данных за единицу времени. Например, пакеты в секунду, байты в секунду, транзакции в минуту и т. д. Скорость также определяет время, требуемое для передачи единицы данных по сети.

Реальный размер передаваемых по сети данных складывается из непосредственно данных и необходимого информационного обрамления, составляющего накладные расходы на передачу. Многие технологии устанавливают ограничения на минимальный и максимальный размеры пакета. Так, например, для технологии X.25 максимальный размер пакета составляет 4096 байт, а в технологии Frame Relay максимальный размер кадра составляет 8096 байт.

Можно выделить четыре наиболее общие характеристики трафика:

- ✓ «взрывообразность»,
- ✓ терпимость к задержкам,
- ✓ время ответа,
- ✓ емкость и пропускная способность.

Эти характеристики с учетом маршрутизации, приоритетов, соединений и т. д. как раз и определяют характер работы приложений в сети.

«Взрывообразность» характеризует частоту посылки трафика пользователем. Чем чаще пользователь посылает свои данные в сеть, тем она больше. Пользователь, который посылает данные регулярно, в одном темпе, сводит показатель «взрывообразности» практически к нулю. Этот показатель можно определить отношением максимального (пикового) значения трафика к среднему. Например, если максимальный объем пересылаемых данных в часы пик составляет 100 Мбит/с, а средний объем — 10 кбит/с, показатель «взрывообразности» будет равен 10.

Терпимость к задержкам характеризует реакцию приложений на все виды задержек в сети. Например, приложения, обрабатывающие финансовые транзакции в реальном масштабе времени, не допускают задержек. Большие задержки могут привести к неправильной работе таких приложений.

Приложения сильно различаются по допустимому времени задержки. Есть приложения, работающие в реальном времени (видеоконференции) — там время задержки должно быть крайне малым. С другой стороны, встречаются приложения, терпимые к задержкам в несколько минут или даже часов (электронная почта и пересылка файлов). На рис. 2. показано, из чего составляется общее время реакции системы.

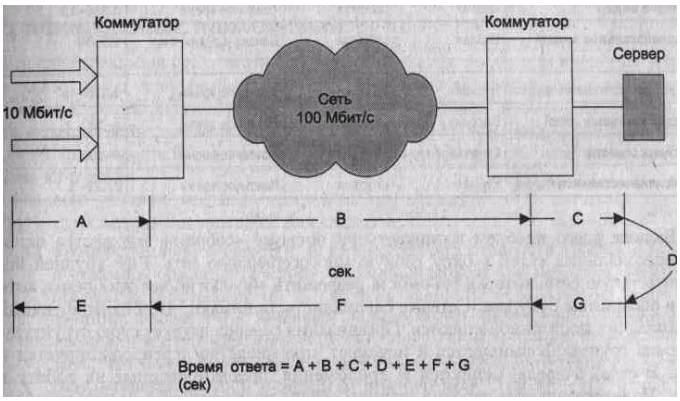


Рис. 5.1 «Общее время ответа сети»

Понятия емкости и пропускной способности сети связаны между собой, но, по сути, это не одно и то же. *Емкость сети* — это реальное количество ресурсов, доступных пользователю на определенном пути передачи данных. *Пропускная способность* сети определяется общим количеством данных, которые могут быть переданы в единицу времени. Емкость сети отличается от пропускной способности сети из-за наличия накладных расходов, которые зависят от способа использования сети. Таблица 2.1 содержит характеристики трафика для различных приложений.

Нет ни пользователей, ни разработчиков, которые не были бы озабочены оптимальностью создаваемой сетевой инфраструктуры. При этом главный вопрос: будет ли работа сети удовлетворительной по истечении некоторого времени после ее внедрения?

Таблица 6.1 «Характеристики трафика разных приложений»

Приложение/ Характеристика	Загруженность трафика	Терпимость к задержкам	Время ответа	Пропускная способность, Мбит/с
Электронная почта	Высокая	Высокая	Регламентируется	0.004-0.20
Передача файлов	Высокая	Высокая	Регламентируется	0.01-600
CAD/CAM-системы	Высокая	Средняя	Близко к РВ	1-100
Обработка транзакций	Высокая	Низкая	Близко к РВ	0.064-1.544
Связь локальных сетей	Высокая	Высокая	Реальное время	4-100
Доступ к серверу	Средняя	Высокая	Реальное время	4-100
Высококачественное аудио	Низкая	Низкая	Реальное время	0.128-1

Трафик разных приложений

В последнее время все отчетливее прослеживается тенденция введения в приложения услуг телефонии, групповой работы над документами, обработки сообщений, видео и т. д.

Эта тенденция определяет требования к сетевой магистрали, которая, комбинируя ЛВС-, MAN- и WAN-магистрали, должна иметь многосервисную основу и передавать любые типы трафика с требуемым качеством.

Можно условно разделить трафик на три категории, отличающиеся друг от друга требованиями к задержке при передаче:

D Трафик реального времени. К этой категории относятся трафик с аудио- и видеоинформацией, не допускающий задержки при передаче. Задержка обычно не превышает 0,1 с, включая время на обработку на конечной станции. Кроме того, задержка должна иметь небольшие колебания во времени (эффект «дрожания» должен быть сведен к нулю). Следует отметить, что при сжатии информации трафик данной категории становится очень чувствительным к ошибкам при передаче. При этом из-за требования малой задержки возникающие ошибки не могут быть исправлены с помощью повторной посылки;

U Трафик транзакций. Эта категория требует задержки до 1 с. Увеличение этого предельного значения заставляет пользователей прерывать свою работу и ждать ответа, потому что только после получения ответа они могут продолжить отправлять свои данные. Поэтому большие задержки приводят к уменьшению производительности труда. Кроме того, разброс в значениях задержки приводит к дискомфорту в работе. В некоторых случаях превышение допустимого времени задержки приведет к сбою рабочей сессии и пользовательским приложениям потребуется начать ее вновь;

O Трафик данных. Эта категория трафика может работать практически с любой задержкой, вплоть до нескольких секунд. Особенностью такого трафика является повышенная чувствительность к доступной пропускной способности, но не к задержкам. Увеличение пропускной способности влечет за собой уменьшение времени передачи. Приложения, передающие большие объемы данных, разработаны, в основном, так, что захватывают всю доступную полосу пропускания сети. Редкими исключениями являются приложения потокового видео. Для них важны и пропускная способность и минимизация времени задержки.

Внутри каждой рассмотренной категории графики классифицируются по присвоенным им приоритетам. Трафик, имеющий более высокий приоритет, получает предпочтение при обработке. Примером приоритетного трафика может быть транзакция с заказом.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Запустите из стартового меню программу NetCracker. Откройте файл - пример проекта Router.net. Далее, читайте и выполняйте задания.

1. Проверьте значение задержки т. н. переходного периода (Global → Model Settings → Simulation → Warm-Up period). В рассматриваемых примерах и заданиях значение задержки должно быть нулевым.
2. Нажмите кнопку «старт» на панели управления  . Вы увидите схему (рис.5.1)
3. Задайте статистические индикаторы Average Workload (средняя нагрузка), Average Utilization (средняя загрузка/использование). Для этого выделите канал MathLab ←→ Cisco7000, щелкнув левой кнопкой мыши по линии канала, и в контекстном меню (щелчок правой кнопкой мыши) выберите Statistics. Пометьте соответствующие индикаторы.

В свойствах индикаторов можно установить единицу измерения и размер шрифта. Запомните значения этих двух индикаторов.

4. Остановите симулятор и измените среднее паузы между пакетами (Time Between Transactions) для трафика **Global → DataFlow → Steve → Chris → Edit → InterLAN traffic → Edit** со значения 0.008 сек на значение 0.08 сек. Запустите снова симулятор и посмотрите показания установленных Вами индикаторов. Объясните изменение показаний

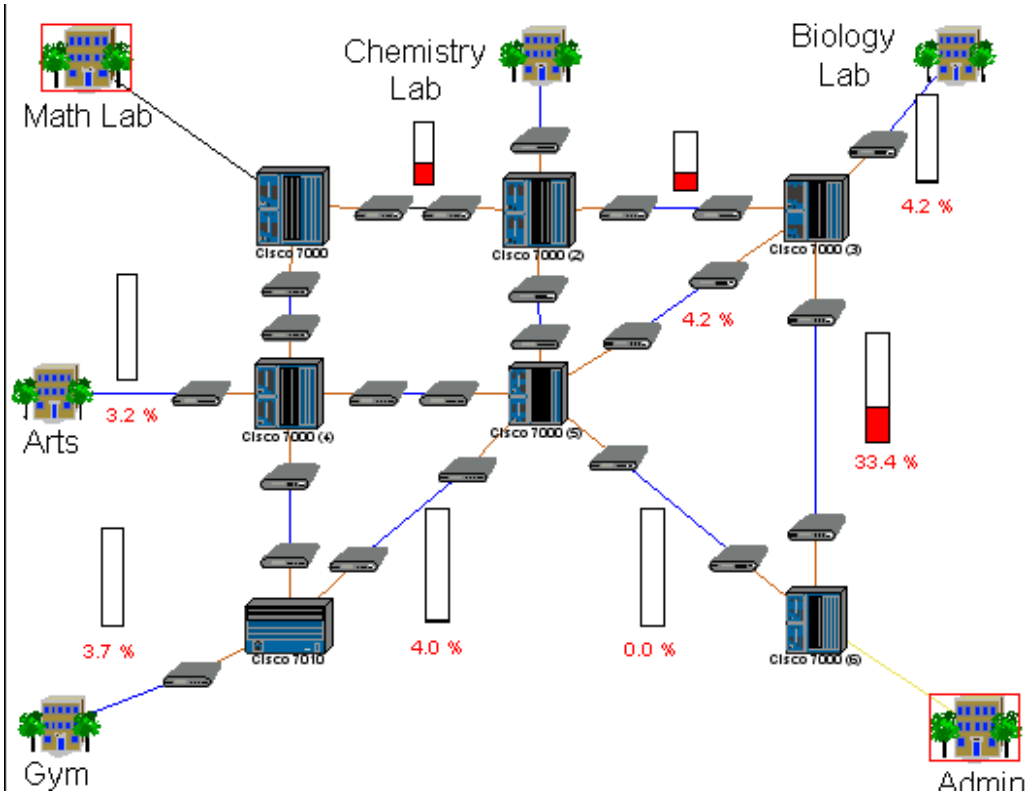


Рис 5.2 «Router.net»

Параметрами анимации можно управлять с помощью меню **Control → Animation Setup** (рис.5.3)

Измените параметры и нажмите на кнопу OK. Обратите внимание на изменения в работе сети проекта.

Рассмотрите работу сети более подробно. Для этого щелкните левой кнопкой мыши на открытом проекте, на здании, отмеченном как Math Lab. Перемещаться по иерархии сети можно и на закладке браузера оборудования «Project Hierarchy».

Среда NetCracker позволяет планировать выделение IP-адресов. Планировщик запускается: **Tools → IP Planner...** Выделение адресов возможно только для отдельных физических сегментов, формируемых парой Hub и порт Switch. В проекте Router.net распределение может выглядеть, например, так (рис.5.4).

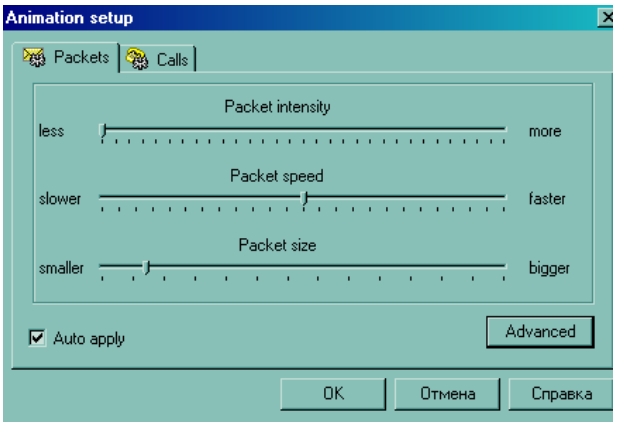


Рис.5.3 «Окно управления параметрами анимации»

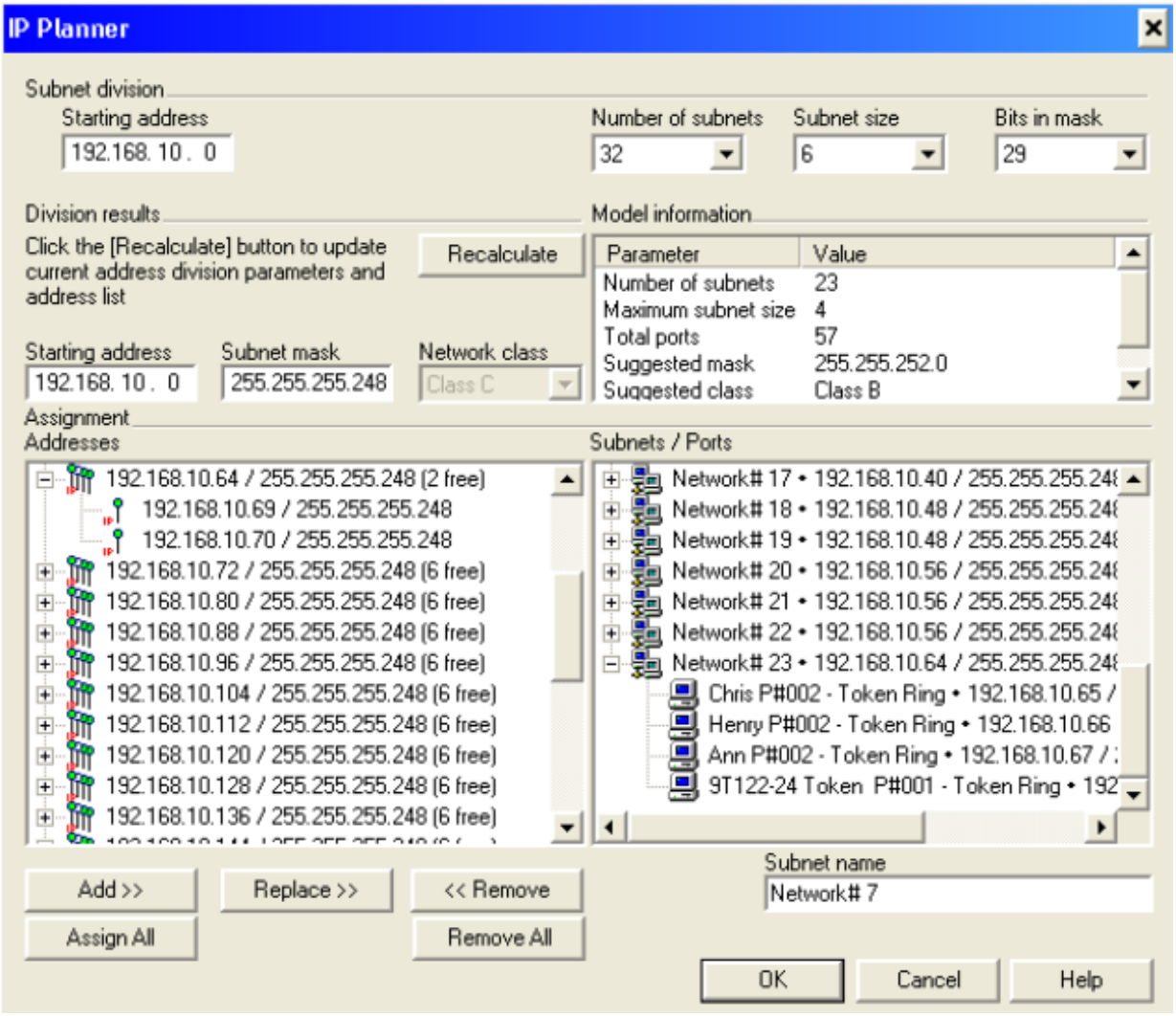


Рис.5.4 «Выделение IP-адресов в проекте Router.net»

Задание на практическую работу

1. Выполнить все пункты практической работы для проекта-примера router.net.
2. Выполнить все пункты практической работы для проекта-примера techno.net.

Контрольные вопросы

1. Какие существуют категории трафика?
2. Для каких приложений какие типы трафика характерны?
3. Перечислите и раскройте общие характеристики трафика.

Практическая работа №6

Построение сетевого проекта на базе технологий Ethernet

Цель работы: освоение Graphical User Interface(GUI) данной программы, знакомство с главными приложениями **NetCracker** и общими принципами моделирования сети в ней

В результате выполнения практических заданий обучающийся должен:

Знать:

- ✓ главные приложения NetCracer;
- ✓ принципы моделирования сети в NetCracer**.**

Уметь:

- ✓ проектировать сегменты сети;
- ✓ собирать статистику, характеризующую работу спроектированного сегмента сети.

Задание для практической работы:

1. Выполнить практическую часть
2. Ответить на контрольные вопросы
3. Оформить отчет.

ПРАКТИЧЕСКАЯ ЧАСТЬ

1. Создайте новый проект сети **File → New**
2. Разместите в рабочем окне устройства, входящие в наш создаваемый проект.

Все устройства, имеющиеся в базе данных NetCracker, из браузера оборудования (страница **Devices**) можно перетаскивать в рабочее поле своего проекта, удерживая левую кнопку мыши. При этом курсор приобретает вид ⃠.

Для примера воспользуемся абстрактными устройствами из раздела **Database → Hierarchy → Vendors → Generic Devices .**

Из раздела **Generic Devices** выберем **Lan workstations** и из них выберем **PCs.** Разместим две таких рабочих станции (PC и PC2) в своем проекте.

3. Щелкните дважды левой кнопкой мыши по размещенному в Вашем проекте устройству PC. Вы увидите окно **PC Configuration,** нажмите кнопку **Device Setup.** При просмотре содержимого закладок, вы увидите, что информация практически на всех из них отсутствует – это объясняется тем, что мы выбрали абстрактное устройство и никаких установок, касающихся его работы, не указали.

На закладке **Ports,** мы видим, что компьютер имеет только COM порт, и для включения его в сеть необходимо добавить оборудование – сетевой адаптер Ethernet. Для этого выберем **Lan adapter→Ethernet**, среди пиктограмм устройств этого семейства (окно внизу) выберем левой кнопкой мыши **Ethernet adapter** и перетащим его в PC1 в проекте. Теперь щелкнем левой кнопкой по PC1 и ознакомимся в окне **PC Configuration** с изменением его комплектации **(<Device Setup>**, закладка **Ports**).

Аналогичные действия необходимо совершить с PC2.

4. Размещенные в нашем проекте рабочие станции должны быть соединены в сегмент Ethernet. Чтобы это сделать выберете

Database→Hierarchy→Types,

Generic LANs→Thin Ethernet Segment

и перетащите его на проект (рис.6.1).

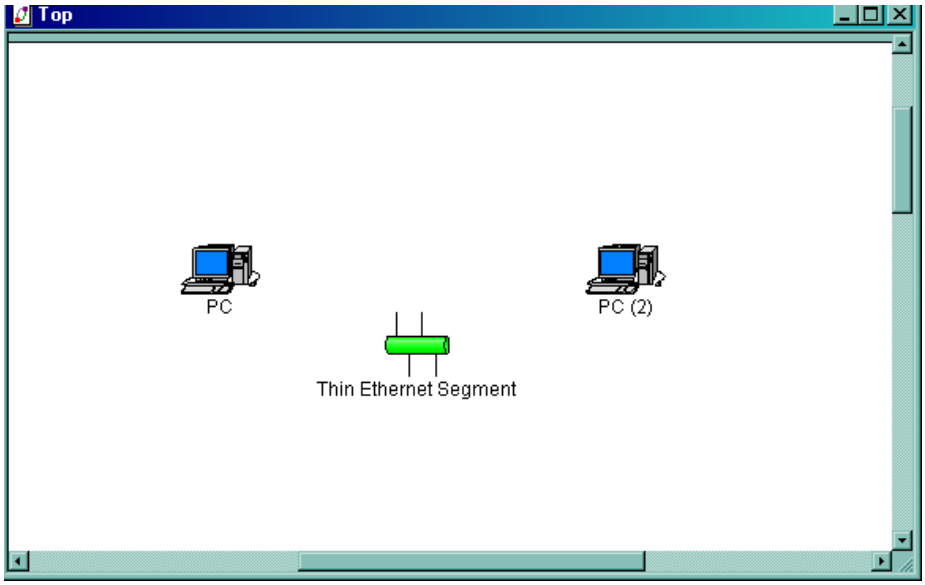


Рис.6.1 «Добавление рабочих станций на проект»

Теперь необходимо нажать кнопку Link devices  и щелкнуть левой кнопкой мыши по **PC1**, затем по **Thin Ethernet Segment** на проекте. Появится окно **Link Assistant**, где Вы увидите вариант соединения слева Device#1 (то есть **PC1**) и справа Device#2 (то есть **Thin Ethernet Segment)** через порт Ethernet.

Нажмите кнопку **<Link>** и введите в метрах длину соединения, например length = 20 m. Вы также увидите установившиеся параметры соединения (Link settings), такие как: стандарт соединения – Ethernet 10Base2 скорость обмена данными - 10 Мбит/с, тип соединения – коаксиальный кабель. Нажмите кнопку **<Close>.** В окне проекта Вы увидите появившееся соединение, цвет линии будет соответствовать установленному Вами цвету соединения коаксиальным кабелем (рис.6.2)

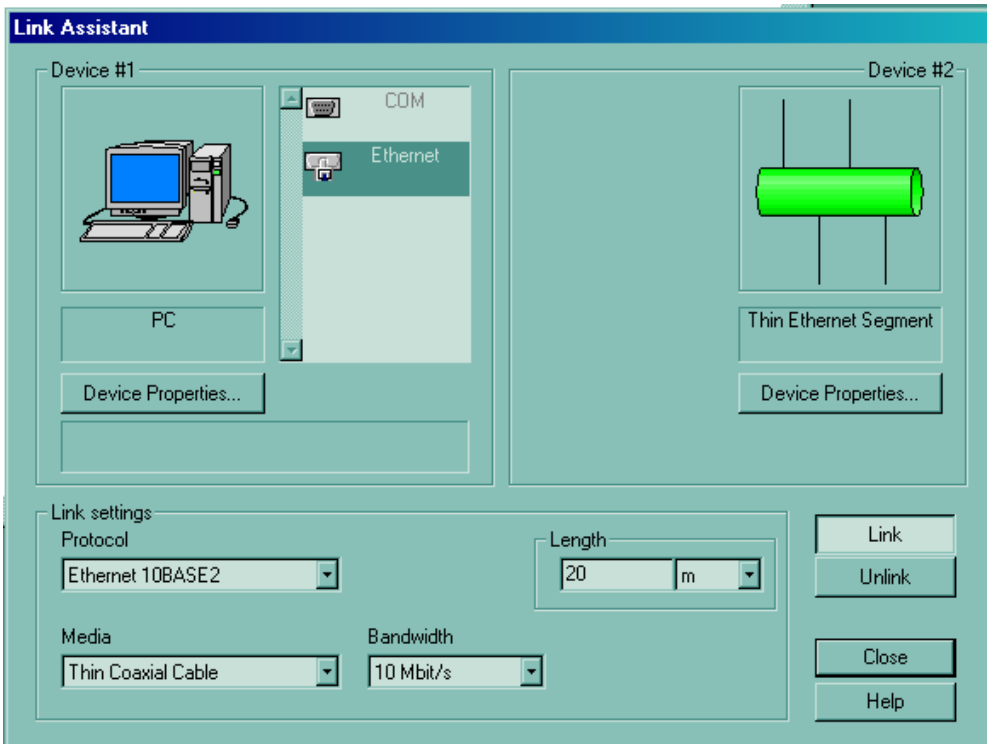


Рис.6.2 «Окно Link Assistant»

Аналогичным образом подключите **PC2.**

Сгенерируйте различные формы отчетов о созданном Вами проекте, ознакомьтесь с их содержанием.

В этой работе используются рабочие станции типа Ethernet-ЭВМ со встроенным сетевым адаптером Ethernet. Сетевой адаптер предназначен для сопряжения сетевых устройств со средой передачи в соответствии с принятыми правилами обмена информацией. Адаптеры Ethernet представляют собой плату, которая вставляется в свободный слот материнской платы.

Рабочие станции соединены между собой коаксиальным кабелем типа Thick Ethernet. Такой кабель способен передавать данные со скоростью 10 Мбит/с на расстояние до 500 м.

Для того, чтобы задать трафик, необходимо войти в соответствующий режим и выбрать нужный тип сетевого трафика. На рис. 6.3 показан пример.

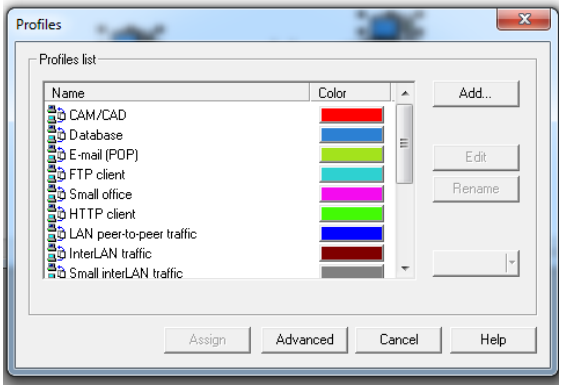


Рис.6.3 «Окно выбора сетевого трафика»

На рис. 6.4 изображен диалог установки параметров трафика.

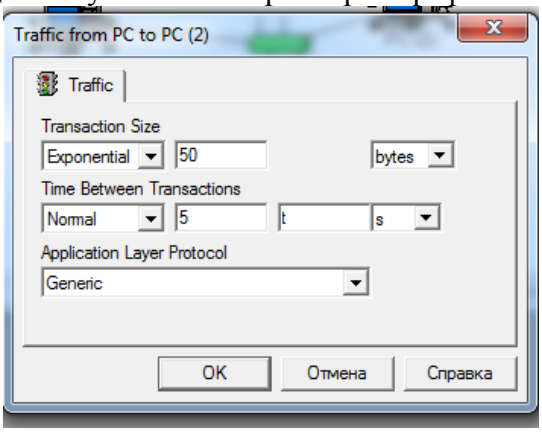


Рис.6.4 «Окно установки параметров трафика»

Задание на практическую работу

Задание на лабораторную работу представляет собой несколько вариантов той или иной конфигурации сетевого шаблона. Данные необходимо брать из табл. 6.1.

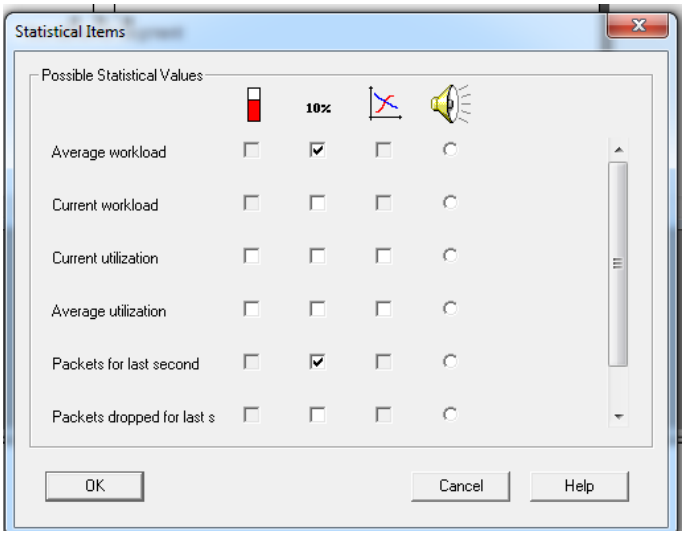
Таблица 6.1 «Задание на практическую работу №6»

№ вариант	Число ра бочих станций	Типы тра фика	Параметры трафика	
			Transaction size	Time between transaction
1	3	Traffic (15)	Exponential 50 bytes	Normal 5 to 1s
2	3	Inter! AN Traffic	Exponential 5 bytes	Constant 10 s
3	3	Small Office	Uniform 500 to 600 bytes	Erlang 0,04 s
4	4	Traffic (15)	Exponential 50 kbyte	Normal 5 to 1 s
5	4	InterLAN Traffic	Exponential 5 kbyte	Conslant 2s
6	4	Small Office	Constant 500 kbits	Longnormal 0.04 to 0.08 s
7	2	Small Office	Constant 500 kbits	Longnormal 0.04 to 0.08 s
8	3	InterLAN Traffic	Longnormal 0.04 to 0.08s	Constant 2s
9	5	Traffic (15)	Exponential 50 kbyte	Normal or 5 to 1 s
10	5	Traffic (15)	Constant 100 kbits	Normal 3 to 1 s
11	3	Inter LAN Traffic	Exponential 50 kbyte	Constant 2s
12	4	Small Office	Constant 5 kbytes	Exponential 0.04 s
13	4	taterLAN Traffic	Gamma 0.5 to 0.5	Normal 0.08 to 0.5s
14	3	Small Office	Constant 5 kbytes	Exponential 0.04 s
15	2	Traffic (15)	Exponential 50 kbytes	Longnormal 0.04 to 0.08

Трафик во всех вариантах должен быть двунаправлен. Изменяя параметры трафика согласно табл. 6.1, необходимо вести статистические данные по сетевым устройствам. Для этого нужно щёлкнуть правой кнопкой мыши на нужном устройстве и появится выпадающее меню.

После выбора соответствующего пункта появится окно «Statistical Items». После выбора статистических данных необходимо запустить процесс моделирования. Для выполнения работы необходимо подставить данные из таблицы 6.1 и получить статистические данные.

На рабочих станциях измерить среднюю рабочую нагрузку (Average Workload) пакеты, обработанные за последнюю секунду (packets last for second).



На линиях связи необходимо измерить среднюю рабочую нагрузку (Average Workload).

Изменить параметры трафика, заданные в табл. 6.1 в любую сторону и проанализировать результаты.

Контрольные вопросы:

1. Какие категории трафика существуют?
2. Для каких приложений какие типы трафика характерны?
3. Как изменить тип трафика? Параметры трафика?

Практическая работа №7

Самостоятельное создание модели в среде NetCracker Professional

Цель работы: получение практических навыков работы с **NetCracker**.

В результате выполнения практических заданий обучающийся должен **иметь практический опыт:**

- создание модели сети;
- задание трафиков и получение результатов моделирования (определение загруженности каналов, «поиск узких мест» и т . п.).

В результате выполнения практических заданий обучающийся должен **уметь:**

- выбирать оборудование, необходимое для создания модели сети;
- задавать трафик в созданной модели сети;
- получать результаты моделирования созданной модели сети;

В результате выполнения практических заданий обучающийся должен **знать:**

- основные этапы создания модели сети.

Задание для практической работы:

1. Выполнить практическую часть
2. Ответить на контрольные вопросы
3. Оформить отчет.

ПРАКТИЧЕСКАЯ ЧАСТЬ

1. Получить у преподавателя вариант задания. Ознакомиться с описанием задания и в NetCracker собрать сеть с заданной топологией и спецификациями.
2. Задать сетевой трафик согласно заданию.

Вывести статистику основных каналов передачи данных. Запустить модель и определить, есть ли перегрузки оборудования или связей. Показать результаты преподавателю или сделать снимок экрана, экспорт сети в JPG-файл, если преподаватель требует письменный отчет.

Общие рекомендации

1. Старайтесь, где это возможно, применять устройства из разделов Generic Devices. Например, компьютеры (LANworkstations→Workstations→Generic devices→Ethernet Workstation), хабы (Hubs→Shared Media→Ethernet→Generic devices →Fast Ethernet Hub), коммутаторы (Switches→Workgroup→Ethernet→Generic devices→Ethernet Switch), маршрутизаторы (Router and Bridges→Backbone→Generic devices→Backbone router).
2. Условные обозначения: хабы (hubs) - см. рис. 7.1 , коммутаторы (switches) - см. рис. 7.2, маршрутизаторы (routers) - см. рис. 7.3.
3. Если в задании требуется оборудование с интерфейсами Gigabit Ethernet (1Gbps), его придется либо создать с помощью Device→Device Factory (см. выше), либо выбрать из пользовательской библиотеки (тулбар Database User), установленной специально для данных лабораторных занятий.
4. Если другое не указано в описании задания или на рисунке, используйте интерфейсы и оборудование Fast Ethernet, сигнальный стандарт 100Base-TX и среду «витая пара».

5. Подразумевается использование значений по умолчанию для статистических характеристик трафиков, определенных во всех готовых профилях LAN peer-to-peer, small InterLAN и других, если в задании не приводятся характеристики этих трафиков или не требуется их изменение, подбор.

ПРИМЕР

Создайте проект сети с топологией и составом оборудования согласно *рис. 7.1*. Задайте трафик с профилем LAN peer-to-peer между всеми рабочими станциями. И клиент-серверный трафик с профилем File server's client от каждой рабочей станции к серверу.

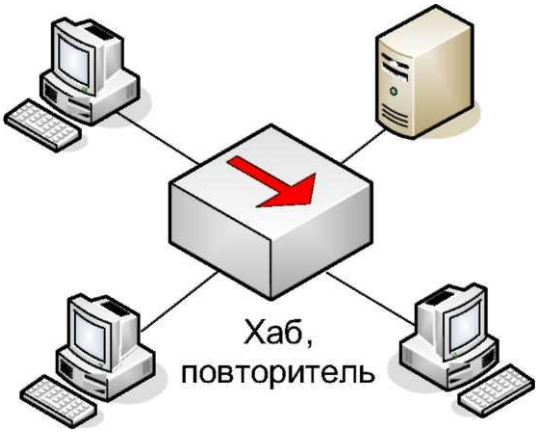


Рис.7.1 «Топология «шина в точке» *(англ. bus-in-a-point)*

1. Добавим необходимое оборудование, следуя общим рекомендациям.
2. Соединим оборудование, задавая необходимые характеристики (рис. 7.4)

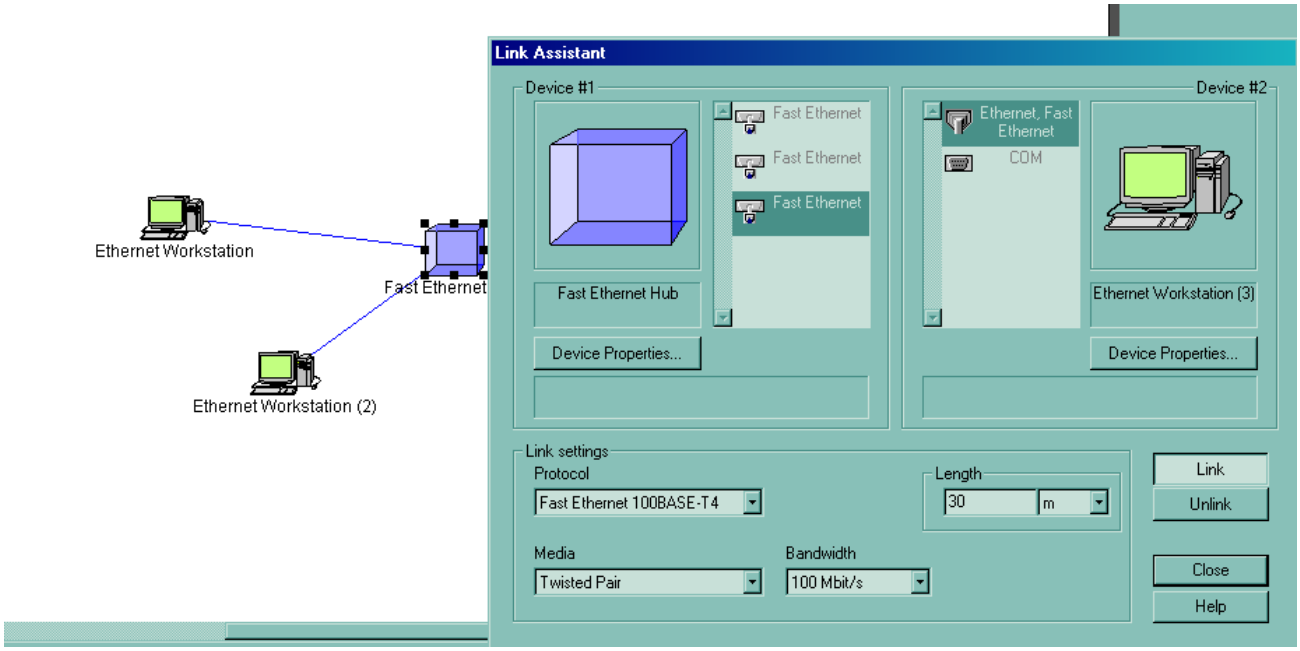


Рис 7.4 «Окно соединения оборудования»

3. Зададим трафик с профилем LAN peer-to-peer между всеми рабочими станциями (рис.7.5)
4. Зададим клиент-серверный трафик с профилем File server's client от каждой рабочей станции к серверу (рис.7.6)

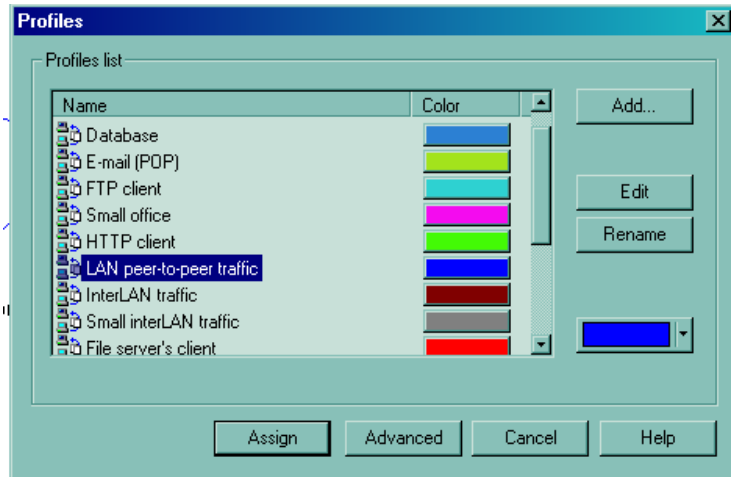
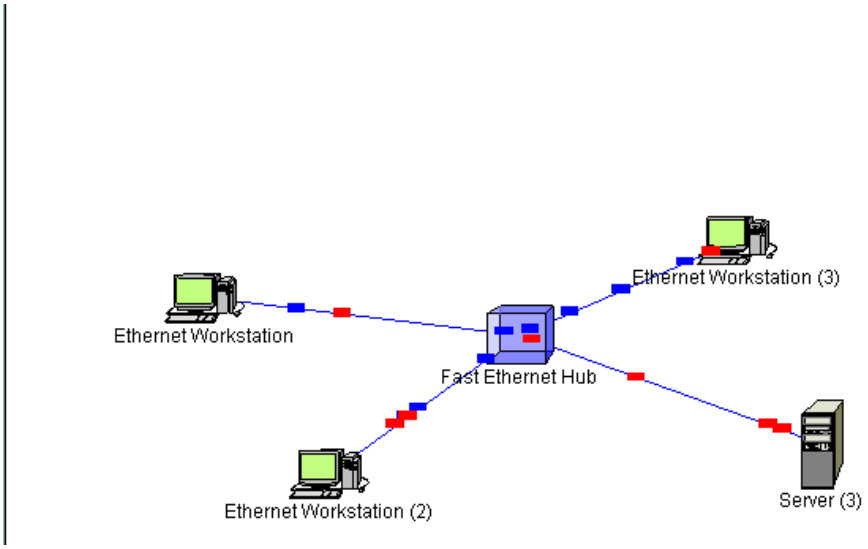


Рис.7.5 «Задание трафика между рабочими станциями»



Задание на практическую работу

Вариант 1. Создайте проект сети с топологией и составом оборудования согласно *Рис. 7.1.* Задайте трафик с профилем LAN peer-to-peer между всеми рабочими станциями. И клиент-серверный трафик с профилем SQL server's client от каждой рабочей станции к серверу. Проанализируйте работу сети.

Вариант 2. Создайте проект сети с топологией и составом оборудования согласно *Рис. 7.1.* Задайте трафик с профилем LAN peer-to-peer между всеми рабочими станциями. И клиент-серверный трафик с профилем FTP client от каждой рабочей станции к серверу. Проанализируйте работу сети.

Вариант 3. Создайте проект сети с топологией и составом оборудования согласно *Рис. 7.1.* Задайте трафик с профилем LAN peer-to-peer между всеми рабочими станциями. И клиент-серверный трафик с профилем HTTP client от каждой рабочей станции к серверу. Проанализируйте работу сети.

Вариант 4. Создайте проект сети с топологией и составом оборудования согласно *Рис. 7.1.* Задайте трафик с профилем LAN peer-to-peer между всеми компьютерами сети. Увеличивая трафик за счет изменения параметра задержки между пакетами «Time between transactions» профиля «LAN peer-to-peer», добейтесь максимально возможной загрузки каналов связи. Запишите полученное значение параметра задержки и процент загрузки каналов. Проанализируйте работу сети.

Вариант 5. Создайте проект сети с топологией и составом оборудования согласно *Рис. 7.2*. Задайте трафик с профилем LAN peer-to-peer между всеми рабочими станциями. И клиент-серверный трафик с профилем File server's client от каждой рабочей станции к серверу. Проанализируйте работу сети.

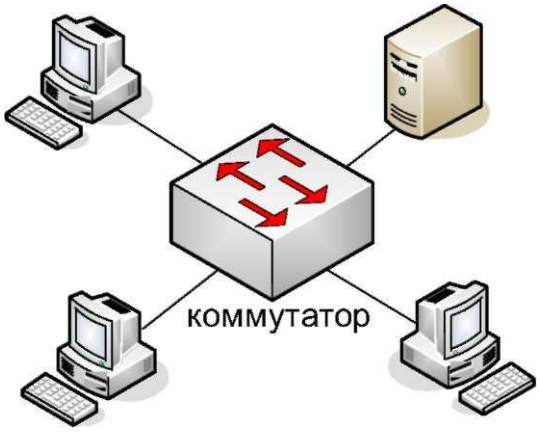


Рис.7.2 «Топология «звезда» (англ. star)»

Вариант 6. Создайте проект сети с топологией и составом оборудования согласно *Рис. 7.2*. Задайте трафик с профилем LAN peer-to-peer между всеми рабочими станциями. И клиент-серверный трафик с профилем SQL server's client от каждой рабочей станции к серверу. Проанализируйте работу сети.

Вариант 7. Создайте проект сети с топологией и составом оборудования согласно *Рис. 7.2*. Задайте трафик с профилем LAN peer-to-peer между всеми рабочими станциями. И клиент-серверный трафик с профилем FTP client от каждой рабочей станции к серверу. Проанализируйте работу сети.

Вариант 8. Создайте проект сети с топологией и составом оборудования согласно *Рис. 7.2*. Задайте трафик с профилем LAN peer-to-peer между всеми рабочими станциями. И клиент-серверный трафик с профилем HTTP client от каждой рабочей станции к серверу. Проанализируйте работу сети.

Вариант 9. Создайте проект сети с топологией и составом оборудования согласно *Рис. 7.2*. Задайте трафик с профилем LAN peer-to-peer между всеми компьютерами сети. Увеличивая трафик за счет изменения параметра задержки между пакетами «Time between transactions» профиля «LAN peer-to-peer», добейтесь максимально возможной загрузки каналов связи. Запишите полученное значение параметра задержки и процент загрузки каналов. Проанализируйте работу сети.

Вариант 10. Создайте проект сети с топологией и составом оборудования согласно *Рис. 7.3*. Задайте трафик с профилем LAN peer-to-peer между всеми рабочими станциями. И клиент-серверный трафик с профилем File server's client от каждой рабочей станции к серверу. Проанализируйте работу сети.

Вариант 11. Создайте проект сети с топологией и составом оборудования согласно *Рис. 7.3*. Задайте трафик с профилем LAN peer-to-peer между всеми рабочими станциями. И клиент-серверный трафик с профилем SQL server's client от каждой рабочей станции к серверу. Проанализируйте работу сети.

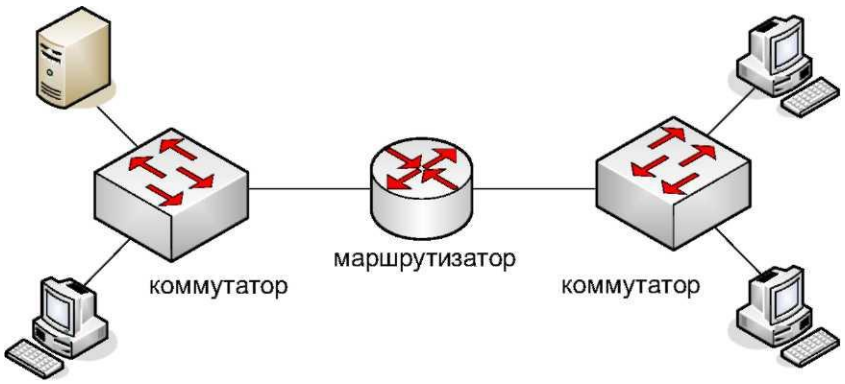


Рис. 7.3 «Иерархическая (неплоская) сеть»

Вариант 12. Создайте проект сети с топологией и составом оборудования согласно *Рис. 7.3*. Задайте трафик с профилем LAN peer-to-peer между всеми рабочими станциями. И клиент-серверный трафик с профилем FTP client от каждой рабочей станции к серверу. Проанализируйте работу сети.

Вариант 13. Создайте проект сети с топологией и составом оборудования согласно *Рис. 7.3*. Задайте трафик с профилем LAN peer-to-peer между всеми рабочими станциями. И клиент-серверный трафик с профилем HTTP client от каждой рабочей станции к серверу. Проанализируйте работу сети.

Вариант 14. Создайте проект сети с топологией и составом оборудования согласно *Рис. 7.3*. Задайте трафик с профилем LAN peer-to-peer между всеми компьютерами сети. Увеличивая трафик за счет изменения параметра задержки между пакетами «Time between transactions» профиля «LAN peer-to-peer», добейтесь максимально возможной загрузки каналов связи. Запишите полученное значение параметра задержки и процент загрузки каналов. Проанализируйте работу сети.

Контрольные вопросы:

1. Опишите процесс задания трафика между рабочими станциями.
2. Как задать клиент-серверный трафик?
3. Для чего служит параметр задержки между пакетами Time between transactions»?
4. Что означает понятие «максимально возможная загрузка каналов связи»?
5. Топология «Звезда»? Ее преимущества и недостатки?

Практическая работа №8

Назначение корректных IP-адресов Расчет масок подсетей. Назначение корректных IP-адресов. Расчет масок подсети

Цель работы: Научиться анализировать правильность выбора масок подсетей и IP-адресов узлов, сетей, широковещания.

В результате выполнения практических заданий обучающийся должен:

Уметь:

- осуществлять правильный выбор масок подсетей IP-адресов узлов, сетей, широковещания;

Знать:

- способы преобразования IP-адреса из двоичного представления в десятичное;
- основы IP-адресации и маршрутизации.

Задание для практической работы:

1. Изучить теоретическую часть
2. Выполнить практическую часть
3. Ответить на контрольные вопросы
4. Оформить отчет.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Одной из наиболее важных тем при обсуждении стека TCP/IP является IP-адресация. *IP-адрес* представляет собой числовой идентификатор, присваиваемый каждому компьютеру сети IP. Он отражает расположение устройства в сети. IP-адрес является программным, а не аппаратным адресом — последний "зашит" в компьютере или плате сетевого интерфейса. IP-адреса позволяют хостам одной сети взаимодействовать с хостами другой сети вне зависимости от типов этих локальных сетей.

Перед подробным изучением IP-адресации нужно усвоить несколько базовых понятий и терминов.

Термины IP-адресации

Byte (байт) 7 или 8 бит, в зависимости от использованной схемы проверки четности. В этой работе мы будем считать, что один байт всегда равен 8 бит.

Octet (октет) - всегда равен 8 бит (разрядам).

Network address (сетевой адрес) - точка назначения, используемая в маршрутизации пакетов к удаленной сети, например, сетевые адреса 10.0.0.0,
172.16.0.0 и 192.168.10.0.

Broadcast address (адрес широковещательной рассылки) - используется приложениями и хостами для пересылки информации всем узлам сети. Примеры адресов широковещательной рассылки: 255.255.255.255 (всем узлам всех сетей), 172.16.255.255 (всем подсетям и хостам сети 17.16.0.0), 10.255.255.255 (широковещательная рассылка всем подсетям и хостам сети 10.0.0.0).

Иерархическая схема IP-адресации

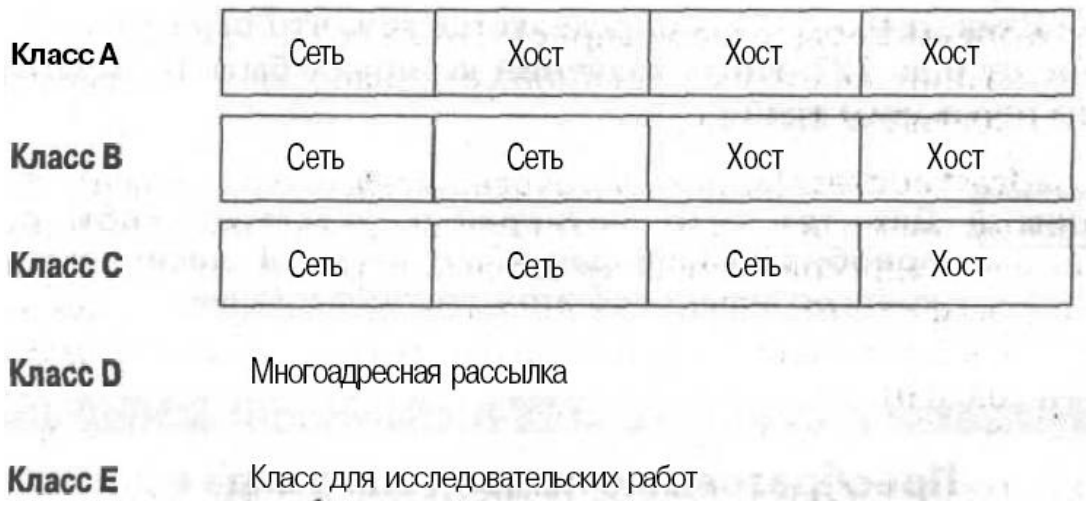
IP-адрес содержит 32 бита информации, которые разделяются на четыре однобайтовые (восьмибитовые) секции, иначе называемые *октетами.* Существуют три способа представления IP-адресов:

- Представление десятичными числами, разделенными точками, например 172.16.30.56
- Двоичное представление, например 10101100.00010000.00011110.00111000
- Шестнадцатеричное представление, например AC 10 IE 38

Здесь показаны три формы представления одного и того же IP-адреса. Шестнадцатеричное представление используется реже, чем двоичное или десятичное, но все же применяется в некоторых программах, например, в реестре Windows IP-адреса компьютеров хранятся в шестнадцатеричном виде.

Для адресации выбрана иерархическая схема с тремя уровнями иерархии: сеть, подсеть и хост.

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



Адресация сетей

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

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

Диапазон сетевых адресов класса A

Создатели схемы IP-адресации установили, что первый бит первого байта сетевого адреса сети класса A всегда выключен (т.е. равен 0). Следовательно, адреса класса A находятся между 0 и 127.

Диапазон сетевых адресов класса B

В сетях класса B спецификация RFC предписывает, что всегда должен быть включен первый бит первого *байта,* однако второй бит должен быть выключен. Если выключить, а затем включить остальные шесть разрядов, то мы получим диапазон для сетей B:

10000000=128

10111111=191

Следовательно, сети класса B имеют в первом байте значения от 128 до 191.

Диапазон сетевых адресов класса C

В сетях класса C спецификация RFC предписывает, что всегда должны быть включены два первых бита первого октета. Найдем диапазон для сети класса C преобразованием из двоичного вида в десятичный:

11000000=192

11011111=223

Следовательно, если начало IP-адреса находится между 192 и 223, то это адрес сети класса C.

Диапазоны сетевых адресов классов D и E

Адреса в диапазоне между 224 и 255 зарезервированы для сетей классов D и E. Класс D используется для многоадресных рассылок, а класс E — для исследовательских разработок. Далее мы не будем возвращаться к этим классам адресов.

Диапазоны сетевых адресов для специального применения

Некоторые IP-адреса зарезервированы для специальных целей и сетевые администраторы не могут присвоить их узлам своих сетей.

Зарезервированные IP-адреса

Адрес	Функция
Сетевой адрес из всех нулей	Означает "эта сеть или сегмент".
Сетевой адрес из всех единиц	Означает "все сети".
Сеть 127.0.0.1	Зарезервирована для кольцевого тестирования. Предназначена для сетевого узла, который может послать пакет себе без генерации сетевого трафика.
Адрес узла из всех нулей	Означает "этот узел".
Адрес узла из всех единиц	Означает "все узлы" определенной сети, например 128.2.255.255 показывает "все узлы сети 128.2 (адреса класса B)".
Весь IP-адрес из нулей	Используется маршрутизаторами Cisco для указания пути по умолчанию.
Весь IP-адрес из единиц (255.255.255.255)	Широковещательная рассылка по всем узлам текущей сети, иногда называется "широковещательной рассылкой по всем единицам".

Адреса класса A

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

Например, в IP-адресе 49.22.102.70 адрес сети равен 49, а адрес узла — 22.102.70. Каждая машина этой сети должна иметь адрес сети, равный 49. Адрес сети класса A имеет длину 1 байт, причем его первый бит зарезервирован, но доступны оставшиеся семь разрядов. Это означает, что можно создать не более 128 сетей класса A. Почему? Потому что

каждый из семи оставшихся битов может принимать значение 0 или 1, т.е. существует 27 или 128 различных комбинаций. Однако, было решено, что нулевой адрес сети (0000 0000) резервируется для обозначения маршрута, выбранного по умолчанию. Однако из-за того, что нулевой адрес зарезервирован, диапазон становится уже: от 1 до 127. В результате реальное число сетей класса А равно 128-2, т.е. 126.

Под адрес узла в IP-адресе сетей класса А отведено 3 байта (24 разряда). В них можно разместить 16777216 различных двоичных комбинаций или адресов узлов. Поскольку адреса, состоящие только из нулей и только из единиц, зарезервированы, точное число узлов в сети класса А составляет 16777216 - 2=16777214.

Допустимые значения идентификаторов хостов в сети класса А

Рассмотрим пример определения допустимого идентификатора хоста для сетевого адреса класса А:

10.0.0.0 В сетевом адресе выключены все разряды, определяющие идентификатор хоста.

10.255.255.255 Все разряды для хостов в широковещательном адресе.

Допустимое количество хостов находится в диапазоне между сетевым адресом и адресом широковещательной рассылки: от 10.0.0.1 до 10.255.255.254. Заметим, что допустимы идентификаторы хостов из всех нулей и 255. Для подсчета количества доступных адресов хостов нужно, помнить, что разряды хоста не могут быть все вместе включены или выключены.

Адреса класса В

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

Сеть. Сеть.Узел.Узел

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

Для адреса сети, состоящего из 16 разрядов, имеется 216 возможных комбинаций. Однако разработчики Интернета решили, что адрес сети класса В должен начинаться с комбинации 10. Поэтому свободными для формирования адреса остаются лишь 14 бит; это означает, что может существовать 214 или 16 384 сетей класса В.

Под адрес узла в IP-адресе сетей класса В отведено 2 байта. Поскольку адреса, состоящие только из нулей и только из единиц, зарезервированы, точное число узлов в сети класса В равно 216 - 2=65 534.

Допустимые значения идентификаторов хостов в сети класса В

Рассмотрим пример определения допустимого идентификатора хоста для сетевого адреса класса В:

172.16.0.0 В сетевом адресе выключены все разряды, определяющие идентификатор хоста.

172.16.255.255 Все разряды для хостов в широковещательном адресе.

Допустимое количество хостов находится в диапазоне между сетевым адресом и адресом широковещательной рассылки: от 172.16.0.1 до 172.16.255.254.

Адреса класса С

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

Сеть.Сеть.Сеть.Узел

Например, в IP-адресе 192.168.100.102 адрес сети равен 192.168.100, а адрес узла — 102.

Первые три разряда адреса сети класса С занимает комбинация 110. Поэтому для формирования адреса остается лишь 24 - 3= 21 разряд. Таким образом, может существовать 221 или 2 097 152 сетей класса С.

Под адрес узла в IP-адресе сетей класса С отведен 1 байт. Следовательно, в каждой сети класса С может быть 28 - 2=254 узла.

Допустимые значения идентификаторов хостов в сети класса С

Рассмотрим пример определения допустимого идентификатора хоста для сетевого адреса класса С:

192.168.100.0 В сетевом адресе выключены все разряды, определяющие идентификатор хоста.

192.168.100.255 Все разряды для хостов в широковещательном адресу.

Допустимое количество хостов находится в диапазоне между сетевым адресом и адресом широковещательной рассылки: от 192.168.100.1 до 192.168.100.254.

Маска подсети

При применении схемы адресации с подсетями каждая машина сети должна знать, какая часть адреса хоста занята адресом подсети. Для этого на каждом компьютере создается *маска подсети.* Это 32-разрядное число, которое позволяет получателю пакета IP отделить идентификатор сети в IP-адресе от идентификатора хоста.

Администратор сети создает 32-разрядную маску подсети, состоящую из 0 и 1. Единицы в маске подсети помечают позиции, относящиеся к адресам сети и подсети.

Не во всех сетях нужны подсети, т.е. иногда используются маски подсети по умолчанию (иными словами, в такой сети нет адресов подсетей).

Маски подсетей по умолчанию

Класс	Формат	Маска по умолчанию
A	Узел.Узел.Узел.Узел	255.0.0.0
B	Сеть.Сеть.Узел.Узел	255.255.0.0
C	Сеть.Сеть.Сеть.Узел	255.255.255.0

Выделение подсетей в классе С

Существуют разные способы выделения подсетей, среди которых можно выбрать наиболее подходящий для себя. Сначала мы обсудим двоичный метод, а затем познакомимся с другим способом выделения подсетей.

В адресном пространстве класса С для определения хостов доступны только 8 разрядов. Биты подсети отсчитываются слева направо без пропусков разрядов. Масками подсетей могут быть:

10000000=128
11000000=192
11100000=224
11110000=240
11111000=248
11111100=252
11111110=254

Спецификация RFC не разрешает использовать для подсетей только один разряд, поскольку он всегда будет либо включен, либо выключен, а это недопустимо. Следовательно, первой правильной маской подсети будет 192, а последней — 252, поскольку нужно не менее двух разрядов для указания хостов.

Двоичный метод: Выделение подсетей в классе С

Рассмотрим выделение подсетей в адресном пространстве класса С с помощью двоичного метода. Сначала следует выявить первую доступную маску подсети, которая заимствует два разряда. Например, можно использовать 255.255.255.192.

192=11000000

Два разряда применяются для выделения подсетей, 6 разрядов определяют хосты в каждой подсети. Какими будут подсети? Поскольку разряды подсети не могут быть одновременно включены или выключены, допустимы только две подсети:

01000000=64 (все разряды хостов выключены)

или

10000000=128 (все разряды хостов выключены)

Корректные адреса хостов находятся между подсетями, за исключением вариантов, когда одновременно включены или выключены все разряды хостов.

Для выявления адресов хостов нужно сначала выключить все разряды хостов в адресе, а затем включить их, чтобы найти широковещательный адрес подсети. Допустимые адреса хостов располагаются между двумя полученными адресами.

В таблице ниже показана подсеть 64, диапазон хостов и адрес широковещательной рассылки.

Подсеть 64

Подсеть	Хост	Описание
01	000000=64	Сеть (первая операция)
01	000001=65	Первый допустимый хост
01	111110=126	Последний допустимый хост
01	111111=127	Широковещательный адрес (вторая операция)

В таблице ниже показана подсеть 128, диапазон хостов и адрес широковещательной рассылки.

Подсеть 128

Подсеть	**Хост**	**Описание**
10	000000=128	Адрес подсети
10	000001=129	Первый допустимый хост
10	111110=190	Последний допустимый хост
10	111111=191	Широковещательный адрес

Операция проста, но в наших примерах рассмотрен только случай с двумя разрядами для подсети. Что делать, когда нужно 9, 10 или даже 20 разрядов? Рассмотрим альтернативный метод, пригодный для выделения большого количества подсетей.

Альтернативный метод:

Выделение подсетей в классе С

Установив маску подсети, следует определить количество подсетей, хостов и широковещательные адреса. Для этого нужно ответить на несколько простых вопросов:

1. Сколько подсетей формирует данная маска?
2. Сколько хостов будет в каждой подсети?
3. Каковы правильные подсети?
4. Каковы правильные хосты в каждой подсети?
5. Какие широковещательные адреса в подсетях?

Приведем примеры ответов на поставленные вопросы:

1. Сколько подсетей? 2x-2 = количество_подсетей, где X равно количеству маскируемых разрядов (т.е. единиц). Например, для 11000000 мы имеем 22 - 2, т.е. 2 подсети.

2. Сколько хостов в подсетях? 2x-2 = количество_хостов_в_подсети, где X равно количеству немаскируемых разрядов (т.е. нулей). Например, для 11000000 мы имеем 26 - 2, т.е. 62 хоста в подсети.

3. Каковы корректные подсети? 256-маска_подсети = базовое_количество. Например, 256 - 192 = 64.

4. Каковы корректные хосты? Количество хостов равно разности между подсетями, минус "все нули" и "все единицы".

5. Каков широковещательный адрес в каждой подсети? Адрес широковещательной рассылки получается после включения всех разрядовхостов, поэтому легко вычисляется для любой подсети.

Примеры выделения подсетей в классе С

Рассмотрим несколько примеров выделения подсетей в классе С с помощью рассмотренных выше методов.

Пример 1: 255.255.255.192

Начнем с адреса подсети в классе С, который использовался в предыдущем примере (255.255.255.192), чтобы показать преимущество альтернативного метода над двоичным. В этом примере мы используем сетевой адрес 192.168.10.0 и маску подсети 255.255.255.192.

192.168.10.0=Сетевой адрес

255.255.255.192=Маска подсети

Не трудно получить ответы на пять основных вопросов:

1. Сколько подсетей? В 192 включены два разряда (11000000), поэтому 22- 2=2. (вычитание 2 связано с некорректными по определению адресами, в которых включены или выключены все разряды подсети).

2. Сколько хостов в подсети? Выключено 6 разрядов хоста (11000000), следовательно, 26 - 2=62 хоста.

3. Какова правильная подсеть? 256-192=64 и мы получаем первую подсеть, а также базовое количество (переменную). Далее следует складывать эту переменную до тех пор, пока не будет достигнута маска подсети. 64+64=128. 128+64=192, но это уже некорректная маска, поскольку в ней включены все разряды подсети. Итак, получаем две подсети: 64 и 128.

4. Каковы правильные хосты? Они находятся между подсетями. Проще всего выявить их адреса, записав адреса подсетей и адреса широковещательных рассылок.

5. Какие широковещательные адреса в подсетях? Это число находится перед следующей подсетью и имеет включенными все биты хостов.

В таблице ниже показаны подсети 64 и 128, диапазон хостов в каждой из них и широковещательные адреса в каждой подсети.

Диапазоны подсетей 64 и 128

Первая подсеть	Вторая подсеть	Описание
64	128	Подсеть (первая операция)
65	129	Первый хост (адреса хостов вычисляются позже)
126	190	Последний хост
127	191	Широковещательный адрес (вторая операция)

Мы получили те же ответы, что и в двоичном методе, но нам уже не пришлось прибегать к преобразованию числа из двоичного вида в десятичный. Однако этот метод не всегда будет проще двоичного. Для первой подсети, где только два разряда подсети, двоичный метод будет удобнее. Возможно, следует хорошо изучить оба метода, поскольку часто приходится выполнять вычисления о подсетях в уме.

Остальные примеры вычисления масок можно найти в литературе:

1) **CCNA** Cisco Certified Network Associate Учебное руководство, Экзамен 640-507, Тодд Леммл, Издательство "Лори", 2002 г.

Задание к практической работе:

1. Классификация IP-адресов.

Перевести число из двоичной системы в десятичную.

Перевести число из десятичной системы в двоичную.

Представить IP-адреса в двоичном формате и определить класс сети.

2. Разбиение сети на подсети

Дана сеть класса B. Необходимо ее разбить на 8 подсетей.

Определить маску каждой из подсетей

Определить номера подсетей

Определить число хостов в каждой из подсетей. Привести примеры IP-адресов хостов во всех подсетях и привести диапазон IP-адресов хостов.

3. Дана сеть класса C. Определить префикс сети, который позволит создать N хостов в каждой подсети.

3.1 Какое число компьютеров можно подключить к каждой подсети?

3.2 Какое максимальное число подсетей может быть определено?

3.3 Привести номера подсетей в двоичном формате и точечной нотации.

3.4 Привести пример IP-адресов хостов в подсети номер M. Привести диапазон IP-адресов в этой подсети.

3.5 Для подсети M определить широковещательный адрес. Привести его в десятичном и двоичном формате.

Варианты заданий см. в таблице ниже

Таблица 8.1 «Варианты к заданиям»

Вар	**Пункт 1.1**	**Пункт 1.2**	**Пункт 1.3**	**Задание 2**	**Задание 3**		
					IP	**N**	**M**
1	01100110, 10111001, 11100111,	165, 254, 23, 56	127.0.1.2, 198.45.238.38, 45.218.75.1	136.56.0.0	196.56.4.0	17	2

	00111011						
2	01011101, 11110010, 00110110, 10011101	24, 156, 89, 246	156.23.65.2, 24.67.149.16, 62.48.179.23	145.78.0.0	210.234.6.0	20	6
3	10011010, 00110110, 10011011, 01111000	254, 125, 23, 156	13.15.56.16, 165.48.14.98, 78.245.11,23	186.5.0.0	208.25.198.0	9	8
4	01101111, 01110100, 00110011, 01101111	248, 26, 89, 183	202.11.23.7, 49.10.22.98, 109.252.26.23	173.98.0.0	194.168.23.0	23	7
5	10111011, 11101101, 01101111, 01011011	35, 81, 193, 46	187.23.65.1, 26.23.26.4, 69.136.32.14	129.37.0.0	199.242.3.0	31	4
6	01101000, 10011011, 01110011, 00111011	149, 167, 23, 49	54.23.65.4, 195.26.156.5, 127.0.0.1	181.64.0.0	193.25.165.0	12	2
7	01101111, 01011101, 01111111, 11111011	45, 64, 121, 221	200.25.121.1, 126.2.23.1, 36.1.46.5	156.23.0.0	205.32.57.0	6	4
8	10111011, 01110111, 01011101, 10010011	158, 172, 45, 250	46.56.66.76, 189.12.136.1, 56.11.46.14,	162.28.0.0	201.34.26.0	18	1
9	01110111, 10011101, 11100110, 10111011	188, 165, 149, 13	38.46.16.16, 159.16.0.4, 168.197.12.3	176.2.0.0	200.234.59.0	28	5
10	10001011, 01101110, 01111111, 01001101	154, 198, 67, 59	86.16.4.3, 74.23.49.1, 136.15.48.1	189.37.0.0	195.65.23.0	22	6
11	01101011, 01011011, 01110010, 10011101	56, 165, 89, 143	194.168.1.3, 65.111.166.1, 24.1.49.1	164.168.0.0	197.148.6.0	14	7
12	10100101, 01011001, 10011011, 10111101	226, 167, 165, 8	33.48.19.16, 126.16.19.4, 176.16.48.3	138.195.0.0	198.29.163.0	7	5

Контрольные вопросы:

1. Что такое IP-адрес?
2. Из каких частей состоит IP-адрес?
3. В каких форматах возможна запись IP-адреса?
4. Для чего предназначены IP-адреса класса A? B? C? D? E?
5. Что такое широковещательный адрес?
6. Для чего используются маски подсети?

Практическая работа №9

Средства сетевой диагностики

Цель работы: Изучить основные средства сетевой диагностики в операционной системе Ubuntu.

В результате выполнения практических заданий обучающийся должен:

Уметь:

- Осуществлять сетевую диагностики в операционной системе Ubuntu.

Знать:

- Основные средства сетевой диагностики в операционной системе Ubuntu.

Задание для практической работы:

1. Изучить теоретическую часть
2. Выполнить практическую часть
3. Ответить на контрольные вопросы
4. Оформить отчет.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Сетевые утилиты

В операционной системе Linux существует большое число утилит (специальных программ), предназначенных для управления и анализа сетевых соединений, рассмотрим три из них: IFCONFIG, ARP, NETSTAT.

Утилита ifconfig

Команда ifconfig используется для подключения и отключения сетевого интерфейса, а также задания его адреса, маски подсети, других опций и параметров. Она обычно выполняется на этапе начальной загрузки, но может применяться и для внесения изменений в работающую систему.

В большинстве случаев команда ifconfig имеет следующий формат:

ifconfig интерфейс адрес опции….

Например:

ifconfig eth0 192.168.1.13 netmask 255.255.255.0 broadcast 192.168.1.255 up

Параметр *интерфейс* обозначает аппаратный интерфейс, к которому применяется команда. В *nix системах это обычно двух- или трехсимвольное устройство, за которым следует число это почти всегда имя вида eth0,eth1 и т.д.

Команда ifconfig интерфейс отображает текущие настройки указанного интерфейса.

С помощью команды ifconfig –a можно узнать, какие интерфейсы присутствуют в системе.

Параметр *адрес* задает IP-адрес интерфейса. Ключевое слово up указывает на активизацию интерфейса, а ключевое слово down на его отключение.

Опция netmask задает маску подсети для данного интерфейса. Эта опция обязательна.

Опция broadcast задает широковещательный IP-адрес интерфейса в шестнадцатеричной или точечной записи.

Команда route: конфигурирование статических маршрутов

Маршрут определяет начальную точку процесса передачи пакета и показывает, какому компьютеру ваша система должна передать пакет, чтобы он достиг пункта назначения. В небольших сетях маршрутизация может осуществляться статически, т.е. маршрут, ведущий от одной системы к другой, строго фиксирован. В более крупных сетях и

в сети Internet маршрутизация осуществляется динамически. Ваша система знает, какому компьютеру пакет должен быть послан вначале. Этот компьютер принимает пакет и передает его другому компьютеру, который определяет, куда следует передать пакет дальше.

Маршруты содержатся в таблице маршрутизации, которая хранится в файле /proc/net/route. Чтобы вывести ее на дисплей, нужно дать команду route без аргументов.

Каждая запись таблицы маршрутизации состоит из нескольких полей, содержащих такую информацию, как, например, конечный пункт маршрута и тип используемого интерфейса. Поля таблицы маршрутизации сведены в следующую таблицу.

В таблице маршрутизации должна содержаться по крайней мере одна запись, предназначенная для закольцовывающего интерфейса, иначе этот интерфейс необходимо сконфигурировать командой route. IP-адрес интерфейса нужно ввести в таблицу до того, как этот интерфейс будет задействован. Адрес добавляется с помощью команды route с опцией add:

route add адрес

Опция add имеет несколько спецификаторов (можно узнать в man-страницах). Если вы добавляете конкретный статический маршрут, то эти спецификаторы понадобятся для ввода таких параметров, как маска сети, шлюз, интерфейс и адрес пункта назначения. Если же интерфейс уже конфигурирован командой ifconfig, то система может получить основную информацию из данных конфигурации интерфейса. Например, чтобы задать маршрут для Ethernet-соединения, которое уже конфигурировано командой ifconfig, нужно лишь ввести спецификатор -net и IP-адрес пункта назначения. С помощью этого адреса ifconfig находит соответствующий интерфейс и на основании этой информации организует маршрут. Задание маршрута для интерфейса Ethernet выглядит так:

route add -net 196.162.0.0

Если система подключена к сети, в таблице маршрутизации должна быть сделана, по крайней мере, одна запись, задающая маршрут по умолчанию. По этому маршруту пакет посылается в том случае, если все остальные маршруты не могут привести его в пункт назначения. Пункт назначения для такого маршрута задается ключевым словом default.

Если нужно удалить один из существующих маршрутов, следует вызвать команду ifconfig с опцией del и IP-адресом маршрута, например:

route del -net 196.162.0.0

Настройка DNS

Чтобы сконфигурировать компьютер в качестве DNS-клиента, достаточно отредактировать файл /etc/resolv.conf.

Файл /etc/resolv.conf содержит список IP-адресов DNS-серверов, в котором осуществляется поиск имен.

Ниже показан пример одной записи из файла /etc/resolv.conf

nameserver 128.138.242.1

Всего можно задать три записи nameserver и желательно указывать более одного сервера.

Сетевое конфигурирование в системе Ubuntu

Сетевые параметры в системе Ubuntu сосредоточены в основном в файлах /etc/hostname и /etc/network/interfaces

Сетевое имя компьютера задается в файле /etc/hostname/.

IP-адрес, маска подсети и стандартный шлюз задаются в файле /etc/network/interfaces. Базовый формат записей файла таков: для каждого интерфейса указывается строка, начинающаяся с ключевого слова iface, а за ней следуют строки с отступом, в которых указаны дополнительные параметры. Например:

iface eth0 inet static
address 192.168.0.1
netmask 255.255.255.0
gateway 192.168.0.254

Ключевое слово static обозначает способ описание интерфейса и указывают на то, что IP-адрес и сетевая маска интерфейса eth0 будут задаваться непосредственно.

Команду ifconfig следует первой использовать для диагностирования возможных проблем с соединением TCP/IP. С ее помощью можно определить, был ли вообще назначен IP-адрес сетевому адаптеру, а также узнать адрес шлюза.

Утилита NETSTAT

Команда позволяет получить подробную информацию о соединениях, активных в настоящее время. Дополнительные ключи позволяют также получить информацию о сетевых портах, об IP-адресах компьютеров, участвующих в подключении, а также о других сетевых параметрах.

Параметры:

-a

Вывод всех активных подключений TCP и прослушиваемых компьютером портов TCP и UDP (рис.).

-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 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.. Параметр **-p** может использоваться для указания набора протоколов.

-r

Вывод содержимого таблицы маршрутизации IP

Утилита ARP

Служит для вывода и изменения записей кэша протокола ARP, который содержит одну или несколько таблиц, использующихся для хранения IP-адресов и соответствующих им физических адресов Ethernet. Для каждого сетевого адаптера Ethernet, установленного в компьютере, используется отдельная таблица. Запущенная без параметров, команда arp выводит справку.

Параметры

-a

Вывод таблиц текущего протокола ARP для всех интерфейсов

Чтобы вывести записи ARP для определенного IP-адреса, следует указать его после ключа через пробел:

arp -a *IP-адрес*

-g -Выполняет те же функции, что и **-a.**

-d *IP-адрес* [*иф_адрес*]

Выполняет удаление записи с определенным IP-адресом. Чтобы удалить запись таблицы для определенного интерфейса, следует указать этот интерфейс после IP-адреса. Чтобы удалить все записи, нужно ввести звездочку (*) вместо параметра *IP-адрес*

-s *IP-адрес Ethernet_адрес* [*иф_адрес*]

Добавление статической записи, которая сопоставляет IP-адрес с физическим адресом в кэш ARP. Чтобы добавить статическую запись кэша ARP в таблицу

Команда PING

Команда PING является едва ли не самой используемой в локальных сетях командой. Она позволяет тестировать сетевое соединение, получая информацию о различных его аспектах. Неудачная попытка соединения с каким-либо компьютером, или ошибка получения доступа к общим файлам и папкам, находящимся на других компьютерах локальной сети, может быть вызвана тем, что другие компьютеры просто не получают отправленных им по сети запросов.

После введения в командной строке имени команды, в качестве параметра для нее, указывается адрес, по которому будут направляться специальные эхо-пакеты, это может быть IP-адрес, или символьное имя компьютера.

Получив эхо-запрос, удаленный компьютер сразу же отправляет его обратно по тому адресу, откуда он пришел, команда ping позволяет узнать, пришли ли обратно посланные запросы, проверяя, таким образом, не только целостность физической среды передачи данных, но и корректную обработку информации на всех остальных семи уровнях модели OSI.

```
ping -c 3 www.ya.ru
PING www.ya.ru (88.204.75.138) 56(84) bytes of data.
64 bytes from portal.tusur.ru (88.204.75.138): icmp_seq=1 ttl=60 time=0.860 ms
64 bytes from portal.tusur.ru (88.204.75.138): icmp_seq=2 ttl=60 time=0.909 ms
64 bytes from portal.tusur.ru (88.204.75.138): icmp_seq=3 ttl=60 time=0.873 ms
--- www.ya.ru ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.860/0.880/0.909/0.040 ms
```

При успешном возвращении запросов можно быть уверенным в том, что среда

передачи данных, программное обеспечение TCP/IP, а также все устройства (маршрутизаторы, повторители и др.), встретившиеся на пути между двумя компьютерами, работают нормально.

Необходимо отметить, что даже при отсутствии каких-либо неисправностей на пути между двумя компьютерами, один или сразу несколько пакетов могут быть утеряны, как правило, это бывает в случае перегруженности сети, а также с тем, что диагностирующие пакеты имеют очень низкий приоритет и могут быть отброшены в процессе передачи. Если хотя бы один из посланных пакетов вернется, это уже будет означать исправность работы сети.

По умолчанию размер эхо-пакета составляет 32 байта, по указанному адресу направляются эхо-пакеты и после выполнения команды вы-водится статистика прохождения эхо-пакетов по сети

Команда TRACEROUTE

Эта команда подобна команде PING, обе посылают в точку назначения эхо-пакеты и затем ожидают их возвращения. Отличие пакетов команды TRACEROUTE от пакетов PING заключается в том, что они имеют различный срок жизни (Time to Live, TTL). Каждый маршрутизатор при прохождении через него пакета уменьшает значение поля TTL в нем на единицу. Первые пакеты, отправляемые командой TRACEROUTE имеют TTL=1, поэтому первый маршрутизатор, получив такой пакет и уменьшив на единицу поле TTL, обнаруживает, что пакет не может быть доставлен по адресу (пакет с TTL=0 не передается маршрутизатором) и возвращает сообщение об ошибке, содержащее IP-адрес маршрутизатора.

Получив это сообщение, команда выводит на экран информацию об IP-адресе маршрутизатора и отправляет по прежнему адресу эхо-пакет с TTL=2. Количество маршрутизаторов, через которые может пройти пакет, будет каждый раз увеличиваться на единицу до тех пор, пока пакет не достигнет точки назначения.

Таким образом, с помощью команды traceroute можно получить подробный маршрут прохождения пакетов данных между компьютером, на котором была запущена traceroute, и любым удаленным компьютером сети.

Это делает traceroute весьма ценным средством обнаружения неисправностей в сетевом соединении: в случае возникновения проблемы с подключением к Web-узлу или к какой-нибудь другой службе Internet можно определить участок, на котором она возникла.

```
traceroute 8.8.8.8
 1    2 ms    1 ms    1 ms  10.22.66.1
 2    2 ms    2 ms    2 ms  192.168.100.29
 3    2 ms    3 ms    2 ms  212.1.241.53
 4    2 ms    2 ms    2 ms  212.1.240.77
 5   16 ms   16 ms   16 ms  212.1.239.21
 6   33 ms   16 ms   16 ms  72.14.212.158
 7   38 ms   39 ms   39 ms  72.14.236.248
 8   47 ms   46 ms   46 ms  209.85.249.40
 9   47 ms   46 ms   46 ms  72.14.233.168
10    *       *       *     Превышен интервал ожидания для запроса.
11   47 ms   47 ms   47 ms  8.8.8.8
```

Утилита NSLOOKUP

Утилита nslookup (англ. name server lookup поиск на сервере имён) — утилита, предоставляющая пользователю интерфейс командной строки для обращения к системе DNS (проще говоря, DNS-клиент). Позволяет задавать различные типы запросов и запрашивать произвольно указываемые сервера.

```
nslookup  ya.ru
Server:         212.192.122.17
Address:        212.192.122.17#53
Name: ya.ru
Address:  94.100.191.245
```

Задание на практическую работу

1. Задайте вашему компьютеру имя хоста
2. Выведите информацию обо всех интерфейсах компьютера
3. Узнайте MAC-адрес компьютера.
4. Запишите полученную информацию в отчет, заполнив таблицу

№ п.п.	Наименование данных	Содержимое данных
1	имя узла **TCP/IP**	
3	**MAC**-адрес сетевой платы	
4	IP-адрес	
5	маска подсети	
6	шлюз по умолчанию	

5. Найдите в Интернете один из публичных DNS-серверов и пропишите его вашему компьютеру.
6. Используя утилиту ifconfig добавьте на существующий интерфейс адрес вида 10.22.0.1+N, где N-номер варианта. Маска подсети должна быть 255.255.255.0
7. Поменяйте маршрут по умолчанию на 10.22.0.1
8. Отключите сетевой интерфейс вашего компьютера. Затем включите его.
9. Заполните файл /etc/network/interfaces согласно приведенным выше настройкам. Перегрузите компьютер и проверьте применились ли настройки.
10. Отредактируйте заново файл /etc/network/interfaces на автоматическое получение сетевых настроек.
11. Выведите таблицу маршрутизации
12. Выведите таблицу arp-кэша
13. С помощью утилиты PING протестировать соединения с различными серверами в Интернете. Указать не менее 5 различных узлов.
14. С помощью утилиты TRACEROUTE протестировать соединения с различными серверами в Интернете. Указать не менее 5 различных узлов.
15. С помощью утилиты NSLOOKUP определить IP-адреса нескольких интернет ресурсов. Не менее 5 интернет ресурсов.

Контрольные вопросы

1. Для чего предназначена утилита ipconfig?
2. Для чего предназначена утилита traceroute?
3. Для чего предназначена утилита nslookup?
4. Какой формат имеет команда ifconfig?
5. Утилита netstat и ее параметры.

Практическая работа №10

Работа с сетевым экраном netfilter/iptables. Таблица filter

Цель работы: Получить представление о настройке сетевых фильтров (файерволлов, брандмауэров) и освоить простейшие настройки Iptables.

В результате выполнения практической работы студент должен:

Уметь:

- производить настройку сетевых фильтров;
- производить простейшие настройки iptables.

Знать:

- порядок построения собственных правил для iptables;
- действия и переходы, используемых при построении правил.
- критерии выделения пакетов.

Задание для практической работы:

1. Изучить теоретическую часть
2. Выполнить практическую часть
3. Ответить на контрольные вопросы
4. Оформить отчет.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Netfilter — межсетевой экран (брандмауэр), встроен в ядро Linux версий 2.4 и 2.6.

iptables — название пользовательской утилиты (запускаемой из командной строки) предназначенной для управления системой *netfilter*.

В системе *netfilter*, пакеты пропускаются через **цепочки**. Цепочка является упорядоченным списком **правил**, а каждое правило может содержать **критерии** и **действие** или **переход**. Когда пакет проходит через цепочку, система *netfilter* по очереди проверяет, соответствует ли пакет всем критериям очередного правила, и если так, то выполняет действие (если критериев в правиле нет, то действие выполняется для всех пакетов проходящих через правило). Вариантов возможных критериев очень много

Порядок прохождения таблиц и цепочек

Когда пакет приходит на брандмауэр, то он сперва попадает на сетевое устройство, перехватывается соответствующим драйвером и далее передается в ядро. Далее пакет проходит ряд таблиц и затем передается либо локальному приложению, либо переправляется на другую машину. Порядок следования пакета приводится ниже.

Таблица 10.1 «Порядок движения транзитных пакетов»

Шаг	Таблица	Цепочка	Примечание
1			Кабель (то есть Интернет)
2			Сетевой интерфейс (например, eth0)
3	Mangle	PREROUTING	Обычно эта цепочка используется для внесения изменений в заголовок пакета, например для изменения битов TOS и пр..
4	Nat	PREROUTING	Эта цепочка используется для трансляции сетевых адресов (Destination Network Address Translation). Source Network Address Translation выполняется позднее, в другой цепочке. Любого рода фильтрация в этой цепочке

Шаг	Таблица	Цепочка	Примечание
			может производиться только в исключительных случаях
5			Принятие решения о дальнейшей маршрутизации, то есть в этой точке решается куда пойдет пакет - локальному приложению или на другой узел сети.
6	Filter	FORWARD	В цепочку FORWARD попадают только те пакеты, которые идут на другой хост Вся фильтрация транзитного трафика должна выполняться здесь. Не забывайте, что через эту цепочку проходит трафик в обоих направлениях, обязательно учитывайте это обстоятельство при написании правил фильтрации.
7	Mangle	FORWARD	Далее пакет попадает в цепочку FORWARD таблицы mangle, которая должна использоваться только в исключительных случаях, когда необходимо внести некоторые изменения в заголовок пакета между двумя точками принятия решения о маршрутизации.
8			Принятие решения о дальнейшей маршрутизации, то есть в этой точке, к примеру, решается на какой интерфейс пойдет пакет.
9	Nat	POSTROUTING	Эта цепочка предназначена в первую очередь для Source Network Address Translation. Не используйте ее для фильтрации без особой на то необходимости. Здесь же выполняется и маскировка (Masquerading).
10	Mangle	POSTROUTING	Эта цепочка предназначена для внесения изменений в заголовок пакета уже после того как принято последнее решение о маршрутизации.
11			Выходной сетевой интерфейс (например, eth1).
12			Кабель (пусть будет LAN).

Из данной таблицы видно, что пакет проходит несколько этапов, прежде чем он будет передан далее. На каждом из них пакет может быть остановлен, будь то цепочка iptables, или что либо еще, но наибольший интерес представляет iptables. Следует заметить, что нет, каких либо цепочек, специфичных для отдельных интерфейсов или чего либо подобного. Цепочку *FORWARD* проходят ВСЕ пакеты, которые движутся через данный брандмауэр/маршрутизатор. Не следует использовать цепочку *INPUT* для фильтрации транзитных пакетов, так как они туда просто не попадают. Через эту цепочку движутся только те пакеты, которые предназначены данной машине.

А теперь рассмотрим порядок движения пакета, предназначенного локальному процессу/приложению.

Таблица 10.2 «Для локального приложения»

Шаг	Таблица	Цепочка	Примечание
1			Кабель (то есть Интернет)
2			Входной сетевой интерфейс (например, eth0)
3	Mangle	PREROUTING	Обычно используется для внесения изменений в заголовок

Шаг	Таблица	Цепочка	Примечание
			пакета, например для установки битов TOS и пр.
4	Nat	PREROUTING	Преобразование адресов (Destination Network Address Translation). Фильтрация пакетов здесь допускается только в исключительных случаях.
5			Принятие решения о маршрутизации.
6	Mangle	INPUT	Пакет попадает в цепочку INPUT таблицы mangle. Здесь вносятся изменения в заголовок пакета перед тем как он будет передан локальному приложению.
7	Filter	INPUT	Здесь производится фильтрация входящего трафика. Помните, что все входящие пакеты, адресованные нам, проходят через эту цепочку, независимо от того с какого интерфейса они поступили.
8			Локальный процесс/приложение

Важно помнить, что на этот раз пакеты идут через цепочку *INPUT*, а не через *FORWARD*. И в заключение рассмотрим порядок движения пакетов, созданных локальными процессами.

Таблица Filter

Как следует из названия, в этой таблице должны содержаться наборы правил для выполнения фильтрации пакетов. Пакеты могут пропускаться далее, либо отвергаться, в зависимости от их содержимого. В этой таблице допускается использование большинства из существующих действий, однако ряд действий, которые были рассмотрены выше в этой главе, должны выполняться только в присущих им таблицах.

Таблица *filter* используется главным образом для фильтрации пакетов. Для примера, здесь мы можем выполнить *DROP, LOG, ACCEPT* или *REJECT* без каких либо сложностей, как в других таблицах. Имеется три встроенных цепочки. Первая- *FORWARD*, используемая для фильтрации пакетов, идущих транзитом через брандмауэр. Цепочку *INPUT* проходят пакеты, которые предназначены локальным приложениям (брандмауэру). И цепочка *OUTPUT*- используется для фильтрации исходящих пакетов

Построение правил

В данной разделе будет обсуждаться порядок построения собственных правил для iptables. Каждая строка, вставляемая ту или иную цепочку, должна содержать отдельное правило. Также рассмотрим основные проверки и действия и порядок создания своих собственных цепочек правил.

Как уже говорилось выше, каждое правило - это строка, содержащая в себе критерии определяющие, подпадает ли пакет под заданное правило, и действие, которое необходимо выполнить в случае выполнения критерия. В общем виде правила записываются примерно так:

iptables [-t table] command [match] [target/jump]

Нигде не утверждается, что описание действия (target/jump) должно стоять последним в строке, однако, будем придерживаться именно такой нотации для удобства.

Если в правило не включается спецификатор [-t table], то по умолчанию предполагается использование таблицы *filter*, если же предполагается использование другой

таблицы, то это требуется указать явно. Спецификатор таблицы так же можно указывать в любом месте строки правила, однако более или менее стандартом считается указание таблицы в начале правила.

Далее, непосредственно за именем таблицы, должна стоять команда. Если спецификатора таблицы нет, то команда всегда должна стоять первой. Команда определяет действие iptables, например: вставить правило, или добавить правило в конец цепочки, или удалить правило и т.п.

Раздел match задает критерии проверки, по которым определяется, подпадает ли пакет под действие этого правила или нет. Здесь можно указать самые разные критерии - и IP-адрес источника пакета или сети, и сетевой интерфейс и т.д. Существует множество критериев, которые будут рассмотрены в данной главе.

И, наконец, target указывает, какое действие должно быть выполнено при условии выполнения критериев в правиле. Здесь можно заставить ядро передать пакет в другую цепочку правил, «сбросить» пакет и забыть про него, выдать на источник сообщение об ошибке и т.п.

Команды

Ниже приводится список команд и правила их использования. Посредством команд пользователь сообщает iptables что он предполагает сделать. Обычно предполагается одно из двух действий это добавление нового правила в цепочку или удаление существующего правила из той или иной таблицы. Далее приведены команды, которые используются в iptables.

Таблица 10.3 «Команды»

Команда	-A, --append
Пример	iptables -A INPUT ...
Пояснения	Добавляет новое правило в конец заданной цепочки.
Команда	-D, --delete
Пример	iptables -D INPUT --dport 80 -j DROP, iptables -D INPUT 1
Пояснения	Удаление правила из цепочки. Команда имеет два формата записи, ый -- когда задается критерий сравнения с опцией -D (см. первый мер), второй -- порядковый номер правила. Если задается критерий внения, то удаляется правило, которое имеет в себе этот критерий, если ется номер правила, то будет удалено правило с заданным номером. т правил в цепочках начинается с 1.
Команда	-R, --replace
Пример	iptables -R INPUT 1 -s 192.168.0.1 -j DROP
Пояснения	Данная команда заменяет одно правило другим. В основном она ользуется во время отладки новых правил.
Команда	-I, --insert
Пример	iptables -I INPUT 1 --dport 80 -j ACCEPT
Пояснения	Вставляет новое правило в цепочку. Число, следующее за именем очки, указывает номер правила, перед которым нужно вставить новое вило, другими словами число задает номер для вставляемого правила. В мере выше, указывается, что данное правило должно быть 1-м в цепочке UT.

Команда	-L, --list
Пример	iptables -L INPUT
Пояснения	Вывод списка правил в заданной цепочке, в данном примере дполагается вывод правил из цепочки INPUT. Если имя цепочки не ывается, то выводится список правил для всех цепочек. Формат вывода исит от наличия дополнительных ключей в команде, например -n, -v, и
Команда	-F, --flush
Пример	iptables -F INPUT
Пояснения	Сброс (удаление) всех правил из заданной цепочки (таблицы). Если цепочки и таблицы не указывается, то удаляются все правила, во всех очках.
Команда	-Z, --zero
Пример	iptables -Z INPUT
Пояснения	Обнуление всех счетчиков в заданной цепочке. Если имя цепочки не ывается, то подразумеваются все цепочки. При использовании ключа -v местно с командой -L, на вывод будут поданы и состояния счетчиков етов, попавших под действие каждого правила. Допускается совместное ользование команд -L и -Z. В этом случае будет выдан сначала список вил со счетчиками, а затем произойдет обнуление счетчиков.
Команда	-N, --new-chain
Пример	iptables -N allowed
Пояснения	Создается новая цепочка с заданным именем в заданной таблице В ше приведенном примере создается новая цепочка с именем allowed. Имя очки должно быть уникальным и не должно совпадать с зервированными именами цепочек и действий (DROP, REJECT и т.п.)
Команда	-X, --delete-chain
Пример	iptables -X allowed
Пояснения	Удаление заданной цепочки из заданной таблицы. Удаляемая цепочка должна иметь правил и не должно быть ссылок из других цепочек на яемую цепочку. Если имя цепочки не указано, то будут удалены все очки, определенные командой -N в заданной таблице.
Команда	-P, --policy
Пример	iptables -P INPUT DROP
Пояснения	Определяет политику по умолчанию для заданной цепочки. Политика умолчанию определяет действие, применяемое к пакетам, не попавшим действие ни одного из правил в цепочке. В качестве политики по лчанию допускается использовать DROP, ACCEPT и REJECT.
Команда	-E, --rename-chain
Пример	iptables -E allowed disallowed
Пояснения	Команда -E выполняет переименование пользовательской цепочки. В мере цепочка allowed будет переименована в цепочку disallowed. Эти еименования не изменяют порядок работы, а носят только метический характер.

Команда должна быть указана всегда. Список доступных команд можно просмотреть с помощью команды iptables -h или, что то же самое, iptables --help. Некоторые команды могут использоваться совместно с дополнительными ключами. Ниже приводится список дополнительных ключей и описывается результат их действия. Следует заметить, что здесь не приводится дополнительных ключей, которые используются при построении критериев (matches) или действий (targets). Эти опции рассмотрим далее.

Таблица 10.4 «Ключи»

Ключ	-v, --verbose
Команды, с которыми ользуется	--list, --append, --insert, --delete, --replace
Описание	Данный ключ используется для повышения информативности вывода и, правило, используется совместно с командой --list. В случае ользования с командой --list, в вывод этой команды включаются так же интерфейса, счетчики пакетов и байт для каждого правила. Формат ода счетчиков предполагает вывод кроме цифр числа еще и символьные жители K (x1000), M (x1,000,000) и G (x1,000,000,000). Для того, чтобы авить команду --list выводить полное число (без употребления жителей) требуется применять ключ -x, который описан ниже. Если ключ -verbose используется с командами --append, --insert, --delete или --replace, о на вывод будет выдан подробный отчет о произведенной операции.
Ключ	-x, --exact
Команды, с которыми ользуется	--list
Описание	Для всех чисел в выходных данных выводятся их точные значения без угления и без применения множителей K, M, G. Важно то, что данный ч используется только с командой --list и не применяется с другими андами.
Ключ	-n, --numeric
Команды, с которыми ользуется	--list
Описание	Заставляет iptables выводить IP-адреса и номера портов в числовом виде дотвращая попытки преобразовать их в символические имена. Данный ч используется только с командой --list.
Ключ	--line-numbers
Команды, с которыми ользуется	--list
Описание	Ключ --line-numbers включает режим вывода номеров строк при бражении списка правил командой --list. Номер строки соответствует иции правила в цепочке. Этот ключ используется только с командой --list.
Ключ	-c, --set-counters
Команды, с которыми ользуется	--insert, --append, --replace
Описание	Этот ключ используется при создании нового правила для установки чиков пакетов и байт в заданное значение. Например, ключ --set-counters 000установит счетчик пакетов = 20, а счетчик байт = 4000.

Ключ	--modprobe
Команды, с которыми ользуется	Все
Описание	Ключ --modprobe определяет команду загрузки модуля ядра. Данный ч используется в случае, если ваша команда modprobe находится вне пути ска (searchpath). Этот ключ может использоваться с любой командой.

Критерии

Здесь будут подробно рассмотрены критерии выделения пакетов. Все критерии разбиты на пять групп. Первая – общие критерии, которые могут использоваться в любых правилах. Вторая - TCP критерии, которые применяются только к TCP пакетам. Третья – UDP критерии, которые применяются только к UDP пакетам.

Таблица 10.5 «Общие критерии»

Критерий	-p, --protocol
Пример	iptables -A INPUT -p tcp
Описание	Этот критерий используется для указания типа протокола. Примерами протоколов могут быть TCP, UDP и ICMP. Список протоколов можно посмотреть в файле /etc/p rotocols. Прежде всего, в качестве имени протокола в данный критерий можно передавать три вышеупомянутых протокола, а также ключевое слово ALL. В качестве протокола допускается передавать число - номер протокола, так например, 255 соответствует протоколу RAW IP. Соответствия между номерами протоколов и их именами можно посмотреть в файле /etc/protocols, который уже упоминался выше. Если данному критерию передается числовое значение 0, то это эквивалентно использованию спецификатора ALL, который подразумевается по умолчанию, когда критерий --protocol не используется. Для логической инверсии критерия, перед именем протокола (списком протоколов) используется символ !, например --protocol ! tcp подразумевает пакеты любого протокола, кроме tcp.
Критерий	-s, --src, --source
Пример	iptables -A INPUT -s 192.168.1.1
Описание	IP-адрес(а) источника пакета. Адрес источника может указываться так, как показано в примере, тогда подразумевается единственный IP-адрес. А можно указать адрес в виде address/mask, например как 192.168.0.0/255.255.255.0, или более современным способом 192.168.0.0/24, то есть фактически определяя диапазон адресов Как и ранее, символ !, установленный перед адресом, означает логическое отрицание, то есть --source ! 192.168.0.0/24 означает любой адрес кроме адресов 192.168.0.x
Критерий	-d, --dst, --destination
Пример	iptables -A INPUT -d 192.168.1.1
Описание	IP-адрес(а) получателя. Имеет синтаксис схожий с критерием --source, за исключением того, что подразумевает адрес места назначения. Точно так же может определять как единственный IP-адрес, так и диапазон адресов. Символ ! используется для логической инверсии критерия.
Критерий	-i, --in-interface
Пример	iptables -A INPUT –I eth0

Описание	Интерфейс, с которого был получен пакет. Использование этого критерия допускается только в цепочках INPUT, FORWARD и PREROUTING, в любых других случаях будет вызывать сообщение об ошибке. При отсутствии этого критерия предполагается любой интерфейс, что равносильно использованию критерия -i +. Как и прежде, символ ! инвертирует результат совпадения. Если имя интерфейса завершается символом +, то критерий задает все интерфейсы, начинающиеся с заданной строки, например -i PPP+ обозначает любой PPP интерфейс, а запись -i ! eth+ -- любой интерфейс, кроме любого eth.
Критерий	-o, --out-interface
Пример	iptables -A FORWARD -o eth0
Описание	Задает имя выходного интерфейса. Этот критерий допускается использовать только в цепочках OUTPUT, FORWARD и POSTROUTING, в противном случае будет генерироваться сообщение об ошибке. При отсутствии этого критерия предполагается любой интерфейс, что равносильно использованию критерия -o +. Как и прежде, символ ! инвертирует результат совпадения. Если имя интерфейса завершается символом +, то критерий задает все интерфейсы, начинающиеся с заданной строки, например -o eth+ обозначает любой eth интерфейс, а запись -o ! eth+ - любой интерфейс, кроме любого eth
Критерий	-f, --fragment
Пример	iptables -A INPUT –f
Описание	Правило распространяется на все фрагменты фрагментированного пакета, кроме первого, сделано это потому, что нет возможности определить исходящий/входящий порт для фрагмента пакета, а для ICMP-пакетов определить их тип. С помощью фрагментированных пакетов могут производиться атаки на ваш МСЭ, так как фрагменты пакетов могут не отлавливаться другими правилами. Как и раньше, допускается использования символа ! для инверсии результата сравнения. только в данном случае символ ! должен предшествовать критерию -f, например ! -f. Инверсия критерия трактуется как «все первые фрагменты фрагментированных пакетов и/или нефрагментированные пакеты, но не вторые и последующие фрагменты фрагментированных пакетов».

Таблица 10.6 «TCP критерии»

Критерий	--sport, --source-port
Пример	iptables -A INPUT -p tcp --sport 22
Описание	Исходный порт, с которого был отправлен пакет. В качестве параметра может указываться номер порта или название сетевой службы. Соответствие имен сервисов и номеров портов вы сможете найти в файле /etc/services При указании номеров портов правила отрабатывают несколько быстрее. однако это менее удобно при разборе листингов скриптов. Номера портов могут задаваться в виде интервала из минимального и максимального номеров, например --source-port 22:80. Если опускается минимальный порт, то есть когда критерий записывается как --source-port :80, то в качестве начала диапазона принимается число 0. Если опускается максимальный порт, то есть когда критерий записывается как --source-port 22:, то в качестве конца диапазона принимается число 65535. Допускается такая запись --source-port 80:22, в этом случае iptables

	поменяет числа 22 и 80 местами, то есть подобного рода запись будет преобразована в --source-port 22:80. Как и раньше, символ ! используется для инверсии. Так критерий --source-port ! 22 подразумевает любой порт, кроме 22. Инверсия может применяться и к диапазону портов, например --source-port ! 22:80.
Критерий	--dport, --destination-port
Пример	iptables -A INPUT -p tcp --dport 22
Описание	Порт, на который адресован пакет. Аргументы задаются в том же формате, что и для --source-port.
Критерий	--tcp-flags
Пример	iptables -p tcp --tcp-flags SYN,ACK,FIN SYN
Описание	Определяет маску и флаги tcp-пакета. Пакет считается удовлетворяющим критерию, если из перечисленных флагов в первом списке в единичное состояние установлены флаги из второго списка. Так для вышеуказанного примера под критерий подпадают пакеты, у которых флаг SYN установлен, а флаги FIN и ACK сброшены. В качестве аргументов критерия могут выступать флаги SYN, ACK, FIN, RST, URG, PSH, а так же зарезервированные идентификаторы ALL и NONE. ALL -- значит ВСЕ флаги и NONE - НИ ОДИН флаг. Так, критерий --tcp-flags ALL NONE означает, что все флаги в пакете должны быть сброшены. Как и ранее, символ ! означает инверсию критерия Важно: имена флагов в каждом списке должны разделяться запятыми, пробелы служат для разделения списков.
Критерий	--syn
Пример	iptables -p tcp –syn
Описание	Критерий --syn является по сути реликтом, перекочевавшим из ipchains. Критерию соответствуют пакеты с установленным флагом SYN и сброшенными флагами ACK и FIN. Этот критерий аналогичен критерию --tcp-flags SYN,ACK,FIN SYN. Такие пакеты используются для открытия соединения TCP. Заблокировав такие пакеты, вы надежно заблокируете все входящие запросы на соединение, однако этот критерий не способен заблокировать исходящие запросы на соединение. Как и ранее, допускается инвертирование критерия символом !. Так критерий ! --syn означает все пакеты, не являющиеся запросом на соединение, то есть все пакеты с установленными флагами FIN или ACK.
Критерий	--tcp-option
Пример	iptables -p tcp --tcp-option 16
Описание	Удовлетворяющим условию данного критерия будет будет считаться пакет, TCP параметр которого равен заданному числу. TCP Option - это часть заголовка пакета. Она состоит из 3 различных полей. Первое 8-ми битовое поле содержит информацию об опциях, используемых в данном соединении. Второе 8-ми битовое поле содержит длину поля опций. Если следовать стандартам до конца, то следовало бы реализовать обработку всех возможных вариантов, однако, вместо этого мы можем проверить первое поле, и в случае, если там указана неподдерживаемая нашим МСЭом опция, то просто перешагнуть через третье поле (длина которого содержится во втором поле). Пакет, который не будет иметь полного TCP заголовка, будет сброшен автоматически при попытке изучения его TCP параметра. Как и ранее, допускается использование флага инверсии

	условия [!]..

Таблица 10.7 «UDP критерии»

Критерий	--sport, --source-port
Пример	iptables -A INPUT -p udp --sport 53
Описание	Исходный порт, с которого был отправлен пакет. В качестве параметра может указываться номер порта или название сетевой службы. Соответствие имен сервисов и номеров портов можно найти в файле /etc/services При указании номеров портов правила отрабатывают несколько быстрее. однако это менее удобно при разборе листингов скриптов. Номера портов могут задаваться в виде интервала из минимального и максимального номеров, например --source-port 22:80. Если опускается минимальный порт, то есть когда критерий записывается как --source-port :80, то в качестве начала диапазона принимается число 0. Если опускается максимальный порт, то есть когда критерий записывается как --source-port 22:, то в качестве конца диапазона принимается число 65535. Допускается такая запись --source-port 80:22, в этом случае iptables поменяет числа 22 и 80 местами, то есть подобного рода запись будет преобразована в --source-port 22:80. Как и раньше, символ ! используется для инверсии. Так критерий --source-port ! 22 подразумевает любой порт, кроме 22. Инверсия может применяться и к диапазону портов, например --source-port ! 22:80.
Критерий	--dport, --destination-port
Пример	iptables -A INPUT –p udp --dport 53
Описание	Порт, на который адресован пакет. Формат аргументов полностью аналогичен принятому в критерии --source-port.

Действия и переходы

Действия и переходы сообщают правилу, что необходимо выполнить, если пакет соответствует заданному критерию. Чаще всего употребляются действия ACCEPT и DROP.

Описание переходов в правилах выглядит точно так же как и описание действий, то есть ставится ключ -j и указывается название цепочки правил, на которую выполняется переход. На переходы накладывается ряд ограничений, первое - цепочка, на которую выполняется переход, должна находиться в той же таблице, что и цепочка, из которой этот переход выполняется, второе - цепочка , являющаяся целью перехода должна быть создана до того как на нее будут выполняться переходы. Например, создадим цепочку tcp_packets в таблице filter с помощью команды

```
iptables -N tcp_packets
```

Теперь можно выполнять переходы на эту цепочку подобно

```
iptables -A INPUT -p tcp -j tcp_packets
```

То есть, встретив пакет протокола tcp, iptables произведет переход на цепочку tcp_packets и продолжит движение пакета по этой цепочке. Если пакет достиг конца цепочки, то он будет возвращен в вызывающую цепочку (в нашем случае это цепочка INPUT) и движение пакета продолжится с правила, следующего за правилом, вызвавшем переход.

Если к пакету во вложенной цепочке будет применено действие ACCEPT, то автоматически пакет будет считаться принятым и в вызывающей цепочке, и уже не будет продолжать движение по вызывающим цепочкам. Однако пакет пойдет по другим цепочкам в других таблицах.

Действие - это предопределенная команда, описывающая действие, которое необходимо выполнить, если пакет совпал с заданным критерием. Например, можно применить действие *DROP* или *ACCEPT* к пакету, в зависимости от требований. Существует и ряд других действий, которые описываются ниже в этой секции. В результате выполнения одних действий, пакет прекращает свое прохождение по цепочке, например *DROP* и *ACCEPT*, в результате других, после выполнения неких операций, продолжает проверку, например, LOG, в результате работы третьих даже видоизменяется, например *DNAT* и *SNAT*, *TTL* и *TOS*, но так же продолжает продвижение по цепочке.

Действие ACCEPT

Данная операция не имеет дополнительных ключей. Если над пакетом выполняется действие *ACCEPT*, то пакет прекращает движение по цепочке (и всем вызвавшим цепочкам, если текущая цепочка была вложенной) и считается ПРИНЯТЫМ, тем не менее, пакет продолжит движение по цепочкам в других таблицах и может быть отвергнут там. Действие задается с помощью ключа -j ACCEPT.

Действие DROP

Данное действие просто «сбрасывает» пакет и iptables «забывает» о его существовании. «Сброшенные» пакеты прекращают свое движение полностью, то есть они не передаются в другие таблицы, как это происходит в случае с действием *ACCEPT*. Следует помнить, что данное действие может иметь негативные последствия, поскольку может оставлять незакрытые «мертвые» сокеты как на стороне сервера, так и на стороне клиента, наилучшим способом защиты будет использование действия *REJECT* особенно при защите от сканирования портов.

Действие RETURN

Действие *RETURN* прекращает движение пакета по текущей цепочке правил и производит возврат в вызывающую цепочку, если текущая цепочка была вложенной, или, если текущая цепочка лежит на самом верхнем уровне (например *INPUT*), то к пакету будет применена политика по умолчанию. Обычно, в качестве политики по умолчанию назначают действия *ACCEPT* или *DROP*

Для примера, допустим, что пакет идет по цепочке *INPUT* и встречает правило, которое производит переход во вложенную цепочку - --jump EXAMPLE_CHAIN. Далее, в цепочке *EXAMPLE_CHAIN* пакет встречает правило, которое выполняет действие --jump RETURN. Тогда произойдет возврат пакета в цепочку *INPUT*. Другой пример, пусть пакет встречает правило, которое выполняет действие --jump RETURN в цепочке *INPUT*. Тогда к пакету будет применена политика по молчанию цепочки *INPUT*.

Действие LOG

LOG - действие, которое служит для журналирования отдельных пакетов и событий. В журнал могут заноситься заголовки IP пакетов и другая интересующая вас информация. Информация из журнала может быть затем прочитана с помощью dmesg или syslogd либо с помощью других программ.

LOG имеет пять ключей, которые перечислены ниже.

Таблица 10.8 «Ключи для действия LOG»

Ключ	--log-level
Пример	iptables -A FORWARD -p tcp -j LOG --log-level debug
Описание	Используется для задания уровня журналирования (log level). Полный список ней можно найти в руководстве (man) по syslog.conf. Обычно, можно задать ующие уровни: debug, info, notice, warning, warn, err, error, crit, alert, emerg и panic. евое слово error означает то же самое, что и err, warn - warning и panic - emerg. о: в последних трех парах слов не следует использовать error, warn и panic. ритет определяет различия в том как будут заноситься сообщения в журнал. Все щения заносятся в журнал средствами ядра. Если вы установите строку kern.=info og/iptables в файле syslog.conf, то все ваши сообщения из iptables, использующие нь info, будут заноситься в файл /var/log/iptables Однако, в этот файл попадут и е сообщения, поступающие из других подсистем, которые используют уровень info.
Ключ	--log-prefix
Пример	iptables -A INPUT -p tcp -j LOG --log-prefix «INPUT packets»
Описание	Ключ задает текст (префикс), которым будут предваряться все сообщения iptables. щения со специфичным префиксом затем легко можно найти, к примеру, с помощью Префикс может содержать до 29 символов, включая и пробелы.
Ключ	--log-tcp-sequence
Пример	iptables -A INPUT -p tcp -j LOG --log-tcp-sequence
Описание	Этот ключ позволяет заносить в журнал номер TCP Sequence пакета. Номер TCP nce идентифицирует каждый пакет в потоке и определяет порядок «сборки» потока. ключ потенциально опасен для безопасности системы, если системный журнал шает доступ «НА ЧТЕНИЕ» всем пользователям. Как и любой другой журнал, жащий сообщения от iptables.
Ключ	--log-tcp-options
Пример	iptables -A FORWARD -p tcp -j LOG --log-tcp-options
Описание	Этот ключ позволяет заносить в системный журнал различные сведения из овка TCP пакета. Такая возможность может быть полезна при отладке. Этот ключ не дополнительных параметров, как и большинство ключей действия LOG.
Ключ	--log-ip-options
Пример	iptables -A FORWARD -p tcp -j LOG --log-ip-options
Описание	Этот ключ позволяет заносить в системный журнал различные сведения из овка IP пакета. Во многом схож с ключом --log-tcp-options, но работает только с IP овком.

Действие REJECT

REJECT используется, как правило, в тех же самых ситуациях, что и *DROP*, но в отличие от *DROP*, команда *REJECT* выдает сообщение об ошибке на машину, передавший пакет. Действие *REJECT* работает только в цепочках *INPUT, FORWARD и OUTPUT* (и во вложенных в них цепочках). Пока существует только единственный ключ, управляющий поведением команды *REJECT.*

Таблица 10.9 «Действие REJECT»

Ключ	--reject-with
Пример	iptables -A FORWARD -p TCP --dport 22 -j REJECT --reject-with tcp-reset

Описание	Указывает, какое сообщение необходимо передать в ответ, если пакет совпал с заданным критерием. При применении действия REJECT к пакету, сначала на машину-отправитель будет отослан указанный ответ, а затем пакет будет «сброшен». Допускается использовать следующие типы ответов: icmp-net-unreachable, icmp-host-unreachable, icmp-port-unreachable, icmp-proto-unreachable, icmp-net-prohibited и icmp-host-prohibited. По-умолчанию передается сообщение port-unreachable. Все вышеуказанные типы ответов являются ICMP error messages. В заключение укажем еще один тип ответа - tcp-reset, который используется только для протокола TCP. Если указано значение tcp-reset, то действие REJECT передаст в ответ пакет TCP RST, пакеты TCP RST используются для закрытия TCP соединений.

Задание на практическую работу

Вариант	Задание
1	• Запретить SSH-трафик из внешней сети • Написать правила для журналирования для всех пакетов, приходящих на интерфейс eth0 • Составить правила для фильтрации входящего трафика с лабораторных машин с целями DROP и REJECT • Создать новую цепочку назвав ее *имя_фамилия* с некоторым действием и после этого выполнить переход из цепочки FORWARD в созданную цепочку • Установить политику по умолчанию DROP для всех цепочек таблицы filter. • Определить IP-адрес(а) сайта *kavicom.ru*. Составить правило фильтрации для этих узлов с действием DROP. Проверить доступность этих интернет-сайтов после применения правила фильтрации.
2	• Запретить установление TELNET- соединения с данным компьютером. • Написать правила для журналирования для всех пакетов, приходящих на интерфейс eth0 • Составить правила для фильтрации входящего трафика с лабораторных машин с целями DROP и REJECT по протоколам tcp,udp и icmp • Разрешить выполнение запросов к DNS и прохождение трафика, необходимого для отображения страниц через браузер • Создать новую цепочку назвав ее *имя_фамилия* с некоторым действием и после этого выполнить переход из цепочки FORWARD в созданную цепочку • Определить IP-адрес(а) сайта *kinopoisk.ru*. Составить правило фильтрации для этих узлов с действием DROP. Проверить доступность этих интернет-сайтов после применения правила фильтрации.
3	• Запретить POP-трафик с внешними сетями • Написать правила для журналирования для всех пакетов, приходящих на

	интерфейс eth0 • Разрешается доступ к сервису ftp, выполняющемуся на узле с IP-адресом 210.210.210.10, с узла с IP-адресом 190.190.190.190 • Разрешить входящие и исходящие соединения на порты 2048:65535 • Создать новую цепочку назвав ее *имя_фамилия* с некоторым действием и после этого выполнить переход из цепочки FORWARD в созданную цепочку • Определить IP-адрес(а) сайта *lostfilm.tv*. Составить правило фильтрации для этих узлов с действием DROP. Проверить доступность этих интернет-сайтов после применения правила фильтрации.
4	• Разрешить для внутренней сети использование FTP-сервера. • Написать правила для журналирования для всех пакетов, приходящих на интерфейс eth0 • Клиентским программам, выполняющимся на узлах сети 210.210.210.0/24, разрешается полный доступ к сервисам, выполняющимся на узлах внешних сетей • Разрешить вход и выход всем пакетам, относящимся к протоколу ICMP: • Создать новую цепочку назвав ее *имя_фамилия* с некоторым действием и после этого выполнить переход из цепочки FORWARD в созданную цепочку • Определить IP-адрес(а) сайта *gismeteo.ru*. Составить правило фильтрации для этих узлов с действием DROP. Проверить доступность этих интернет-сайтов после применения правила фильтрации.
5	• Разрешить всем пользователям внешней сети использование только web- и FTP-серверов • Написать правила для журналирования для всех пакетов, приходящих на интерфейс eth0 • Определить номер порта по которому работает протокол aol (ICQ) и составить правило фильтрации с действием DROP для этого протокола • Создать новую цепочку назвав ее *имя_фамилия* с некоторым действием и после этого выполнить переход из цепочки FORWARD в созданную цепочку • Разрешить входящие и исходящие соединения на порты 3312,2234 • Определить IP-адрес(а) сайта *misis.ru*. Составить правило фильтрации для этих узлов с действием DROP. Проверить доступность этих интернет-сайтов после применения правила фильтрации.
6	• Запретить подключение к 80 порту Web – сервера • Написать правила для журналирования для всех пакетов, приходящих на интерфейс eth0 • Составить правила для фильтрации входящего трафика с лабораторных машин с целями DROP и REJECT по критерию интерфейса • Создать новую цепочку назвав ее *имя_фамилия* с некоторым действием и после этого выполнить переход из цепочки FORWARD в созданную цепочку • Разрешить доступ входящих и исходящих соединений по протоколу ftp,ftp-data • Определить IP-адрес(а) сайта rutracker.org Составить правило фильтрации

	для этих узлов с действием DROP. Проверить доступность этих интернет-сайтов после применения правила фильтрации.
7	• Запретить использование TFTP • Написать правила для журналирования для всех пакетов, приходящих на интерфейс eth0 • Составить правило фильтрации для входящего трафика на интерфейс eth0, по протоколу tcp порту №22 • Запретить входящий доступ по портам и протоколу udp 16825 • Создать новую цепочку назвав ее *имя_фамилия* с некоторым действием и после этого выполнить переход из цепочки FORWARD в созданную цепочку • Определить IP-адрес(а) сайта facebook.com. Составить правило фильтрации для этих узлов с действием DROP. Проверить доступность этих интернет-сайтов после применения правила фильтрации.
8	• Разрешить подключение к MySQL – серверу только с определенного адреса. • Написать правила для журналирования для всех пакетов, приходящих на интерфейс eth0 • Составить правила для фильтрации входящего трафика с лабораторных машин с целями DROP и REJECT по протоколам tcp, с использованием портов №22,№80,№445 • Запретить доступ исходящих соединений по протоколу ICMP • Создать новую цепочку назвав ее *имя_фамилия* с некоторым действием и после этого выполнить переход из цепочки FORWARD в созданную цепочку • Определить IP-адрес(а) сайта google.com. Составить правило фильтрации для этих узлов с действием DROP. Проверить доступность этих интернет-сайтов после применения правила фильтрации.
9	• Разрешить почтовому серверу обмен SMTP-трафиком с внешней сетью • Написать правила для журналирования для всех пакетов, приходящих на интерфейс eth0 • Разрешается доступ из внешней сети к сервису http, выполняющемуся на узле с IP-адресом 205.205.205.20 • Создать новую цепочку назвав ее *имя_фамилия* с некоторым действием и после этого выполнить переход из цепочки FORWARD в созданную цепочку • Запретить входящий доступ по портам и протоколу udp к портам 1456,3333 • Определить IP-адрес(а) сайта sf-misis.ru. Составить правило фильтрации для этих узлов с действием DROP. Проверить доступность этих интернет-сайтов после применения правила фильтрации.
10	• Разрешить SSH-трафик из внешней сети • Написать правила для журналирования для всех пакетов, приходящих на интерфейс eth0 • Разрешается доступ из внешней сети к сервисам ftp, dns, echo, выполняющимся на узле с IP-адресом 205.205.205.30 • Создать новую цепочку назвав ее *имя_фамилия* с некоторым действием и после этого выполнить переход из цепочки FORWARD в созданную цепочку

	• Запретить входящий доступ по портам и протоколу tcp к портам 456,3128 • Определить IP-адрес(а) сайта youtube.com. Составить правило фильтрации для этих узлов с действием DROP. Проверить доступность этих интернет-сайтов после применения правила фильтрации.
11	• Разрешить пользователям внутренней сети использование почтовых сервисов SMTP и POP3. Запретить остальным использование почтовых сервисов • Написать правила для журналирования для всех пакетов, приходящих на интерфейс eth0 • Разрешается доступ из внешнейсети к сервисам smtp, pop3, выполняющимся на узле с IP-адресом 205.205.205.40 • Разрешить исходящий доступ по портам и протоколу tcp к портам 368:774 • Создать новую цепочку назвав ее *имя_фамилия* с некоторым действием и после этого выполнить переход из цепочки FORWARD в созданную цепочку • Определить IP-адрес(а) сайта mail.ru Составить правило фильтрации для этих узлов с действием DROP. Проверить доступность этих интернет-сайтов после применения правила фильтрации.
12	• Разрешить для внутренней сети использование только web-сервера • Написать правила для журналирования для всех пакетов, приходящих на интерфейс eth0 • Разрешается доступ из внешней сети к сервису ssh, выполняющемуся на узлах сети 205.205.205.0.24 • Разрешить доступ исходящих соединений по протоколу smtp • Создать новую цепочку назвав ее *имя_фамилия* с некоторым действием и после этого выполнить переход из цепочки FORWARD в созданную цепочку • Определить IP-адрес(а) сайта ya.ru. Составить правило фильтрации для этих узлов с действием DROP. Проверить доступность этих интернет-сайтов после применения правила фильтрации.
13	• Запретить для внешней сети использование FTP-сервера. • Написать правила для журналирования для всех пакетов, приходящих на интерфейс eth0 • Клиентским программам, выполняющимся на узлах сети 210.210.210.0/24, разрешается полный доступ к сервисам, выполняющимся на узлах сети 205.205.205.0.24 • Создать новую цепочку назвав ее *имя_фамилия* с некоторым действием и после этого выполнить переход из цепочки FORWARD в созданную цепочку • Разрешить доступ исходящих соединений по протоколу DNS • Определить IP-адрес(а) сайта vk.com. Составить правило фильтрации для этих узлов с действием DROP. Проверить доступность этих интернет-сайтов после применения правила фильтрации.
14	• Разрешить подключение к 80 порту Web – сервера • Написать правила для журналирования для всех пакетов, приходящих на интерфейс eth0

	• Составить правила для фильтрации входящего icmp-трафика с лабораторных машин с целями DROP и REJECT • Создать новую цепочку назвав ее *имя_фамилия* с некоторым действием и после этого выполнить переход из цепочки FORWARD в созданную цепочку • Запретить входящий доступ по портам и протоколу udp к портам 334:1658 • Определить IP-адрес(а) сайта odnoklassniki.ru. Составить правило фильтрации для этих узлов с действием DROP. Проверить доступность этих интернет-сайтов после применения правила фильтрации.

Контрольные вопросы:

1. Понятия: фильтрация пакетов, файерволл (межсетевой экран, брандмауэр).
2. Соотношение фильтрации пакетов и маршрутизации (взаимозаменяемы ли эти понятия, в каком порядке происходят действия над пакетом, какие протоколы и приложения за что отвечают).
3. Iptables: цепочки и таблицы: назначение, функциональные особенности.
4. Iptables: политики.
5. TCP и UDP критерии.

Практическая работа №11

Построение одноуровневого сетевого проекта на базе технологии Fast Ethernet.

Цель работы: Приобрести практические навыки в построении одноуровневого сетевого проекта на базе технологии Fast Ethernet, проанализировать результаты, оценить превосходство технологии Fast Ethernet.

В результате выполнения практической работы студент должен:

Знать:

- общие характеристики стандартов Fast Ethernet;
- преимущества технологии Fast Ethernet перед другими технологиями.

Уметь:

- строить одноуровневые сетевые проекты на базе технологии Fast Ethernet;
- оценивать результаты работы сетевого проекта.

Задание для практической работы:

1. Изучить теоретическую часть
2. Выполнить практическую часть
2. Ответить на контрольные вопросы
3. Оформить отчет.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Рост требований к пропускной способности локальных сетей.

Сегодня все чаще и чаще возникают повышенные требование к пропускной способности каналов между клиентами сети и серверами. Это происходит по разным причинам:

- повышение производительности клиентских компьютеров;
- увеличение числа пользователей в сети;
- появление приложений, работающих с мультимедийной информацией, которая хранится в файлах очень больших размеров;
- увеличение числа сервисов, работающих в реальном масштабе времени.

Следовательно, имеется потребность в экономичном решении, предоставляющем нужную пропускную способность во всех перечисленных случаях. Ситуация усложняется еще и тем, что нужны различные технологические решения - для организации магистралей сети и подключения серверов одни, а для подключения настольных клиентов - другие.

10-Мегабитный Ethernet устраивал большинство пользователей на протяжении около 15 лет. Однако в начале 90-х годов начала ощущаться его недостаточная пропускная способность. Если для компьютеров на процессорах Intel 80286 или 80386 с шинами ISA (8 Мбайт/с) или EISA (32 Мбайт/с) пропускная способность сегмента Ethernet составляла 1/8 или 1/32 канала "память - диск", то это хорошо согласовывалась с соотношением объемов локальных данных и внешних данных для компьютера. Теперь же у мощных клиентских станций с процессорами Pentium или Pentium PRO и шиной PCI (133 Мбайт/с) эта доля упала до 1/133, что явно недостаточно. Поэтому многие сегменты 10-Мегабитного Ethernet'а стали перегруженными, реакция серверов в них значительно упала, а частота возникновения коллизий существенно возросла, еще более снижая номинальную пропускную способность.

Способы повышения пропускной способности сети

Для повышения пропускной способности сети можно применить несколько способов: сегментация сети с помощью мостов и маршрутизаторов, сегментация сети с помощью коммутаторов и повышение пропускной способности самого протокола.

Сегментация сети с помощью мостов или маршрутизаторов может повысить пропускную способность сегментов сети за счет их разгрузки от трафика других сегментов только в том случае, когда межсегментный трафик составляет незначительную долю от внутрисегментного, поскольку и мосты, и маршрутизаторы не обладают высокой внутренней пропускной способностью.

В начале 90-х годов произошло два значительных события, которые дали возможность повысить пропускную способность сегментов локальных сетей, и в первую очередь сегментов технологии Ethernet.

Первое событие состояло в появлении мостов нового поколения - коммутаторов, которые в отличие от традиционного моста имели большое количество портов и обеспечивали передачу кадров между портами одновременно. Это позволило теперь эффективно применять коммутаторы и для тех сетей, в которых межсегментный трафик не очень отличался от внутрисегментного. Будущее технологии Ethernet после появления коммутаторов стало более устойчивым, так как появилась возможность соединить низкую стоимость технологии Ethernet с высокой производительностью сетей, построенных на основе коммутаторов.

Второе событие заключалось в появлении экспериментальных сетей, в которых использовался протокол Ethernet с более высокой битовой скоростью передачи данных, а именно 100 Мб/с. До этого только технология *Fiber Distributed Data Interface (FDDI)* обеспечивала такую битовую скорость, но она была специально разработана для построения магистралей сетей и была слишком дорогой для подключения к сети отдельных рабочих станций или серверов.

Создание стандарта Fast Ethernet

В 1992 году группа производителей сетевого оборудования, включая таких лидеров технологии Ethernet как SynOptics, 3Com и ряд других, образовали некоммерческое объединение *Fast Ethernet Alliance* для разработки стандарта на новую технологию, которая обобщила бы достижения отдельных компаний в области Ethernet-преемственного высокоскоростного стандарта. Новая технология получила название Fast Ethernet.

Одновременно были начаты работы в институте IEEE по стандартизации новой технологии - там была сформирована исследовательская группа для изучения технического потенциала высокоскоростных технологий. За период с конца 1992 года и по конец 1993 года группа IEEE изучила 100-Мегабитные решения, предложенные различными производителями. Наряду с предложениями Fast Ethernet Alliance группа рассмотрела также и другую высокоскоростную технологию, предложенную компаниями *Hewlett-Packard и AT&T*.

В центре дискуссий была проблема сохранения соревновательного метода доступа CSMA/CD. Предложение по Fast Ethernet'у сохраняло этот метод и тем самым обеспечивало преемственность и согласованность сетей 10Base-T и 100Base-T. Коалиция HP и AT&T, которая имела поддержку гораздо меньшего числа производителей в сетевой индустрии, чем Fast Ethernet Alliance, предложила совершенно новый метод доступа, называемый *Demand Priority*. Он существенно менял картину поведения узлов в сети, поэтому не смог вписаться в

технологию Ethernet и стандарт 802.3, и для его стандартизации был организован новый комитет IEEE 802.12.

В мае 1995 года комитет IEEE принял спецификацию Fast Ethernet в качестве стандарта 802.3u, который не является самостоятельным стандартом, а представляет собой дополнение к существующему стандарту 802.3 в виде глав с 21 по 30. Отличия Fast Ethernet от Ethernet сосредоточены на физическом уровне (рисунок 11.1).

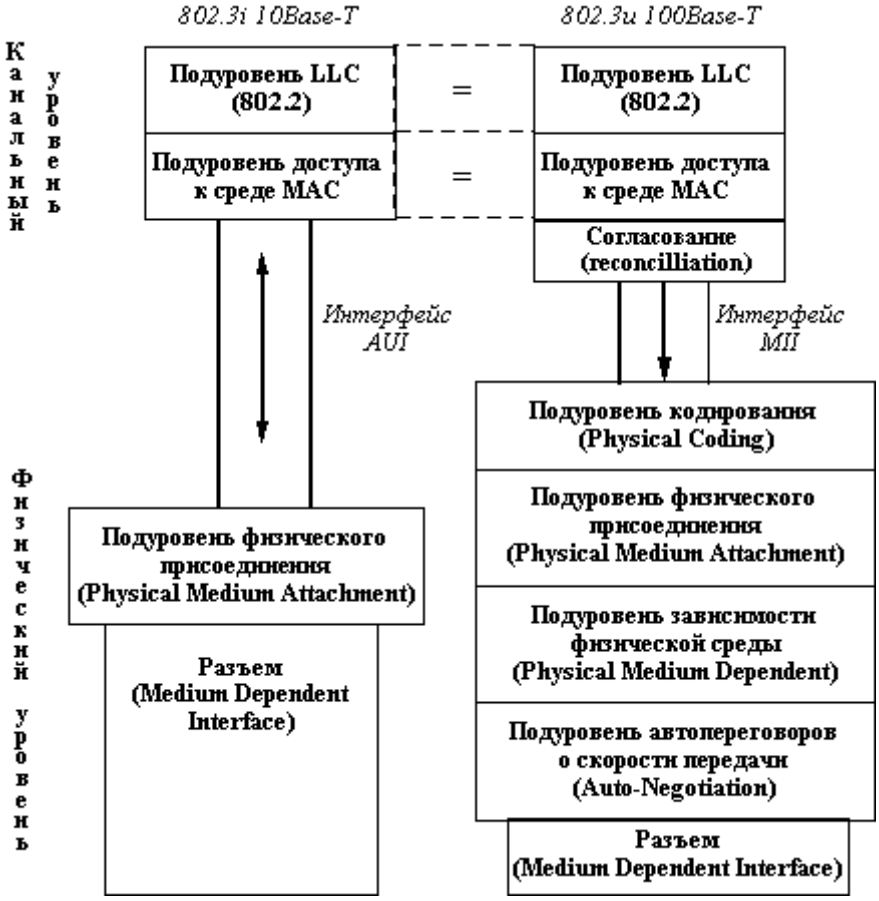


Рис. 11.1 «Отличия стека протоколов 100Base-T от стека протоколов 10Base-T»

Более сложная структура физического уровня технологии Fast Ethernet вызвана тем, что в ней используется три варианта кабельных систем - оптоволокно, 2-х парная витая пара категории 5 и 4-х парная витая пара категории 3, причем по сравнению с вариантами физической реализации Ethernet (а их насчитывается шесть), здесь отличия каждого варианта от других глубже - меняется и количество проводников, и методы кодирования. А так как физические варианты Fast Ethernet создавались одновременно, а не эволюционно, как для сетей Ethernet, то имелась возможность детально определить те подуровни физического уровня, которые не изменяются от варианта к варианту, и остальные подуровни, специфические для каждого варианта

Технология Fast Ethernet является эволюционным развитием классической технологии Ethernet. Ее основными достоинствами являются:

- увеличение пропускной способности сегментов сети до 100 Мб/с;
- сохранение метода случайного доступа Ethernet;
- сохранение звездообразной топологии сетей и поддержка традиционных сред передачи данных - витой пары и оптоволоконного кабеля.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Шаблон лабораторной работы представлен на рис. 11.2

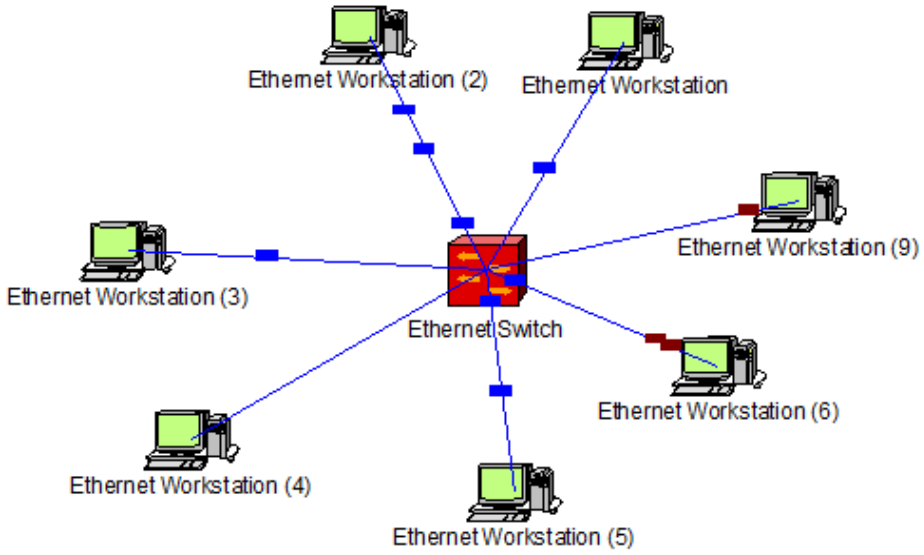


Рис.11.2 «Схема сети на практическую работу»

Схема представляет собой одноуровневый сетевой проект на базе технологии Fast Ethernet.

Задание на практическую работу

Рабочие станции в данной работе подключены к компьютеру типа «Ethernet Switch», который относится к первой категории. Используется тип кабеля 10 Base T4.

Варианты заданий приведены в табл. 11.1.

Для выполнения работы необходимо подставить данные табл. 11.1 и получить статистические данные:

1) на рабочих станциях - среднюю рабочую нагрузку (Average Workload), пакеты, обработанные за последнюю секунду (packets for last second);

2) на коммутаторах - среднее время задержки (Average delay);

3) на линиях связи - среднюю рабочую нагрузку (Average Workload).

Изменить параметры трафика, заданные в табл. 11.1, в любую сторону и проанализировать результаты.

Таблица 11.1

Число рабочих станций	Типы трафика	Параметры трафика	
		Transaction size	Time between transaction
6	Smalt InterLAN Traffic	Constant 500 betes	Exponential 0.08 s
6	LAN peer-to-peer Traffic	Uniform 500 to 1500 3ytcs	Exponential 0. 1 s
6	Small Office peer-to-peer	Uniform 500 to 600 lytcs	Exponential 0.04 s
8	LAN peer-to-peer Traffic	hxponential 500 Cbylcs	Exponential 0.001s
8	Small office peer-to-peer	Exponential 500 to 600 bytes	Exponential 0.004 s
8	Small mterLAN Traffic	Constant 500 bytes	Exponential 0.04 s
3	LAN peer-to-pcer Traffic	Exponential 500 Kbytes	Exponential 0.001s
3	Small InterLAN Traffic	Constant 500 bytes	Exponential 0. 1 s
3	Small office peer-o-peer	Jniform 500 to 600 lytes	Exponential 0.004 s
5	LAN peer-to-peer Traffic	Exponential 100 Cbytes	Constant 0. 1 s
5	Small InterLAN Traffic	Constant 500 Kbytes	Exponential 0. 1 s
5	Small office pecr-.Qjjcei- Traffic	Uniform 5 to 6 Kbytes	Exponential 1 s
4	Small inerLAN Traffic	Constant 30 Kbytes	Constant 0. 1 s
4	Traffic(15)	Normal 5 to 1 bytes	Normal 5 to 1 s
4	Small office peer-to-peer	Uniform 5 to 6 Kbytcs	Exponential 3 s

Контрольные вопросы

1. Перечислить средства объединения больших сетей;
2. Перечислить разновидности сетей Ethernet.
3. Какой метод доступа используется в сетях Ethernet?
4. Способы повышения пропускной способности сети?
5. Отличия стека протоколов 100Base-T от стека протоколов 10Base-T?

Практическая работа №12

Построение сетевого проекта, состоящего из нескольких подсетей на базе технологии Fast Ethernet.

Цель работы: Приобрести практические навыки в построении сетевых проектов, состоящих из нескольких подсетей на базе технологии Fast Ethernet. Проанализировать результаты, оценить превосходство технологии Fast Ethernet.

В результате выполнения практической работы студент должен:

Знать:

- общие характеристики стандартов Fast Ethernet;
- преимущества технологии Fast Ethernet перед другими технологиями.

Уметь:

- строить сетевые проекты, состоящие из нескольких подсетей на базе технологии Fast Ethernet;
- оценивать результаты работы сетевого проекта.

Задание для практической работы:

1. Выполнить практическую часть
2. Ответить на контрольные вопросы
3. Оформить отчет.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Шаблон лабораторной работы представлен на рис. 12.1

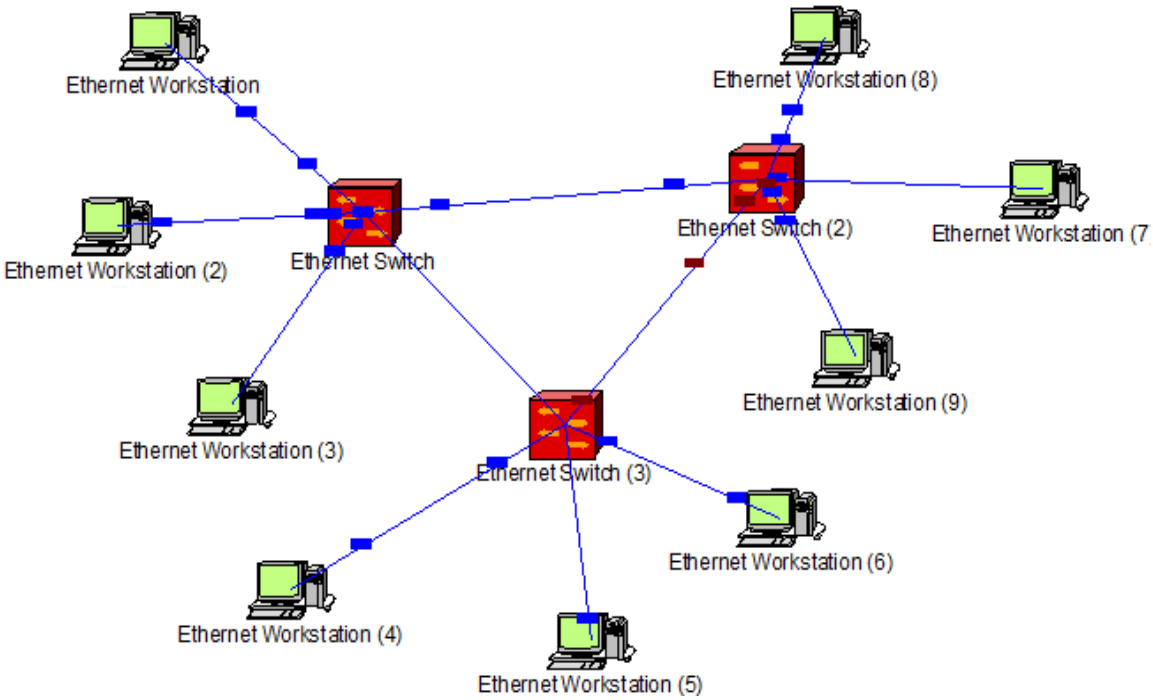
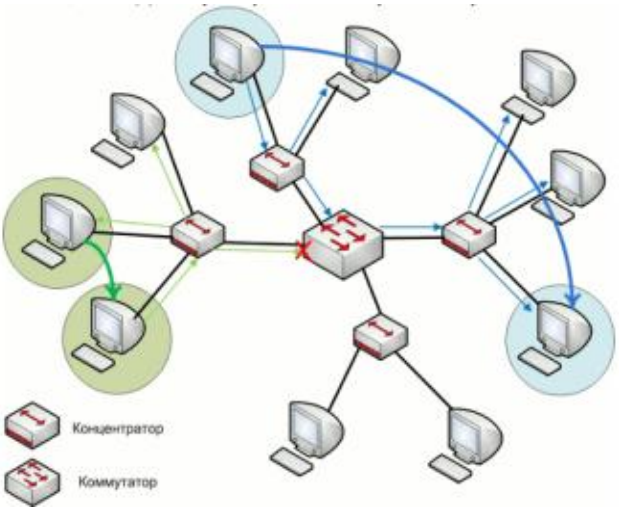


Рис.12.1 «Схема сети на практическую работу»

Схема представляет собой одноуровневый сетевой проект на базе технологии Fast Ethernet.

Для объединения сегментов сети используются коммутаторы. Сетевой коммутатор (switch) – это устройство, используемое в сетях передачи пакетов, предназначенное для объединения нескольких сегментов. В отличие от маршрутизатора (router) коммутатор

работает на канальном уровне модели OSI, что и определяет главные различия между ними. Коммутатор не занимается расчетом маршрута для дальнейшей передачи пакетов по сети, анализируя различные факторы, как это делает маршрутизатор. Switch только передает данные от одного порта к другому на основе содержащейся в пакете информации. Обычно признаком выбора выходного порта служит MAC-адрес устройства, к которому передаются данные. В свою очередь коммутатор в отличие от концентратора или репитера не просто транслирует порты ко всем выходам, которые у него есть, а к одному, заранее выбранному.



Сетевые коммутаторы применяются в нескольких технологиях, но наибольшее распространение нашли в Ethernet. Главной их задачей в сети Ethernet является разделение сети на сегменты. Это особенно актуально в сетях с большим числом рабочих станций, т.к. чем больше оконечных устройств работают одновременно с единой средой передачи данных, тем выше вероятность возникновения коллизии (одновременной передачи данных несколькими устройствами) и, следовательно, ниже эффективность работы сети. Коммутатор позволяет разбить единую сеть на несколько сегментов и увеличить число одновременно работающих устройств.

Задание на практическую работу

Для выполнения работы необходимо задать параметры шаблона, приведенные в табл. 12.1.

Таблица 12.1

	Параметры трафика	
Типы трафика	Transaction si/e	Time between transaction
Small Office	Uniform 500 to 600 bytes	Krlang 0.04 s
LAN peer-to-peer traffic	500 to 1500 Kbytes	KxponcntialO.! s
Traffic (15)	5 to 1 bytes	Normal 5 to 1 s
LAN pcer-to-peer traffic	500 to 1500 bytes	Exponential 0.0001 s
Small Office	Uniform 500 to 600 Kbytes	lirlang 0.04 s

Traffic (15)	5 to 1 Kbytes	Normal 2 to 1 s
LAN peer-to-peer traffic	500 to 1500 Kbytes	Exponential 0.01 s

При выполнении лабораторной работы необходимо осуществить следующие измерения на сетевых устройствах. На коммутаторах необходимо измерить параметры: рабочую нагрузку (average workload), среднее время задержки (average delay), число пакетов за последнюю секунду (packets for last second). На петле коммутаторов разорвать одну из линий и оценить, каким образом меняются величины. Далее подобрать параметры трафика, при которых загрузка будет минимальной. Необходимо изменять размеры входных и выходных буферов коммутаторов и анализировать соответствующие статистические данные.

Контрольные вопросы

1. Как работает коммутатор?
2. Перечислить виды коммутаторов.
3. В чем преимущество коммутаторов перед концентраторами?
4. Преимущества технологии Fast Ethernet?

Практическая работа №13

Построение многоуровневого сетевого проекта с использованием мостов.

Цель работы: Научиться строить многоуровневые сетевые проекты, анализировать их работу, изучить функции мостов.

В результате выполнения практической работы студент должен:

Знать:

- Принцип работы моста.

Уметь:

- Строить многоуровневые сетевые проекты с использованием мостов;
- оценивать результаты работы многоуровневых сетевых проектов с использованием мостов.

Задание для практической работы:

1. Изучить теоретическую часть
2. Выполнить практическую часть
2. Ответить на контрольные вопросы
3. Оформить отчет.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Прозрачные мосты незаметны для сетевых адаптеров конечных узлов, так как они самостоятельно строят специальную адресную таблицу, на основании которой можно решить, нужно передавать пришедший кадр в какой-либо другой сегмент или нет. Сетевые адаптеры при использовании прозрачных мостов работают точно так же, как и в случае их отсутствия, то есть не предпринимают никаких дополнительных действий, чтобы кадр прошел через мост. Алгоритм прозрачного моста не зависит от технологии локальной сети, в которой устанавливается мост, поэтому прозрачные мосты Ethernet работают точно так же, как прозрачные мосты FDDI.

Прозрачный мост строит свою адресную таблицу на основании пассивного наблюдения за трафиком, циркулирующим в подключенных к его портам сегментах. При этом мост учитывает адреса источников кадров данных, поступающих на порты моста. По адресу источника кадра мост делает вывод о принадлежности этого узла тому или иному сегменту сети.

Рассмотрим процесс автоматического создания адресной таблицы моста и ее использования на примере простой сети, представленной на рис. 13.1.

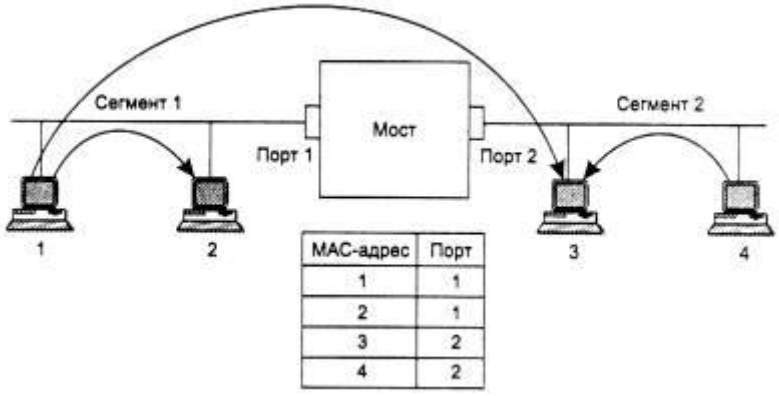


Рис. 13.1 «Принцип работы прозрачного моста»

Мост соединяет два логических сегмента. Сегмент 1 составляют компьютеры, подключенные с помощью одного отрезка коаксиального кабеля к порту 1 моста, а сегмент 2 - компьютеры, подключенные с помощью другого отрезка коаксиального кабеля к порту 2 моста.

Каждый порт моста работает как конечный узел своего сегмента за одним исключением - порт моста не имеет собственного MAC - адреса. Порт моста работает в так называемом *неразборчивом (promisquous)* режиме захвата пакетов, когда все поступающие на порт пакеты запоминаются в буферной памяти. С помощью такого режима мост следит за всем трафиком, передаваемым в присоединенных к нему сегментах, и использует проходящие через него пакеты для изучения состава сети. Так как в буфер записываются все пакеты, то адрес порта мосту не нужен.

В исходном состоянии мост ничего не знает о том, компьютеры с какими MAC - адресами подключены к каждому из его портов. Поэтому в этом случае мост просто передает любой захваченный и буферизованный кадр на все свои порты за исключением того, от которого этот кадр получен. В нашем примере у моста только два порта, поэтому он передает кадры с порта 1 на порт 2, и наоборот. Отличие работы моста в этом режиме от повторителя в том, что он передает кадр не побитно, а с буферизацией. Буферизация разрывает логику работы всех сегментов как единой разделяемой среды. Когда мост собирается передать кадр с сегмента на сегмент, например с сегмента 1 на сегмент 2, он заново пытается получить доступ к сегменту 2 как конечный узел по правилам алгоритма доступа, в данном примере - по правилам алгоритма CSMA/CD.

Одновременно с передачей кадра на все порты мост изучает адрес источника кадра и делает новую запись о его принадлежности в своей адресной таблице, которую также называют таблицей фильтрации или маршрутизации. Например, получив на свой порт 1 кадр от компьютера 1, мост делает первую запись в своей адресной таблице: MAC - адрес 1 - порт 1. Если все четыре компьютера данной сети проявляют активность и посылают друг другу кадры, то скоро мост построит полную адресную таблицу сети, состоящую из 4 записей - по одной записи на узел.

После того как мост прошел этап обучения, он может работать более рационально. При получении кадра, направленного, например, от компьютера 1 компьютеру 3, он просматривает адресную таблицу на предмет совпадения ее адресов с адресом назначения 3. Поскольку такая запись есть, то мост выполняет второй этап анализа таблицы - проверяет, находятся ли компьютеры с адресами источника (в нашем случае - это адрес 1) и адресом назначения (адрес 3) в одном сегменте. Так как в нашем примере они находятся в разных сегментах, то мост выполняет операцию *продвижения (forwarding)* кадра - передает кадр на другой порт, предварительно получив доступ к другому сегменту.

Если бы оказалось, что компьютеры принадлежат одному сегменту, то кадр просто был бы удален из буфера и работа с ним на этом бы закончилась. Такая операция называется *фильтрацией (filtering)*.

Если же адрес назначения неизвестен, то мост передает кадр на все свои порты, кроме порта - источника кадра, как и на начальной стадии процесса обучения.

На самом деле мы несколько упростили алгоритм работы моста. Его процесс обучения никогда не заканчивается. Мост постоянно следит за адресами источника буферизуемых кадров, чтобы быть в состоянии автоматически приспосабливаться к

изменениям, происходящим в сети, - перемещениям компьютеров из одного сегмента сети в другой, появлению новых компьютеров. С другой стороны, мост не ждет, когда адресная таблица заполнится полностью (да это и невозможно, поскольку заранее не известно, сколько компьютеров и адресов будут находиться в сегментах моста). Как только в таблице появляется первый адрес, мост пытается его использовать, проверяя совпадение с ним адресов назначения всех поступающих пакетов.

Входы адресной таблицы могут быть динамическими, создаваемыми в процессе самообучения моста, и статическими, создаваемыми вручную администратором сети. Динамические входы имеют срок жизни - при создании или обновлении записи в адресной таблице с ней связывается отметка времени. По истечении определенного тайм-аута запись помечается как недействительная, если за это время мост не принял ни одного кадра с данным адресом в поле адреса источника. Это дает возможность автоматически реагировать на перемещения компьютера из сегмента в сегмент - при его отключении от старого сегмента запись о его принадлежности к нему со временем вычеркивается из адресной таблицы. После включения этого компьютера в работу в другом сегменте его кадры начнут попадать в буфер моста через другой порт, и в адресной таблице появится новая запись, соответствующая текущему состоянию сети.

Статические записи не имеют срока жизни, что дает администратору возможность подправлять работу моста, если это необходимо.

Кадры с широковещательными MAC - адресами передаются мостом на все его порты, как и кадры с неизвестным адресом назначения. Такой режим распространения кадров называется *затоплением сети (flood).* Наличие мостов в сети не препятствует распространению широковещательных кадров по всем сегментам сети, сохраняя ее прозрачность. Однако это является достоинством только в том случае, когда широковещательный адрес выработан корректно работающим узлом. Однако часто случается так, что в результате каких-либо программных или аппаратных сбоев протокол верхнего уровня или сам сетевой адаптер начинают работать некорректно и постоянно с высокой интенсивностью генерировать кадры с широковещательным адресом в течение длительного промежутка времени. Мост в этом случае передает эти кадры во все сегменты, затапливая сеть ошибочным трафиком. Такая ситуация называется *широковещательным штормом (broadcast storm)*.

К сожалению, мосты не защищают сети от широковещательного шторма, во всяком случае, по умолчанию, как это делают маршрутизаторы. Максимум, что может сделать администратор с помощью моста для борьбы с широковещательным штормом - установить для каждого узла предельно допустимую интенсивность генерации кадров с широковещательным адресом. Но при этом нужно точно знать, какая интенсивность является нормальной, а какая - ошибочной. При смене протоколов ситуация в сети может измениться, и то, что вчера считалось ошибочным, сегодня может оказаться нормой. Таким образом, мосты располагают весьма грубыми средствами борьбы с широковещательным штормом.

На рис. 13.2 показана типичная структура моста. Функции доступа к среде при приеме и передаче кадров выполняют микросхемы MAC, которые идентичны микросхемам сетевого адаптера.

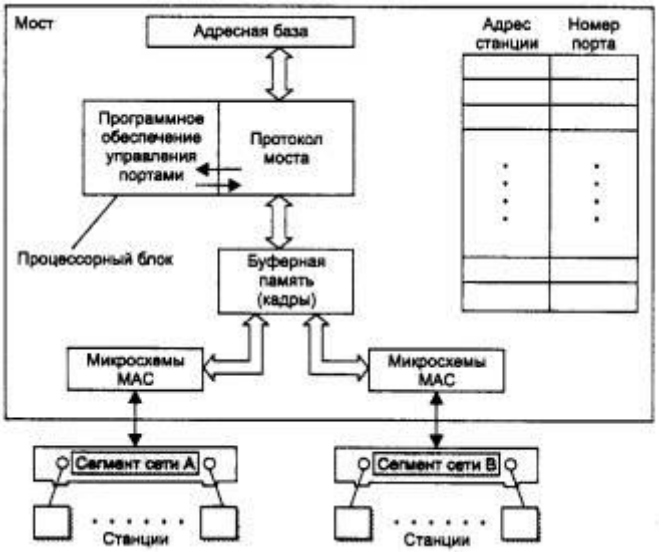


Рис. 13.2 «Структура моста»

На рис. 13.3 показана копия экрана терминала с адресной таблицей модуля локального моста концентратора System 3000 компании SynOptics (сам концентратор уже не выпускается, но в свое время он сыграл роль пионера в становлении многосегментных концентраторов Ethernet на витой паре, причем концентратор имел модуль моста, который мог соединять внутренние сегменты без привлечения внешнего моста). Терминал подключен к консольному порту, и информация на его экране высвечена модулем управления моста.

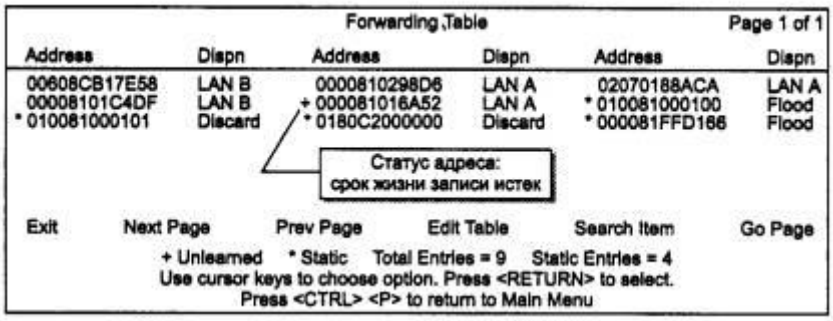


Рис. 13.3 «Адресная таблица моста System 3000 local Bridge»

Из помещенной на экране адресной таблицы (Forwarding Table) видно, что сеть состоит из двух сегментов - LAN A и LAN B. В сегменте LAN A имеются, по крайней мере, 3 станции, а в сегменте LAN B - 2 станции. Четыре адреса, помеченные звездочками, являются статическими, то есть назначенными администратором вручную. Адрес, помеченный знаком «+», является динамическим адресом с истекшим сроком жизни.

Таблица имеет столбец «Dispn» - «Распоряжение», которое говорит мосту, какую операцию нужно проделать с кадром, имеющим данный адрес назначения, Обычно при автоматическом составлении таблицы в этом поле ставится условное обозначение порта назначения, но при ручном задании адреса в это поле можно внести нестандартную операцию обработки кадра. Например, операция «Flood» - «Затопление» заставляет мост распространять кадр в широковещательном режиме, несмотря на то что его адрес назначения не является широковещательным. Операция «Discard» - «Отбросить» говорит мосту, что кадр с таким адресом не нужно передавать на порт назначения.

Собственно, операции, задаваемые в поле «Dispn», являются особыми условиями фильтрации кадров, дополняющими стандартные условия распространения кадров. Такие условия обычно называют *пользовательскими фильтрами*.

ПРАКТИЧЕСКАЯ ЧАСТЬ

1. Запустите приложение NetCracker Professional.
2. Откройте NetCracker Professional (.NET) файл.

Чтобы отобразить диалог открытия, из меню **File** выберите **Открыть**.

В папке **Samples** выберите файл *Tutor.net* и нажмите кнопку **Открыть (**Open**)**, или дважды щелкните на *Tutor.net*.

3. Разверните окно в рабочем пространстве для удобного просмотра.
4. Перейдите на вкладку **Project** из меню **View** выберите **Project Hierarchy.**

Вкладка **Project** показывает иерархическую структуру проекта, начиная с самого верхнего уровня структуры заканчивая зависимыми вложенными уровнями. Для проектов только с одним уровнем будет показываться только верхний уровень. Каждый уровень имеет символ раскрытия списка для раскрытия или свертывание иерархический структуры.

5. Посмотрите на объект «Building» (здание) расположенный с левой стороны окна, и сделайте двойной щелчок по нему.

Рис. 13.1 «Контейнерный объект»

На экране появится окно «Building».

6. Теперь вернитесь обратно к главному окну, выбрав команду **Top** из меню **Window**.
7. Чтобы отобразить оба окна в рабочем пространстве выберите команду **Cascade** из меню **Window**.
8. Впишите рабочую область каждого из окон в окно приложения используя кнопки **Zoom**. Ваше рабочее пространство может выглядеть следующим образом:

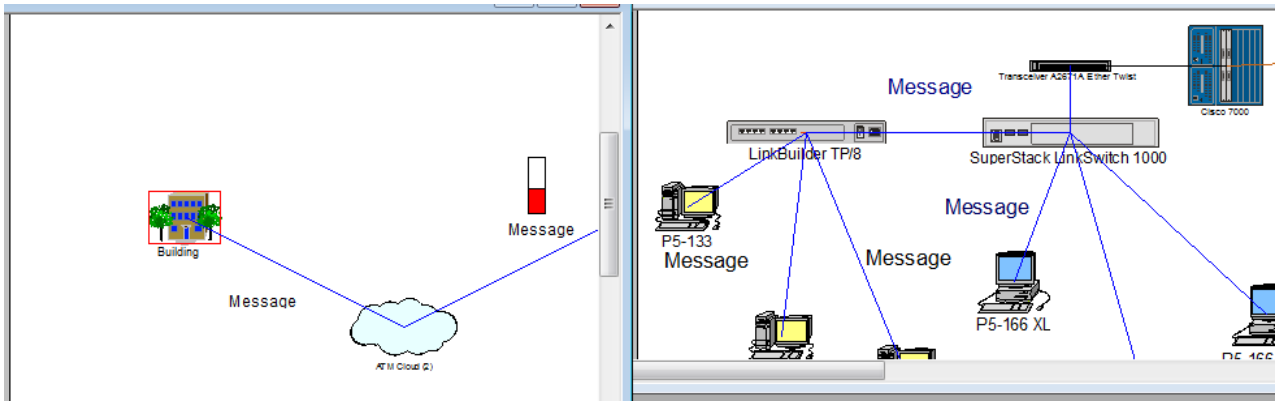


Рис. 13.1 «Многоуровневый проект»

Теперь закройте Top, нажав кнопку **Close**.

9. Повторно откройте его, дважды щелкнув на Top в browser.

Отрегулируйте расположение и видимость окна с помощь полос прокрутки и кнопок

масштабирования.

10. Переименуйте окно.

a. Сначала, сделайте окно Top активным, нажав на него.

b. Теперь обратитесь к диалогу **Site Setup** одним из двух способов:

c. Из меню **Sites** выберите команду **Site Setup**.

d. На заднем плане окна Top щелкните правой кнопкой мыши, чтобы отобразить локальное меню и выберите команду **Site Setup**.

В диалоговом окне Site Setup выберите вкладку **Names**. Выделите название («Top») в поле имени окна и напечатайте название для проекта.

Нажмите кнопку **OK**, чтобы применить ваши изменения и закрыть диалоговое окно.

Переименуйте окно «Building», повторив действия, указанные в шагах 10 (a-d) для Building.

Если новое название нечитаемо, значит надо поменять шрифт для правильного отображения русских букв:

щёлкните правой кнопкой мыши на названии и выберите **Properties**. В списке шрифтов выберите что-нибудь вроде «Arial Cyr», «Courier New Cyr», «Times New Roman Cyr» или любой другой шрифт, при котором текст будет читаться в окошке Sample (образец):

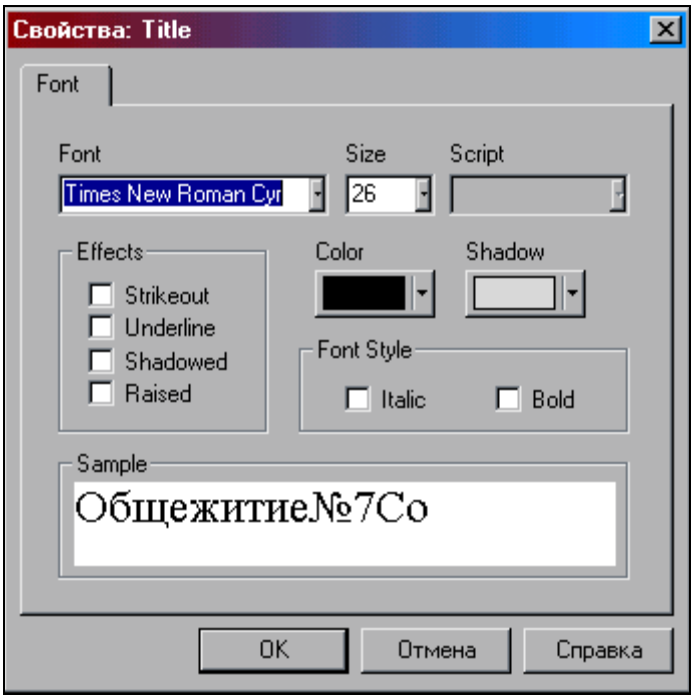


Рис. 13.2 «Изменение шрифта для русского текста»

Новые названия появятся в заголовках и командах меню **Window**.

11. Использование инструментальных средств рисования для аннотирования проекта.

a. Сделайте одно из окон текущим.

b. На инструментальной панели **Modes**, нажмите на кнопку режима **Draw**  . Появится панель рисования:

Рис. 13.3 «Панель рисования»

На панели рисования нажмите кнопку **Line**. Используйте инструмент **Line**, чтобы нарисовать стрелку, которая указывает на верхний правый угол окна. Перейдите к стандартному режиму, нажав на кнопку со стрелкой.

12. Измените цвет и толщину стрелки, которую вы начертили:

выберите начерченную линию

- ✓ войдите в меню **Object → Styles → Draw color**,
- ✓ выберите нужный цвет, толщину, тип линии и нажмите кнопку **OK**.
- ✓ повторите это для каждого сегмента стрелки.

Чтобы создать подпись к стрелке делайте следующее:

- ✓ на инструментальной панели **Modes** нажмите кнопку **Draw**,
- ✓ на Панели рисования выберите инструмент **Текст**,
- ✓ выделите над стрелкой прямоугольник, в котором будет находиться текст,
- ✓ напечатайте *«Выход на Студгородок»* и нажмите клавишу ENTER.

Отредактируйте свойства шрифта в надписи, щёлкнув по ней правой кнопкой мыши, через пункт меню **Proprties**. Измените шрифт на тот, который отображает русские буквы и размер шрифта на 20.

Вернитесь к стандартному режиму, нажав кнопку со стрелкой на панели **Modes.**

13. Определим путь прохождения трафика от одного устройства к другому в пределах окна, используя режим **Trace.**

Запустите анимацию, нажимая кнопку **Пуск.**

В двух видимых окнах, Вы можете видеть трафик, текущий от индивидуальных рабочих станций в «Общежитие», сквозь *Cisco* маршрутизатор в «Студгородок». Из «Студгородок» в «Общежитие» также движутся пакеты.

На инструментальной панели **Modes**, нажмите кнопку режима **Trace**  , щёлкните мышкой рабочую станцию *P5-166 XL(3)* в крайнем правом углу окна «Общежитие», затем щёлкните мышкой рабочую станцию с левой стороны (*P5-133XL (3)*).

Путь прохождения трафика между этими рабочими станциями подсвечивается красным цветом.

14. Определим теперь путь прохождения трафика, текущего от устройства в одном окне к объекту в другом окне.

Нажмите кнопку режима **Trace**, нажмите на крайнюю левую рабочую станцию (*P5-133 XL(3)*) в окне «Общежитие».

Теперь нажмите на *Building (2)* в окне «Студгородок».

Путь прохождения трафика между двумя объектами подсвечивается красным цветом.

Остановите анимацию, щелкнув кнопку **Stop**, чтобы лучше увидеть подсвеченный путь.

Перейдите к стандартному режиму, нажав на кнопку со стрелкой (standard mode).

15. Закройте текущий проект без его сохранения, выбирая **Close** из меню **File**, либо сохраните его под другим именем, выбирая **Save As**.

16. Создайте новый проект через меню **File → New.**

17. В browser устройств нажмите вкладку **Devices**. Щёлкните на **Buildings, campuses and LAN workgroups** (самый верхний пункт списка).

В панели «Изображения» появятся изображения зданий, университетских городков и рабочих групп LAN.

18. Выберите одно из изображений объекта Building из панели «Изображения» и перетащите в окно Top.

19. Выделите объект Building. Сделайте его контейнером для подсети Building, выполнив одно из:

Щелкните правой кнопкой мыши, чтобы открыть локальное меню и выберите команду **Expand**,

Из меню **Object** выберите команду **Expand**.

Вы создали многоуровневый сетевой проект, который включает верхний уровень и второй уровень в объекте Building. Изображение объекта Building в окне Top показывается с красным контуром вокруг него, указывая, что это вложенный объект.

чтобы увидеть иерархическую структуру, в броузере выберите вкладку **Project Hierarchy**.

---

20. Давайте завершим проект, заполнив архитектуру клиент/сервер использования здания.

Мы будем использовать прежде всего *универсальные устройства***, которые предконфигурированы. Универсальные устройства включены в базу данных устройств NetCracker Professional.**

В броузере нажмите вкладку **Devices**, затем в броузере устройств выделите **LAN workstation**.

Универсальные рабочие станции будут отображены в панели «Изображения».

Рис. 13.4 «Типовые изображения рабочих станций»

В панели «Изображения» выберите и перетащите рабочую станцию *Ethernet* в окно объекта Building.

рабочая станция *Ethernet* уже конфигурирована с платой адаптера LAN.

Из меню **Edit**, выберите **Duplicate**.

В броузере устройств выделите **Switches**.

Универсальный *коммутатор Ethernet* отображен в панели «Изображения».

Рис. 13.5 «Типовое изображение устройства коммутатора»

В панели «Изображения» выберите **Ethernet Switch** и перетащите его в окно объекта Building.

Щелкните по кнопке связь устройств  .

Щелкните по рабочей станции и перетащите связь к коммутатору. Отпустите левую кнопку мыши.

Появится диалог помощника связи. В окне диалога помощника связи, нажмите кнопку **Link**, затем нажмите кнопку **Close**.

Повторите то же для другой рабочей станции.

Сделайте окно Top текущим окном, щёлкнув по нему мышью.

Перейдите к Стандартному режиму, затем выберите **Buildings → Campuses and LAN workgroups** в броузере устройств.

В панели «Изображения» появятся здания, университетские городки и изображения устройства рабочей группы LAN.

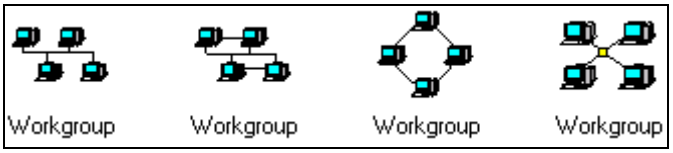


Рис. 13.6 «Изображения универсальных устройств рабочих групп»

Выберите и перетащите изображение устройства рабочей группы из панели «Изображения» в окна Top

Чтобы связать рабочую группу с объектом Building в окне Top, на инструментальной панели **Modes** выберите инструмент связи устройств, нажмите на рабочую группу, затем нажмите на значок Building.

пунктир указывает, что эта связь не завершена!

Перейдите в стандартный режим и сделайте двойной щелчок на значке Building в окне Top.

Окно Building становится текущим окном.

На инструментальной панели **Modes** выберите кнопку связи устройств. Щелкните в окне Building на значке разъема  , затем на коммутатор, чтобы завершить подключение.

Появится окно диалога помощника связи.

значок разъема обычно располагается в углу окна. Если необходимо, можно значок разъема увеличить или расположить по удобству.

Выберите порт *Ethernet* в панели опции Switch Port Configuration, нажмите кнопку **Lin**k, затем нажмите кнопку **Close**:

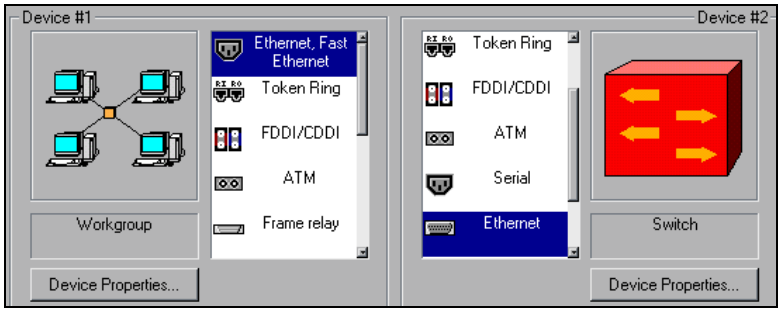


Рис. 13.7 «Выбор портов в окне конфигурации соединения»

Связь Building с рабочей группой закончена.

21. Сделайте одну из рабочих станций сервером следующим образом:

- В броузере Устройств пролистайте до вкладки “**Network and enterprise software**” (“Сеть и программное обеспечение предприятия”) и разверните ее, нажимая на знак плюс (+). Нажмите на “**Server software**” (“Программное обеспечение Сервера”). Имеющиеся типы серверов теперь отображены в панели «Изображения».

- Перетащите *E-mail server* (сервер электронной почты) на рабочую станцию. Указатель должен измениться на стрелку со знаком "плюс", который означает что, Вы можете установить это программное обеспечение на этот компьютер.

22. Создайте трафик клиента/сервера по следующим шагам:
    - На инструментальной панели **Modes** нажмите кнопку **Set Traffic**.
    - В окне Building нажмите на Рабочую станцию без программного обеспечения сервера, затем в том же самом окне нажмите на Рабочую станцию с программным обеспечением сервера.
    - Выберите **E-mail** и нажмите кнопку **Assign**.
23. Назначьте другой трафик по следующим шагам:
    - В окне Top нажмите на изображение Рабочей группы, затем в окне Building нажмите на Рабочую станцию с программным обеспечением сервера.
    - Выберите **Small office** как тип трафика и нажимите кнопку **Assign**.

    Запустите анимацию, нажав кнопку **Start** на инструментальной панели **Control**.

    Чтобы остановить анимацию, нажмите кнопку **Stop**.
24. Из меню **File** выберите команду **Save**.

    Так как Вы ещё не сохраняли этот файл, появится диалог сохранения.
25. В поле имени отображено заданное по умолчанию имя файла *Net1.net*. Введите свое имя и нажмите **Save** (Сохранить). Расширение *.NET будет добавлено автоматически.
26. Чтобы закрыть этот проект из меню **File** выберите команду **Close**.

Задание на лабораторную работу

Многоуровневый сетевой проект (гиперсеть) представляет собой три отдельные сети (соединенные между собой мостами на базе технологий Fast Ethernet, которые находятся в отдельных зданиях. Шаблон лабораторной работы изображен на рис. 13.9.

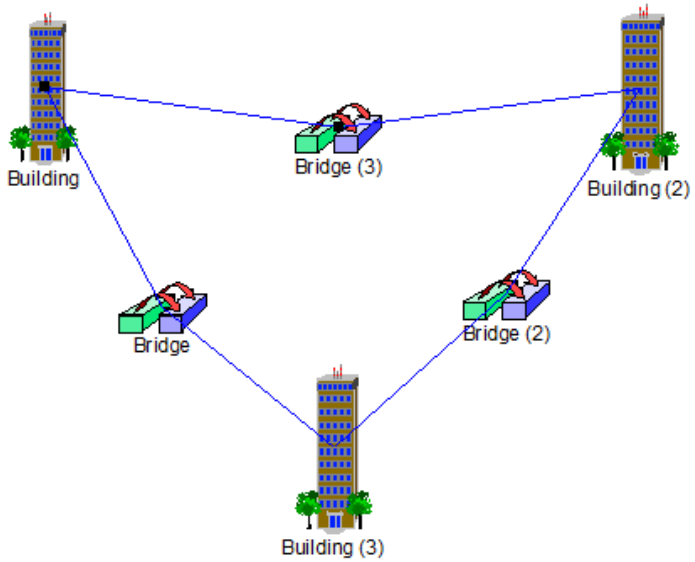


Рис. 13.9 «Шаблон лабораторной работы»

Гиперсеть состоит из трех «зданий», в каждом из которых находится локальная сеть (рис. 13.10 -13.12). Эти «здания» объединены между собой мостами.

Мостом называется устройство, которое служит для *связи* между двумя локальными сетями. Мост передаст кадры из одной сети в другую. Эти устройства достаточно интеллектуальны - не повторяют шумы сети, ошибки или испорченные кадры. Для каждой соединяемой сети мост является узлом абонентом сети. Узлом сети также может быть компьютер, специальная рабочая станции или другое устройство. Доступ к среде осуществляется в соответствии с теми же правилами, что и для обычного узла.

По принадлежности к разным типам сетей различают локальные и глобальные (удаленные) мосты. Одним из самых важных достоинств локальных мостов является их способность соединять локальные сети, использующие разные среды.

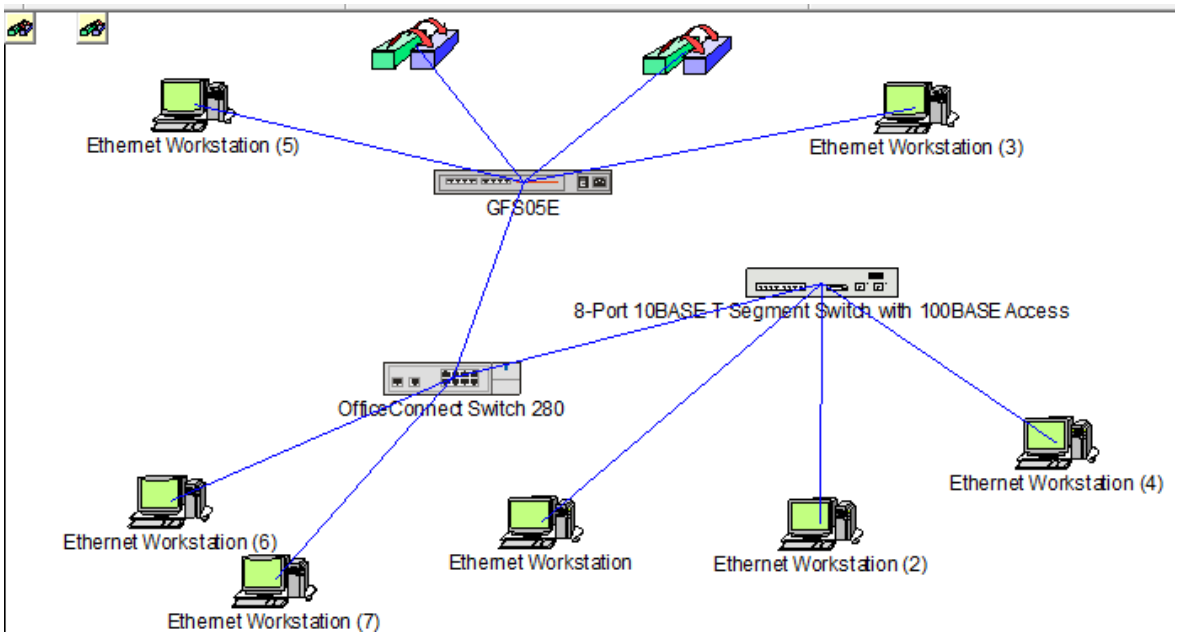


Рис. 13.10 «Сеть первого здания»

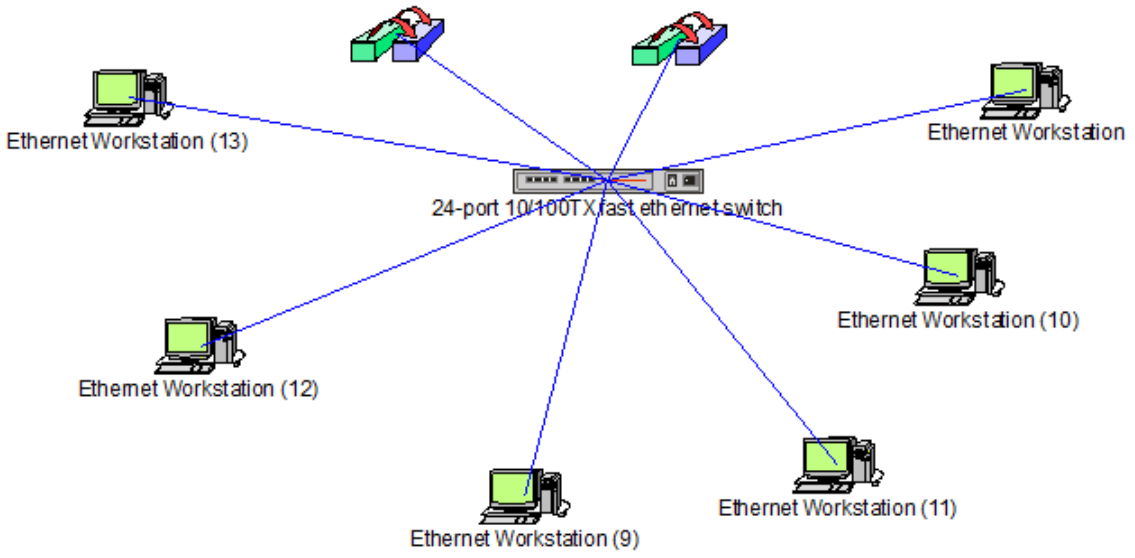


Рис. 13.11 «Сеть второго здания»

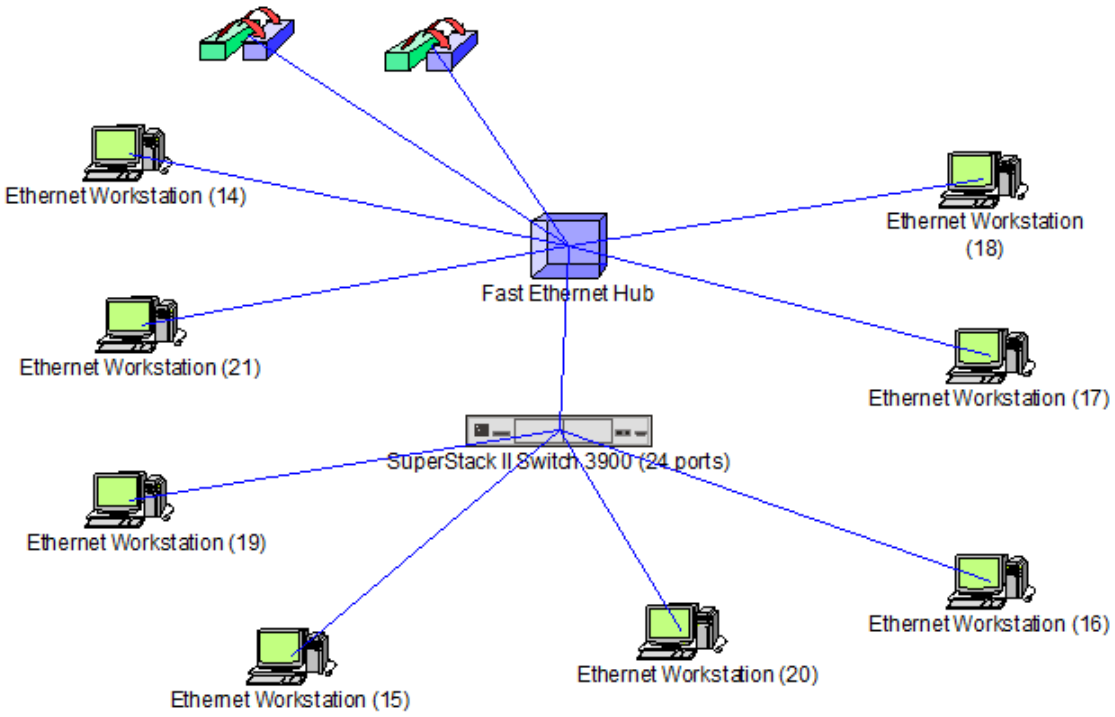


Рис. 13.12 «Сеть третьего здания»

Глобальные мосты устанавливаются и сетях передачи информации на большие расстояния. При этом глобальные мосты могут быть оборудованы локальными портами. Термин «прозрачные» мосты объединяет большую группу устройств. Если рассматривать устройства этой группы с точки зрения решаемых ими задач, то эту группу можно разделить на три подгруппы:

1) прозрачные мосты объединяют сети с едиными протоколами канального и физического уровней модели OSI (Etherned-Hetherned, Tokening- Token Ring);

2) транслирующие мосты объединяют сети с различными протоколами канального и физического уровней;

3) инкапсулирующие мосты соединяют сети с единым протоколами канального и физического уровня (например, Ethernet) через сети с другими протоколами (например, FDDI).

Задание на практическую работу

Варианты заданий приведены в табл. 13.1. Параметры трафика необходимо взять из практической работы №11.

На мостах измерить и сопоставить следующие величины: среднюю задержку, среднюю рабочую нагрузку, количество пакетов за последнюю секунду. На линиях связи измерить среднюю рабочую нагрузку. Разорвать линию связи одного моста и проследить за изменением величин на других мостах.

Таблица 13.1

Длина входного и выходною буфера	
fnbound buffer length	Outbound buffer length
Для первого моста	
100 Kbytes	100 Kbytes

10 Kbytes	10 Kbytes
50 Kbytes	50 Kbytes
30 Kbytes	30 Kbytes
120 Kbytes	50 Kbytes
Для второго моста	
5 Kbytes	10 Kbytes
25 Kbytes	15 Kbytes
35 Kbytes	35 Kbytes
120 Kbytes	50 Kbytes
100 Kbytes	100 Kbytes
Для третьего моста	
25 Kbytes	15 Kbytes
12Kbyles	25 Kbytes
30 Kbytes	30 Kbytes
120 Kbytes	50 Kbytes
5 Kbytes	10 Kbytes

Контрольные вопросы

1. Как работает мост?
2. В чем отличие моста от репитера?
3. Что такое прозрачный мост?
4. Какие бывают разновидности мостов?

Практическая работа №14

Разработка проекта локальной компьютерной сети.

Цель работы: Научиться проектировать локальные компьютерные сети.

В результате выполнения практической работы студент должен:

Знать:

- принципы организации и функционирования локальных сетей.

Уметь:

- проектировать локальные компьютерные сети;
- применять ранее полученные навыки работы в Net Cracker.

Задание для практической работы:

1. Выполнить практическую часть
2. Ответить на контрольные вопросы
3. Оформить отчет.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Техническое задание

Разработать проект локальной компьютерной сети (ЛКС), объединяющей три рабочих группы на основе технологии Ethernet. Рабочие группы содержат 8, 11 и 12 рабочих станций на базе Pentium 4, Pentium 3, Pentium 2, Celeron и AMD процессоров. ЛКС должна имеет выделенные серверы: файл-сервер, сервер баз данных и Интернет-сервер, расположенные в разных сегментах. Каждая рабочая станция в рабочих группах должна иметь доступ ко всем серверам. Также необходимо организовать подключение ЛКС к сети Internet по каналу ISDN. Предполагается следующая специализация рабочих групп:

- рабочая группа №1 – разработчики системного программного обеспечения (ПО);
- рабочая группа №2 – разработчики прикладного ПО;
- рабочая группа №3 – разработчики тестирования ПО.

Основные виды трафика в сети – внутренний (локальный), работа с файл- сервером, работа с сервером баз данных и работа в Internet. Необходимо осуществить моделирование полученного проекта сети, используя программную оболочку NetCracker Professional, и сформировать протоколы результатов моделирования.

Анализ зоны проектирования и информационных потоков

Зона проектирования располагается на территории ≈200 м^2. В связи со специализацией отделов (рабочих групп) для каждого из них выделяется помещение площадью ≈60 м^2; высота потолков – 2,8 м. Для серверов выделяется отдельное помещение площадью ≈12 м^2. Примерный план расположения всех отделов приведен на рис. 14.1

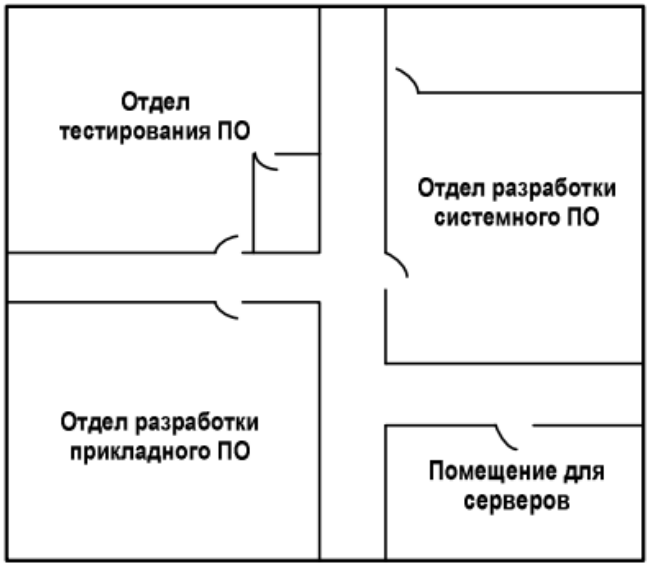


Рис.14.1 «План расположения отделов»

Количество рабочих станций, распределяемых по отделам:

- отдел тестирования ПО – 11 рабочих станций;
- отдел разработки системного ПО – 8 рабочих станций;
- отдел разработки прикладного ПО – 12 рабочих станций;
- помещение для серверов – один файл-сервер, один сервер баз данных, web-server, маршрутизатор и рабочая станция администратора.

Отдел тестирования ПО занимается разработкой программного обеспечения (тестов) для отделов системного и прикладного ПО. Обмен информацией между отделами происходит либо непосредственно, либо через файл-сервер. Вся необходимая информация (базы данных) хранится на сервере баз данных, доступ к которому имеют все рабочие группы. К файл-серверу доступ имеют также все рабочие группы. В задачи администратора входит:

- мониторинг сети;
- установка программного обеспечения и настройка его на рабочих станциях и серверах;
- администрирование файл-сервера;
- администрирование сервера баз данных.

Из проведенного выше анализа следует, что циркулирующий в сети трафик не большой и представляет собой совокупность информационных потоков разного профиля (в основном – LAN, File server client, Database client и Internet client).

Анализ топологии сети и сетевой технологии

В соответствии с исходными параметрами проектируемой сети в качестве базовой топологии целесообразно выбрать топологию многоуровневой звезды. Главное преимущество данной структуры – это высокая надежность и простота реализации на базе коммутаторов. На рис. 12.2 приведена структурная схема сети.

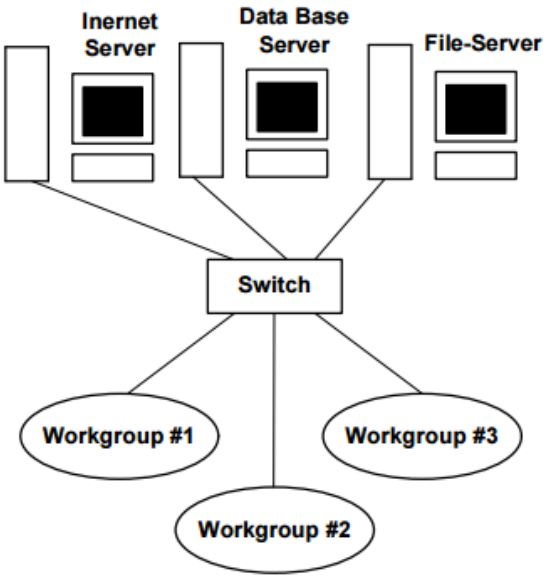


Рис.14.2 «Структура сети»

Основным устройством в проектируемой сети, в данном случае, является коммутатор (Switch), который выполняет основные функции по коммутации и передаче кадров. Каждый порт коммутатора содержит специализированный процессор, который обрабатывает кадры. Организация каждой рабочей группы имеет типовую схему, представленную на рис. 14.3.

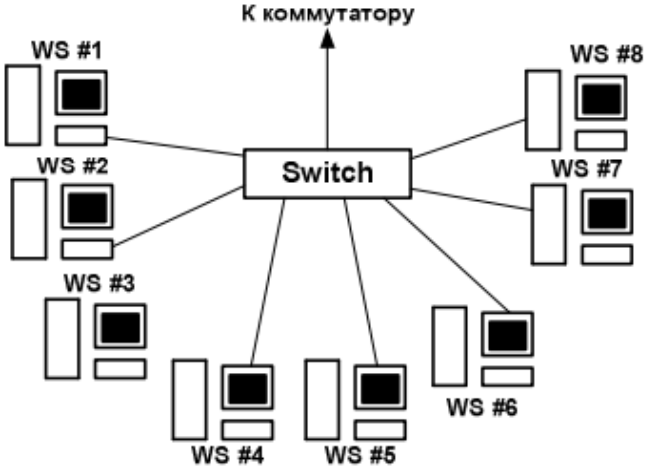


Рис. 14.3 «Организация рабочей группы»

Организационно сеть состоит из трех сегментов (по числу отделов), взаимодействующих друг с другом через коммутатор верхнего уровня, который образует магистраль сети (магистраль, стянутая в точку) и объединяет все сегменты. К данному коммутатору также подключены сервер баз данных, файл- сервер и Интернет-сервер. Первый сегмент (Workgroup #1) содержит рабочие станции отдела тестирования ПО. Все рабочие станции подключены через коммутатор второго уровня к вышележащему уровню. Второй и третий сегменты (Workgroup #2, Workgroup #3) включают в себя рабочие станции соответственно отделов системного ПО и прикладного ПО. Эти сегменты так же, как и первый, подключены через коммутаторы сегментов к магистральному коммутатору. Проведем анализ наиболее использующихся сетевых технологий, применяемых для построения локальных сетей.

Ethernet

10-Мегабитный Ethernet устраивал большинство пользователей на протяжении около 15 лет. Однако в начале 90-х годов начала ощущаться его недостаточная пропускная способность. Теперь же у мощных клиентских станций с процессорами Pentium и шиной PCI (133 Мбайт/с) эта доля упала до 1/133, что явно недостаточно. Поэтому многие сегменты 10-Мегабитного Ethernet'а стали перегруженными, реакция серверов в них значительно упала, а частота возникновения коллизий существенно возросла, еще более снижая84 номинальную пропускную способность. Кроме того, данная технология подразумевает использование коаксиального кабеля в качестве среды передачи. В настоящий момент времени данный тип кабеля весьма неудобен в использовании из-за своей низкой пропускной способности и низких эксплуатационных качеств.

FastEthernet

Технология Fast Ethernet является эволюционным развитием классической технологии Ethernet. Ее основными достоинствами являются:

- увеличение пропускной способности сегментов сети до 100 Мб/с;
- сохранение метода случайного доступа Ethernet;
- сохранение звездообразной топологии сетей и поддержка традиционных сред передачи данных - витой пары и оптоволоконного кабеля.

Указанные свойства позволяют осуществлять постепенный переход от сетей 10Base-T – наиболее популярного на сегодняшний день варианта Ethernet – к скоростным сетям, сохраняющим значительную преемственность с хорошо знакомой технологией: FastEthernet не требует переобучения персонала и замены оборудования во всех узлах сети. Официальный стандарт 100Base-T (802.3u) установил три различных спецификации для физического уровня (в терминах семиуровневой модели OSI) для поддержки следующих типов кабельных систем:

- 100Base-TX для двухпарного кабеля на неэкранированной витой паре UTP Category 5, или экранированной витой паре STP Type 1;
- 100Base-T4 для четырехпарного кабеля на неэкранированной витой паре UTP Category 3, 4 или 5;
- 100Base-FX для многомодового оптоволоконного кабеля.

Gigabit Ethernet

Скорость обмена повысилась до 1000 Мбит/с, сохранился формат кадра Ethernet и метод доступа CSMA/CD с минимальными изменениями. В качестве физической среды передачи определены следующие типы:

- одномодовый волоконно-оптический кабель;
- многомодовый волоконно-оптический кабель 62,5/125;
- многомодовый волоконно-оптический кабель 50/125;
- двойной коаксиал с волновым сопротивлением 75 Ом.

Из всех вышеперечисленных вариантов сетевых технологий Ethernet нам не подходит в силу малой пропускной способности и проблем с обслуживанием сети. Gigabit Ethernet обеспечивает самую высокую скорость передачи данных, но в качестве физической среды передачи использует достаточно дорогие носители (цена сетевых адаптеров тоже достаточно высока). К тому же такая высокая скорость обмена необходима лишь для построения достаточно крупных локальных сетей, в которых мощные серверы и магистрали нижних уровней сети работают на скорости 100 Мбит/с, а магистраль Gigabit Ethernet объединяет их,

обеспечивая достаточно большой запас пропускной способности. В нашем случае сеть невелика; скорость обмена 100 Мбит/с – достаточная для полноценного обмена (она позволяет организовывать даже обмен мультимедийным трафиком). К тому же цена витой пары не сопоставима с ценой оптоволокна (сеть на витой паре (к примеру, UTP5) получается намного дешевле, чем на оптоволокне). Проанализировав соотношение цена/качества, мы пришли к выводу, что в качестве сетевой технологии для проектирования сети будем использовать Fast Ethernet.

Выбор активного сетевого оборудования

Рынок коммутаторов сегодня очень обширен, поэтому остановимся только на некоторых популярных моделях коммутаторов различного класса. Обычно коммутаторы делят в первую очередь на классы в соответствии с их областями применения – настольные коммутаторы, коммутаторы рабочих групп, коммутаторы отделов и магистральные (корпоративные коммутаторы). У каждого класса коммутаторов есть свои отличительные признаки. Коммутаторы отделов:

- модульное исполнение;
- поддержка нескольких протоколов;
- встроенные средства обеспечения отказоустойчивости:
- избыточные источники питания;
- модули hot-swap. - пользовательские фильтры;
- поддержка виртуальных сегментов;

Коммутаторы магистралей зданий:

- те же свойства, что и у коммутаторов отделов;
- шасси с большим количеством слотов (10 - 14);\
- внутренняя пропускная способность 1 - 10 Гб/с;
- поддержка 1 - 2 протоколов маршрутизации (локальные интерфейсы) для образования виртуальных сетей.

Рассмотрим примеры некоторых продуктов компаний 3Com, Hewlett-Packard и Cisco, поддерживающих технологию Fast Ethernet.

Коммутатор HP ProCurve Switch 1600M фирмы Hewlett-Packard имеет 16 портов RJ-45 с автоопределением скорости 10/100Base-TX, открытый модульный слот, RS-232C DB-9 консольный порт RJ-45, пропускная способность - 3,87 миллионов пакетов в сек (64 байтных).

Пропускная способность внутренней магистрали - 3,5Гбит/с, емкость таблицы адресов - 10000. Управление осуществляется HP Top Tools for Hubs & Switches, поддерживает протоколы SNMP, RMON.

Коммутатор Cisco 1548M Micro Switch 10/100 компании Cisco Systems представляет собой недорогое, в пересчете на порт, решение масштаба отдела. Этот коммутатор имеет 8 портов RJ-45 с автоопределением скорости 10/100Base-TX, консольный порт для начальной настройки и конфигурирования, поддерживает 4096 MAC – адресов, поддерживается алгоритм Spanning Tree Protocol (STP) и SNMP-управление (Simple Network Management Protocol). Имеется возможность удаленного администрирования с помощью протокола Remote Monitoring (RMON). Коммутатор поддерживает 4 виртуальные сети и позволяет фильтровать трафик по адресу источника и адресу назначения.

Коммутатор можно использовать автономно, а можно объединять в модули несколько коммутаторов, с помощью поставляемого специального приспособления.

Такие опции, как поддержка Cisco Discovery Protocol (CDP) и поставляемое вместе с коммутатором программное обеспечение Cisco ConfigMaker, значительно облегчают процесс настройки и конфигурирования сети.

Коммутатор Super Stack II Baseline 10/100 фирмы 3COM предназначен для сети с числом компьютеров более 10. Ключевые особенности коммутаторов Super Stack II Baseline 10/100:

- 12 или 24 портов RJ-45 10Base-T/100Base-TX с автоматическим определением скорости передачи обеспечивают высочайшую скорость коммутируемого соединения.
- Таблица MAC - адресов дают возможность поддерживать до 4000 устройств локальной вычислительной сети.
- Функция управления протоколом IEEE 802.3x гарантирует отсутствие потерь пакетов в высокоскоростных дуплексных соединениях во время пиков трафика.
- Размер устройства обеспечивает легкость установки в стойку с помощью поставляемого комплекта для монтажа. Устройство может также использоваться автономно.
- Диагностические индикаторы показывают состояние сети и статус каждого порта, облегчая поиск и проверку статуса каждого порта.
- Возможность подключения резервной системы питания обеспечивает надежную защиту от простоев в сети.

Коммутатор SuperStack II Switch 3300 фирмы 3COM обеспечивает 24 коммутируемых порта с автоматической установкой скорости 10/100 Мбит/с на каждом порту в зависимости от присоединенных устройств. Так же имеется возможность в установке высокоскоростных модулей, для этого на задней панели имеется дополнительный слот. Еще на задней панели коммутаторов SuperStack II Switch 3300 имеется встроенный разъем Matrix Port. Два коммутатора можно соединять между собой, подключив к этим портам специальный кабель. Если необходимо объединить более двух устройств, в слот для скоростного модуля на одном из объединенных коммутаторов нужно установить SuperStack II Switch Matrix Mobule, к которому специальными кабелями следует подключить все остальные коммутаторы. Таким образом, можно получить один большой виртуальный коммутатор общей емкостью до 110 портов.

Исходя из соотношения цена/производительность, для разрабатываемой локальной сети в качестве активного сетевого оборудования целесообразно выбрать:

- **Коммутатор** SuperStack II Baseline 10/100 Switch 12 ports компании 3Com
- **Коммутатор** SuperStack II Switch 3300 (24 ports) компании 3Com

Выбор сетевого адаптера

В качестве сетевых аппаратных средств рабочих станций и сервера выбираем сетевые платы 10/100 TX PCI UTP компании Compaq Computer.

Этот адаптер предназначен для работы по технологиям 10Base-T или 100Base-TX. Адаптер обладает свойством чувствительности к скорости порта, к которому он подключен, поэтому выбор скорости работы - 10 Мб/с или 100 Мб/с - происходит автоматически. Адаптер имеет один порт RJ-45 и может работать с кабелем UTP Category5. Выпускаются варианты адаптера для шины EISA и шины PCI. Адаптер не имеет переключателей и не

требует ручного конфигурирования перед установкой в компьютер. Адаптер имеет низкий коэффициент использования ресурсов центрального процессора. На карте имеется буферная память, конфигурирования перед установкой в компьютер.

Выбор типа сервера

Основная работа File Server заключается в хранении, приёме и передаче файлов, поэтому для файлового сервера требуется оборудование, способное удовлетворить основным требованиям, а именно: высокая скорость записи и чтения данных с накопителей на жестких дисках, высокая скорость внутренней шины компьютера и достаточный объем оперативной памяти для организации кэширования используемых пользователями данных.

Для Database Server оборудование должно быть более производительным в плане вычислительной мощности. Поскольку при формировании запросов на операции с данными требуется производить вычисления, то необходимо оснастить сервер баз данных помимо высокоскоростных накопителей и большого объема памяти и мощным процессором.

Исходя из этого, подходящим будет следующее аппаратное обеспечение:

Сервер Cosmos II-Pentium II-400MHz компании DTK Computer.

На данный сервер установлен File server.

Сервер имеет следующие параметры: процессор – Intel Pentium II 400 МГц, оперативная память – 256 Мб, системная шина ввода/вывода – PCI.

Сервер Cosmos III-Pentium III-550MHz компании DTK Computer.

На данный сервер установлен Small Office Database Server и SQL Server.

Сервер имеет следующие параметры: процессор – Intel Pentium II 400 МГц, оперативная память – 512 Мб, системная шина ввода/вывода – PCI.

Сервер Cosmos II-Pentium II-400MHz компании DTK Computer.

На данный сервер установлен HTTP Server, E-mail server, FTP server, Security Server. Сервер имеет следующие параметры: процессор – Intel Pentium II 400 МГц, оперативная память – 256 Мб, системная шина ввода/вывода – PCI.

Проектирование

Создадим проект в среде проектирования сетей NetCracker Professional согласно техническому заданию. В любом графическом редакторе (в нашем случае использовался Visio) рисуем план расположения отделов, который будем использовать в виде карты (подложки) для построения сети. В главном меню NetCracker выбираем **Site → Site Setup**. В появившемся окне выбираем закладку **Background** и устанавливаем флажок **Map**. Активизируется диалог **Selected map file**, в котором при помощи кнопки **Browse…** выбираем наш графический файл плана расположения отделов. Далее заполняем устройствами созданный проект. Устанавливаем соединения. Для наглядности проекта можно использовать изогнутые связи.

Задаем трафик. Всё готово, теперь можно переходить к моделированию. Проект сети представлен на рис. 14.4.

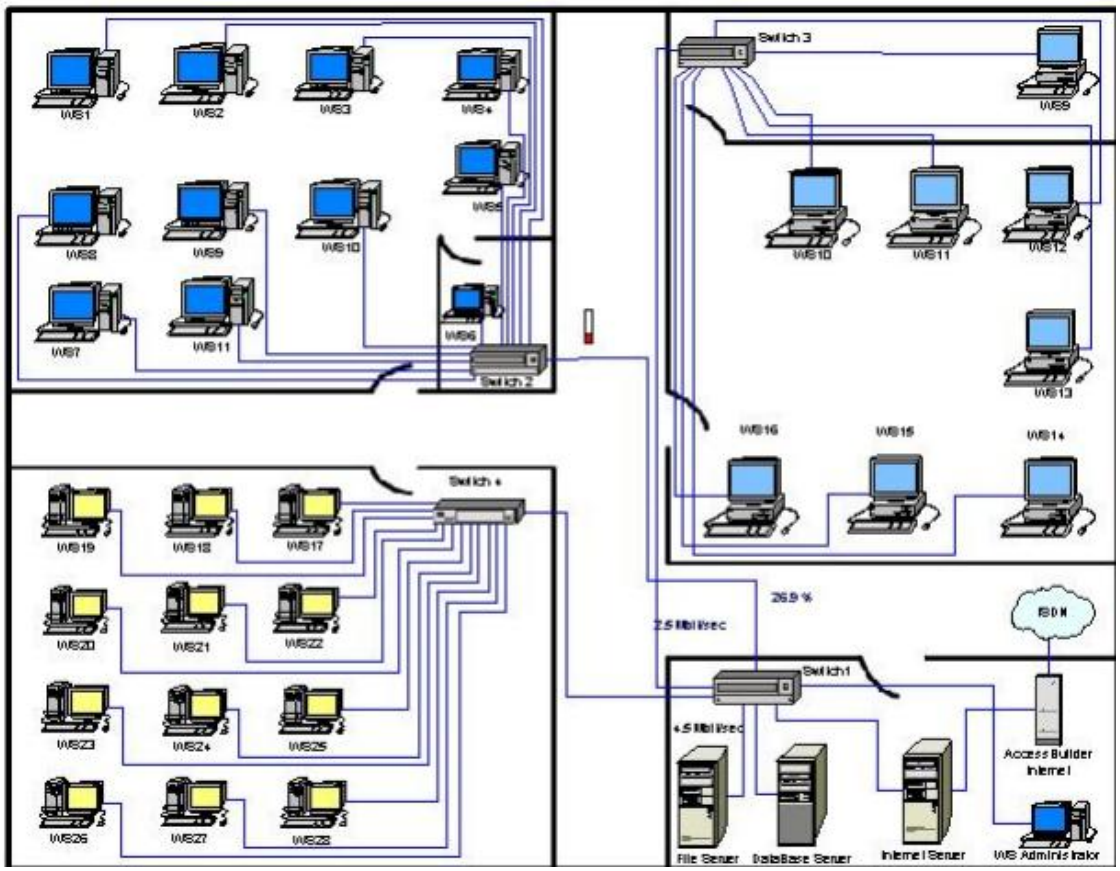


Рис.14.4 «Проект сети»

Моделирование и результаты

Используя программную оболочку NetCracker Professional, проведём моделирование полученного проекта сети. Графическое представление полученной сети в режиме моделирования с отображением статистики приведено на рис. 14.5.

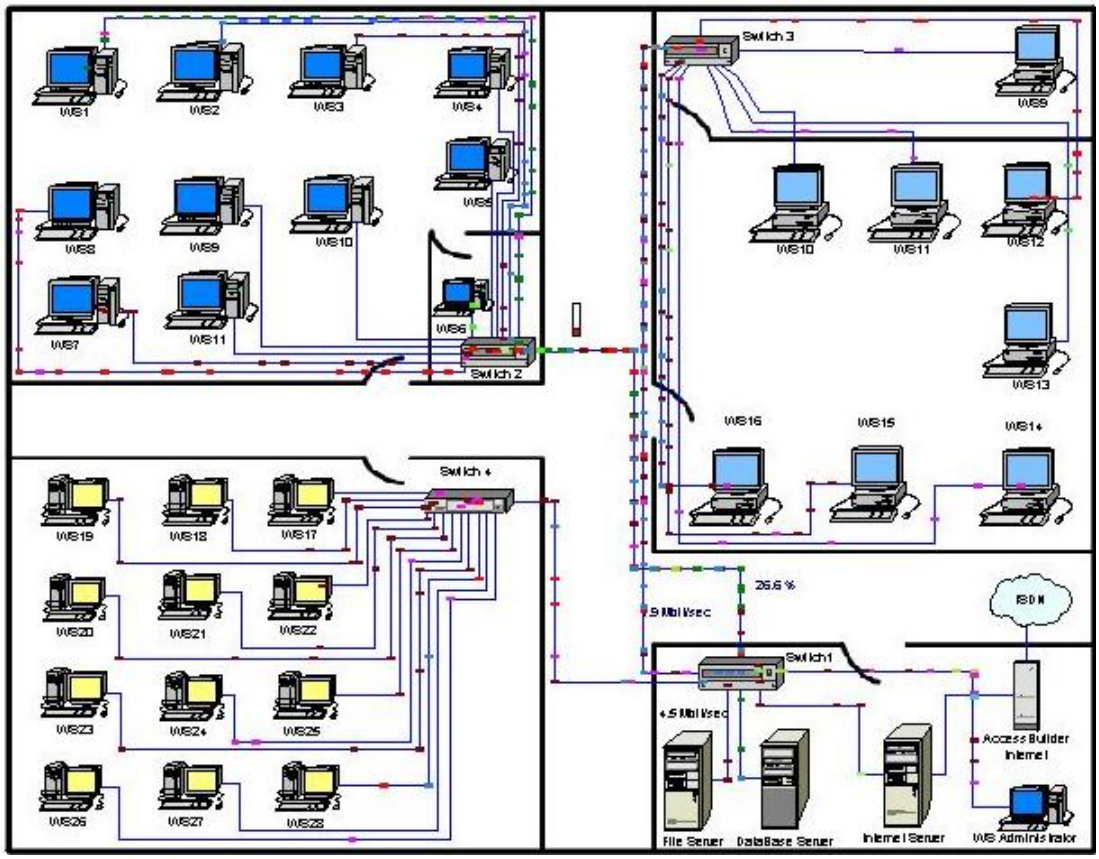


Рис.14.5 «Моделирование проекта сети»

Результаты моделирования демонстрируют работоспособность разработанной сети и практическую реализацию проекта, поскольку он выполнен на основе реального оборудования ведущих производителей из базы данных NetCracker. В процессе моделирования были установлены типовые трафики, характерные для рабочих групп в соответствии с их специализацией. Спроектированная сеть обеспечивает свободное прохождение пакетов, при этом она не перегружена, а имеет запас как по производительности, так и по оборудованию, что позволяет при необходимости наращивать сеть.

Перечень используемого оборудования (DeviceSummary)

Device	Device name	Vendor	Model
AccessBuilder Internet	AccessBuilder Internet	3Com Corp.	AccessBuilder Internet 400, U
DataBase Server	DataBase Server 10/100 TX PCI UTP	DTK Computer	Cosmos III-Pentium III-550MHz 10/100 TX PCI UTP
File Server	File Server 10/100 TX PCI UTP	DTK Computer	Cosmos II-Pentium II-400MHz 10/100 TX PCI UTP
Internet Server	Internet Server 10/100 TX PCI UTP 10/100 TX PCI UTP (2)	DTK Computer	Cosmos II-Pentium II-400MHz 10/100 TX PCI UTP 10/100 TX PCI UTP
Switch 2	Switch 2	3Com Corp.	SuperStack II Baseline 10/100 Switch 12 ports
Switch 3	Switch 3	3Com Corp.	SuperStack II Baseline 10/100 Switch 12 ports
Switch 4	Switch 4	3Com Corp.	SuperStack II Switch 3300 (24 ports)
Switch1	Switch1	3Com Corp.	SuperStack II Baseline 10/100 Switch 12 ports
WS Administrator	WS Administrator 10/100 TX PCI UTP (9)	IBM	PC 300GL (with K7 AMD processors)-627575U 10/100 TX PCI UTP
WS1	WS1 10/100 TX PCI UTP (3)	IBM	PC 300GL (with Pentium III processors)-627575U 10/100 TX PCI UTP
WS10	WS10 10/100 TX PCI UTP (41)	IBM	PC 300GL (with Celeron processors)-627575U 10/100 TX PCI UTP
WS11	WS11 10/100 TX PCI UTP (42)	IBM	PC 300GL (with Celeron processors)-627575U 10/100 TX PCI UTP
WS12	WS12 10/100 TX PCI UTP (16)	IBM	PC 300GL (with Pentium II processors)-627557U 10/100 TX PCI UTP
WS13	WS13 10/100 TX PCI UTP (12)	IBM	PC 300GL (with Pentium II processors)-627557U 10/100 TX PCI UTP
WS14	WS14 10/100 TX PCI UTP (13)	IBM	PC 300GL (with Pentium III processors)-627557U 10/100 TX PCI UTP
WS15	WS15 10/100 TX PCI UTP (14)	IBM	PC 300GL (with Pentium II processors)-627557U 10/100 TX PCI UTP
WS16	WS16 10/100 TX PCI UTP (15)	IBM	PC 300GL (with K7 AMD processors)-627557U 10/100 TX PCI UTP
WS17	WS17 10/100 TX PCI UTP (25)	IBM	PC 300GL (with Pentium III processors)-628786U 10/100 TX PCI UTP
WS18	WS18 10/100 TX PCI UTP (24)	IBM	PC 300GL (with Celeron processors)-628786U 10/100 TX PCI UTP
WS19	WS19	IBM	PC 300GL (with Pentium III processors)-628786U

	10/100 TX PCI UTP (23)		10/100 TX PCI UTP
WS2	WS2	IBM	PC 300GL (with K7 AMD processors)
	10/100 TX PCI UTP (4)		10/100 TX PCI UTP
WS20	WS20	IBM	PC 300GL (with K7 AMD processors)-628786U
	10/100 TX PCI UTP (19)		10/100 TX PCI UTP
WS21	WS21	IBM	PC 300GL (with Celeron processors)-628786U
	10/100 TX PCI UTP (18)		10/100 TX PCI UTP
WS22	WS22	IBM	PC 300GL (with Pentium 4 processors)-628786U
	10/100 TX PCI UTP (22)		10/100 TX PCI UTP
WS23	WS23	IBM	PC 300GL (with K7 AMD processors)-628786U
	10/100 TX PCI UTP (21)		10/100 TX PCI UTP
WS24	WS24	IBM	PC 300GL (with Pentium II processors)-628786U
	10/100 TX PCI UTP (20)		10/100 TX PCI UTP
WS25	WS25	IBM	PC 300GL (with Pentium II processors)-628786U
	10/100 TX PCI UTP (36)		10/100 TX PCI UTP
WS26	WS26	IBM	PC 300GL (with Pentium II processors)-628786U
	10/100 TX PCI UTP (37)		10/100 TX PCI UTP
WS27	WS27	IBM	PC 300GL (with Pentium II processors)-628786U
	10/100 TX PCI UTP (38)		10/100 TX PCI UTP
WS28	WS28	IBM	PC 300GL (with Pentium II processors)-628786U
	10/100 TX PCI UTP (39)		10/100 TX PCI UTP
WS3	WS3	IBM	PC 300GL (with Pentium II processors)-627575U
	10/100 TX PCI UTP (5)		10/100 TX PCI UTP
WS4	WS4	IBM	PC 300GL (with Pentium 4 processors)-627575U
	10/100 TX PCI UTP (6)		10/100 TX PCI UTP
WS5	WS5	IBM	PC 300GL (with K7 AMD processors)-627575U
	10/100 TX PCI UTP (7)		10/100 TX PCI UTP
WS6	WS6	IBM	PC 300GL (with Pentium III processors)-627575U
	10/100 TX PCI UTP (8)		10/100 TX PCI UTP
WS7	WS7	IBM	PC 300GL (with Pentium II processors)-627575U
	10/100 TX PCI UTP		10/100 TX PCI UTP
WS8	WS8	IBM	PC 300GL (with Celeron processors)-627575U
	10/100 TX PCI UTP (2)		10/100 TX PCI UTP
WS9	WS9	IBM	PC 300GL (with Celeron processors)-627575U
	10/100 TX PCI UTP (40)		10/100 TX PCI UTP

Расчет стоимости проекта

Model	Vendor	Part Number	Price	Qty	Total
10/100 TX PCI UTP	Compaq Computer	169845-001	139.00	36	5,004.00
AccessBuilder Internet 400, U	3Com Corp.	3C480000		1	
Cosmos III-Pentium III-550MHz	DTK Computer			1	
Cosmos II-Pentium II-400MHz	DTK Computer	Cosmos II-Pentium II		2	
LinkBuilder Bridge MicroModule	3Com Corp.	3C16060		1	
PC 300GL (with Celeron processors)-627557U	IBM	627557U		1	
PC 300GL (with Celeron processors)-627575U	IBM	627575U		4	
PC 300GL (with Celeron processors)-628786U	IBM	628786U		2	
PC 300GL (with K7 AMD processors)-627557U	IBM	627557U		2	
PC 300GL (with K7 AMD processors)-627575U	IBM	627575U		2	
PC 300GL (with K7 AMD processors)-628786U	IBM	628786U		2	
PC 300GL (with Pentium 4 processors)-627557U	IBM	627557U		3	
PC 300GL (with Pentium II processors)-627557U	IBM	627557U		3	
PC 300GL (with Pentium II processors)-627575U	IBM	627575U		2	
PC 300GL (with Pentium II processors)-628786U	IBM	628786U		5	
PC 300GL (with Pentium III processors)-627557U	IBM	627557U		2	
PC 300GL (with Pentium III processors)-627575U	IBM	627575U		2	
PC 300GL (with Pentium III processors)-628786U	IBM	628786U		2	
SuperStack II Baseline 10/100 Switch 12 ports	3Com Corp.	3C16464A		3	
SuperStack II Switch 3300 (24 ports)	3Com Corp.	3C16980		1	

Задание на практическую работу

Разработать и промоделировать в среде NetCracker проект компьютерной сети, состоящей из нескольких сегментов (рабочих групп), имеющих определенную специализацию и объединенных посредством высокоскоростной магистрали, обеспечивая также подключение удаленных подразделений и установку необходимых серверов.

В соответствии с номером варианта производится выборка исходных данных по проектируемой сети из таблиц, приведенных ниже. Для выполнения индивидуального задания необходимо:

1) произвести анализ исходных данных и зоны проектирования;

2) произвести обоснованный выбор и размещение активного и пассивного сетевого оборудования в среде NetCracker;

3) произвести построение сети и соединение выбранного оборудования, учитывая рекомендуемые в задании параметры сети;

4) провести моделирование полученной сети и анализ результатов ее работы; 5) составить отчет о проделанной работе.

ВНИМАНИЕ. Данные по специализации рабочих групп, типам серверов и основным протоколам выбираются из таблиц ниже. (Специализация рабочих групп, Типы серверов и Перечень протоколов) в соответствии с индексами, приведенными в табл. 14.3-14.5.

Таблица 14.1. «Варианты заданий»

№ варианта	Количество рабочих групп	Специализация рабочих групп	Количество рабочих станций в рабочей группе
1	3	1, 4, 15	5-4-5
2	4	16, 7, 9, 8	10-2-4-3
3	3	4, 5, 15	7-6-3
4	5	6, 11, 12, 13, 14	3-5-3-6-5

5	3	6, 7, 8	5-2-8
6	4	1, 2, 3, 15	10-10-5-5
7	2	7, 10	4-5
8	4	1, 2, 3, 15	8-12-3-7
9	3	1, 4, 15	3-6-2
10	5	4, 5, 7, 10, 1	3-5-5-3-4
11	2	9, 7	3-3
12	4	1, 4, 5, 15	3-4-5-2

Таблица 14.2

№ варианта	Предпочтительная магистральная технология	Тип серверов	Основной протокол	Примечание
1	По выбору	1-1,1-3	1	
2	ISDN, FDDI	4-1,4-5	по выбору	Удаленное
3	FastEthernet	6-1,6-4,6-6	1	
4	Gigabit Ethernet	1-1	1	
5	Ethernet	1-1	2	
6	Gigabit Ethernet	5-1,5-5,5-6	1,2,>>	
7	По выбору	1-1,1-5	1	Удаленное
8	ATM	5-1,5-5,5-6	1,2	Удаленное
9	Frame Relay	1-1,1-3	2	Удаленное
10	По выбору	1-все	по выбору	Удаленное
11	Token Ring	3-1	2	Удаленное
12	ATM	5-1,5-3,5-4,5-6	1	

Таблица 14.3. «Специализация рабочих групп»

Индекс специализации	Наименование специализации
1	Разработка прикладного программного обеспечения
2	Разработка системного программного обеспечения
3	Разработка драйверов
4	Разработка баз данных
5	Разработка WEB приложений
6	Издательское дело
7	Бухгалтерия
8	Рекламное дело
9	Склад
10	Отдел кадров
11	Видео продукция
12	Аудио продукция
13	3D разработка
14	Анимационный отдел
15	Группа отладки программного обеспечения
16	Торговля

Таблица 14.4. «Типы серверов»

Индекс сервера	Тип сервера	Индекс назначения	Назначение сервера
1	Server	1	File server
2	Ethernet Server	2	E-mail server
3	Token Ring Server	3	SQL server
4	FDDI Server	4	FTP server
5	ATM Server	5	Small office database server
6	FastEthernet	6	HTTP server

Таблица 14.5. «Перечень протоколов»

Индекс протокола	Наименование протокола
1	TCP/IP
2	IPX/SPX
3	XNS
4	SNA

Отчет по индивидуальному заданию должен содержать следующие разделы:

1. Техническое задание.
2. Графическое представление полученной сети в режиме моделирования с отображением статистики (для магистрали сети и всех рабочих групп) с анализом полученных результатов.
3. Перечень используемого оборудования (Device Summary).
4. Полный перечень оборудования с расчетом стоимости проекта (Bill of Materials).
5. Выводы

Контрольные вопросы

1. Технология Ethernet.
2. Технология Fast Ethernet.
3. Технология Gigabit Ethernet.
4. Что в себя включает выбор активного сетевого оборудования?
5. Какие характеристики вы учитывали при выборе активного сетевого оборудования?

Список использованных источников

Основные источники:

1. Кузьменко, Н.Г. Компьютерные сети и сетевые технологии / Н.Г. Кузьменко. - СПб.: Наука и техника, 2013. - 368 с
2. Олифер, В.Г. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. Стандарт третьего поколения / В.Г. Олифер, Н.А. Олифер.. - СПб.: Питер, 2013. - 944 с.
3. Таненбаум, Э. Компьютерные сети / Э. Таненбаум. - СПб.: Питер, 2013. - 960 с.
4. Ю.Ю. Громов, В.Е, Дидрих, И.В, Дидрих, Ю.Ф.Мартемьянов, В.О. Драчев, Серегин М.Ю. Компьютерные телекоммуникации - Издательство ФГБОУ ВПО «ТГТУ», 2012. – 224 с. (http://biblioclub.ru/)
5. Смирнова Е. В. , Баскаков И. В. , Пролетарский А. В. , Федотов Р. А. Построение коммутируемых компьютерных сетей - М.: Национальный Открытый Университет «ИНТУИТ», 2016. – 429 с. (http://biblioclub.ru)

Дополнительные источники:

1. Фролов, А.В.; Фролов, Г.В. Локальные сети персональных компьютеров. Использование протоколов IPX, SPX, NETBIOS; Диалог-Мифи - Москва, 2013. - 160 с.
2. Гузик, В.Ф.; Поленов М.Ю.; Ляпунцова Е.В.; Мунтян О.А.; Гушанский С.М.; Кондратенко С.В. Проектирование и моделирование компьютерных сетей; Издательство Таганрогского государственного радиотехнического университета, 2003. – 94 с.
3. NetCracker User’s Guide and Reference Manual. NetCracker Technology (Division of Advanced Visual Data, Inc.), 2012

Интернет-источники:

1. Руководство по Iptables http://www.opennet.ru/docs/RUS/iptables/

Учебное издание

Андрей Владимирович Семенов

Компьютерные сети

Учебное пособие для студентов специальности
09.02.04 – Информационные системы (по отраслям)

Технический редактор: Иванова Н.И.
Компьютерная верстка: Семенов А.В.
Подписано к печати _____Бумага для множительной техники
Формат _____Усл.печ.лист._____Тираж_____экз. Заказ_____

Отпечатано с авторского оригинала в отделе оперативной печати
Старооскольского технологического института.
Старый Оскол, микрорайон Макаренко, 40.