Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Оренбургский государственный университет»

Е.И. Ряполова, Ю.И. Синицын

ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ

Рекомендовано к изданию Редакционно-издательским советом федерального государственного бюджетного образовательного учреждения высшего профессионального образования «Оренбургский государственный университет» в качестве методических указаний для студентов, обучающихся по программам высшего профессионального образования по направлениям подготовки 090900.62 Информационная безопасность, 230100.62 Информатика и вычислительная техника

Оренбург
2012

УДК 004.77
ББК 32.973.202 я 7
Р 91

Рецензент – кандидат педагогических наук, доцент Л.Ф. Насейкина

Ряполова, Е.И.

Р 91 Вычислительные сети: методические указания к лабораторным работам/ Е.И. Ряполова, Ю.И. Синицын; Оренбургский гос. ун-т. – Оренбург: ОГУ, 2012. – 169 с.

Методические указания предназначен для проведения лабораторных работ студентами, изучающими дисциплины «Вычислительные сети» и «Сети ЭВМ и телекоммуникации». Приводятся варианты заданий для лабораторных работ и содержание этапов их выполнения.

Методические указания предназначены для студентов направлений подготовки 090900.62 Информационная безопасность, 230100.62 Информатика и вычислительная техника.

УДК 004.77
ББК 32.973.202 я 7

Содержание

1 Лабораторная работа № 1. Стек TCP/IP

Цель работы. Получить основные теоретические сведения по стеку TCP/IP.

Теоретическая справка.

В настоящее время в сетях используется несколько стеков коммуникационных протоколов. Наиболее популярны следующие стеки:

- TCP/IP;
- IPX/SPX;
- NetBIOS/SMB;
- DECnet;
- SNA;
- OSI.

Все эти стеки, кроме SNA на нижних уровнях — физическом и канальномиспользуют одни и те же хорошо стандартизованные протоколы Ethernet, Token Ring, FDDI и ряд других, которые позволяют задействовать во всех сетях одну и ту же аппаратуру. Зато на верхних уровнях все стеки работают по своим протоколам. Эти протоколы часто не соответствуют рекомендуемому моделью OSI разбиению на уровни. В частности, функции сеансового и представительного уровня, как правило, объединены с прикладным уровнем. Такое несоответствие связано с тем, что модель OSI появилась как результат обобщения уже существующих и реально используемых стеков [1].

Стек TCP/IP был разработан для связи экспериментальной сети ARPAnet с другими сетями как набор общих протоколов для разнородной вычислительной среды. Стек TCP/IP на нижнем уровне поддерживает все популярные стандарты физического и канального уровней для локальных сетей — это Ethernet, Token Ring, FDDI, для глобальных — протоколы работы на аналоговых коммутируемых и выделенных линиях (SLIP, PPP) протоколы территориальных сетей X.25 и ISDN.

Основными протоколами стека, давшими ему название, являются протоколы IP и TCP. Эти протоколы в терминологии модели OSI относятся к сетевому и транспортному уровням, соответственно. IP обеспечивает продвижение пакета по составной сети, а TCP гарантирует надежность его доставки. Стек TCP/IP вобрал в себя большое количество протоколов прикладного уровня. К ним относятся такие протоколы, как протокол пересылки файлов FTP, протокол эмуляции терминала telnet, почтовый протокол SMTP, используемый в электронной почте сети Internet, гипертекстовые сервисы службы WWW и другие.

Уровни.

Сетевые протоколы обычно разрабатываются по уровням, причем каждый уровень отвечает за собственную фазу коммуникаций. Семейства протоколов, такие как TCP/IP, это комбинации различных протоколов на различных уровнях. TCP/IP состоит из четырех уровней, как показано в таблице 1 [1].

Таблица 1 – Уровни протокола TCP/IP

Прикладной	Telnet, FTP, e-mail и т.д.
Транспортный	TCP,UDP
Сетевой	IP, ICMP, IGMP
Канальный	драйвер устройства и интерфейсная плата

Каждый уровень несет собственную функциональную нагрузку.

1. Канальный уровень (link layer). Его называют уровнем сетевого интефейса. Обычно включает в себя драйвер устройства в операционной системе и соответствующую сетевую интерфейсную плату в компьютере. Вместе они обеспечивают аппаратную поддержку физического соединения с сетью (с кабелем или с другой средой передачи).

2. Сетевой уровень (network layer), иногда называемый уровнем межсетевого взаимодействия, отвечает за передачу пакетов по сети. Маршрутизация пакетов осуществляется на этом уровне. IP (Internet Protocol - протокол Internet), ICMP (Internet Control Message Protocol - протокол управления сообщениями Internet) и

IGMP (Internet Group Management Protocol - протокол управления группами Internet) обеспечивают сетевой уровень в семействе протоколов TCP/IP.

3. Транспортный уровень (transport layer) отвечает за передачу потока данных между двумя компьютерами и обеспечивает работу прикладного уровня, который находится выше. В семействе протоколов TCP/IP существует два транспортных протокола - TCP (Transmission Control Protocol) и UDP (User Datagram Protocol). TCP осуществляет передачу данных между двумя компьютерами. Он обеспечивает деление данных, передающихся от одного приложения к другому, на пакеты подходящего для сетевого уровня размера, подтверждение принятых пакетов, установку тайм-аутов, в течение которых должно прийти подтверждение на пакет, и так далее. Так как надежность передачи данных гарантируется на транспортном уровне, на прикладном уровне эти детали игнорируются. UDP предоставляет более простой сервис для прикладного уровня. Он просто отсылает пакеты, которые называются датаграммами (datagram) от одного компьютера к другому. За надежность передачи данных, при использовании датаграмм отвечает прикладной уровень.

4. Прикладной уровень (application layer) определяет детали каждого конкретного приложения. Существует несколько приложений TCP/IP, которые присутствуют практически в каждой реализации:

– Telnet - удаленный терминал;

– FTP, File Transfer Protocol - протокол передачи файлов;

– SMTP, Simple Mail Transfer Protocol - простой протокол передачи электронной почты;

– SNMP, Simple Network Management Protocol - простой протокол управления сетью [1].

Полезным свойством протокола TCP/IP является его способность фрагментировать пакеты. Сложная составная сеть часто состоит из сетей, построенных на совершенно разных принципах. В каждой из этих сетей может быть установлена собственная величина максимальной длины единицы передаваемых данных (кадра). В таком случае при переходе из одной сети, имеющей большую

максимальную длину, в другую, с меньшей максимальной длиной, может возникнуть необходимость разделения передаваемого кадра на несколько частей. Протокол IP стека TCP/IP решает эту задачу.

Другой особенностью технологии TCP/IP является гибкая система адресации, позволяющая более просто по сравнению с другими протоколами аналогичного назначения включать в интерсеть (объединенную или составную сеть) сети других технологий. Это свойство также способствует применению стека TCP/IP для построения больших гетерогенных сетей.

Недостаток использования этого протокола - требования к ресурсам и сложность администрирования IP - сетей. Для реализации функциональных возможностей протоколов стека TCP/IP требуются большие вычислительные затраты. Гибкая система адресации и отказ от широковещательных рассылок приводят к наличию в IP-сети различных централизованных служб типа DNS, DHCP и т. п. Каждая из этих служб упрощает администрирование сети и конфигурирование оборудования, но в то же время сама требует внимания со стороны администраторов.

В стеке TCP/IP используются три типа адресов - локальные (называемые также аппаратными), IP - адреса и символьные доменные имена.

В терминологии TCP/IP под локальным адресом понимается такой тип адреса, который используется средствами базовой технологии для доставки данных в пределах подсети, являющейся элементом составной интерсети. В разных подсетях допустимы разные сетевые технологии, разные стеки протоколов, поэтому при создании стека TCP/IP предполагалось наличие разных типов локальных адресов. Если подсетью интерсети является локальная сеть, то локальный адрес — это MAC - адрес. MAC - адрес назначается сетевым адаптерам и сетевым интерфейсам маршрутизаторов. MAC - адрес назначаются производителями оборудования и являются уникальными, так как управляются централизованно. Для всех существующих технологий локальных сетей MAC - адрес имеет формат 6 байт, например 11-A0-17-3D-BC-01. Однако протокол IP может работать и над протоколами более высокого уровня, например над протоколом IPX или X.25. В

этом случае локальными адресами для протокола IP соответственно будут адреса IPX и X.25. Следует учесть, что компьютер в локальной сети может иметь несколько локальных адресов даже при одном сетевом адаптере. Некоторые сетевые устройства не имеют локальных адресов (глобальные порты маршрутизаторов, предназначенные для соединений типа «точка-точка»).

Символьные доменные имена. Символьные имена в IP - сетях называются доменными и строятся по иерархическому признаку.

Составляющие полного символьного имени в IP - сетях разделяются точкой и перечисляются в следующем порядке - сначала простое имя конечного узла, затем имя группы узлов (например, имя организации), затем имя более крупной группы (поддомена) и так до имени домена самого высокого уровня (например, домена объединяющего организации по географическому принципу (RU — Россия, UK — Великобритания, SU — США). В сетях TCP/IP используется специальная распределенная служба Domain Name System (DNS), которая устанавливает это соответствие на основании создаваемых администраторами сети таблиц соответствия. Поэтому доменные имена называют также DNS – именами [1].

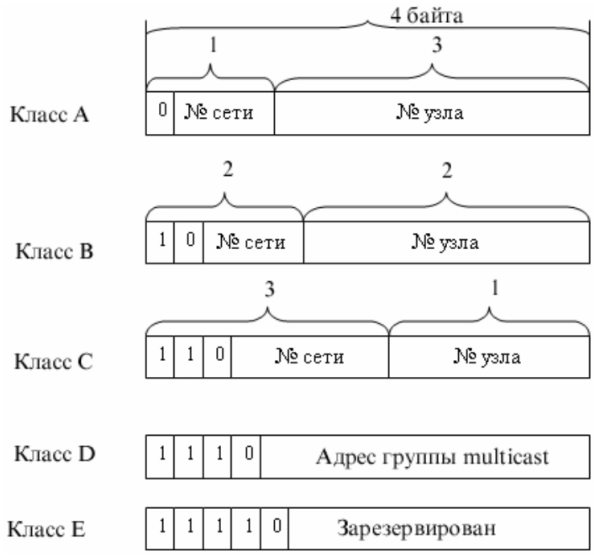


Рисунок 1 – Маски классов сетей

IP - адрес имеет длину 4 байта и обычно записывается в виде четырех чисел, представляющих значения каждого байта в десятичной форме и разделенных точками. Адрес состоит из двух логических частей — номера сети и номера узла в

сети. Какая часть адреса относится к номеру сети, а какая — к номеру узла, определяется значениями первых бит адреса. Значения этих бит являются также признаками того, к какому классу относится тот или иной IP - адрес.

Для установки границы между номером сети и номером узла используются маски. Маска — это число, которое используется в паре с IP - адресом (двоичная запись маски содержит единицы в тех разрядах, которые должны в IP - адресе интерпретироваться как номер сети). Поскольку номер сети является цельной частью адреса, единицы в маске также должны представлять непрерывную последовательность. Для стандартных классов сетей маски имеют следующие значения:

– класс A - 11111111.00000000.00000000. 00000000 (255.0.0.0);

– класс B - 11111111.11111111.00000000. 00000000 (255.255.0.0);

– класс C - 11111111.11111111.11111111.00000000 (255.255.255.0).

В масках количество единиц в последовательности, определяющей границу номера сети, не обязательно должно быть кратным 8, чтобы повторять деление адреса на байты.

Пример. Для IP-адреса 129.64.134.5 указана маска 255.255.128.0, то есть в двоичном виде:

– IP - адрес 129.64.134.5 - 10000001. 01000000.10000110. 00000101;

– маска 255.255.128.0 - 11111111.11111111.10000000.00000000.

Если игнорировать маску, то в соответствии с системой классов адрес 129.64.134.5 относится к классу B, а значит, номером сети являются первые 2 байта — 129.64.0.0, а номером узла — 0.0.134.5.

Если же использовать для определения границы номера сети маску, то 17 последовательных единиц в маске, «наложенные» на IP-адрес, определяют в качестве номера сети в двоичном выражении число: 10000001.01000000.10000000. 00000000 или в десятичной форме записи — номер сети 129.64.128.0, а номер узла 0.0.6.5. IP - пакет состоит из заголовка и поля данных. Заголовок, как правило, имеющий длину 20 байт и имеет структуру, показанную на рисунке 2.

Поле «Номер версии» (Version), занимающее 4 бит, указывает версию протокола IP. Сейчас используется версия 4 (IPv4) (новая версия 6 (IPv6)).

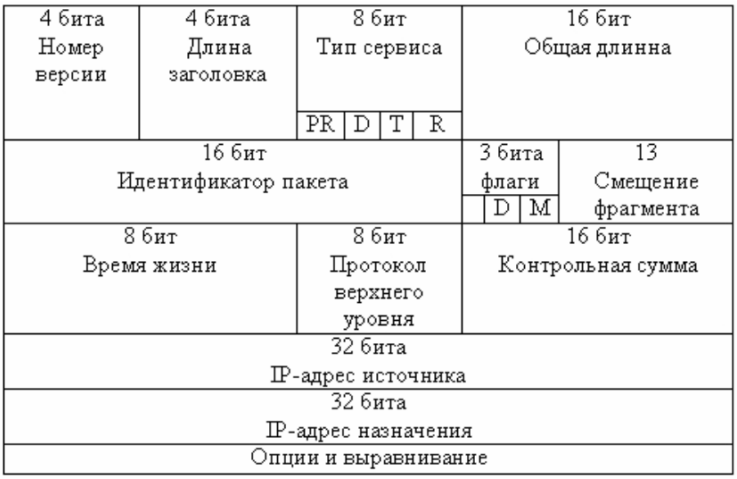


Рисунок 2 – Структура заголовка

Поле «Длина заголовка» (IHL) IP - пакета занимает 4 бит и указывает значение длины заголовка, измеренное в 32-битовых словах. Обычно заголовок имеет длину в 20 байт (пять 32-битовых слов), но при увеличении объема служебной информации эта длина может быть увеличена за счет использования дополнительных байт в поле Опции (IP Options) [1].

Поле «Тип сервиса» (Type of Service) занимает один байт и задает приоритетность пакета и вид критерия выбора маршрута. Первые три бита этого поля образуют подполе приоритета пакета (Precedence). Приоритет может иметь значения от самого низкого - 0 (нормальный пакет) до самого высокого - 7 (пакет управляющей информации). Маршрутизаторы и компьютеры могут принимать во внимание приоритет пакета и обрабатывать более важные пакеты в первую очередь. Поле Тип сервиса содержит также три бита, определяющие критерий выбора маршрута. Реально выбор осуществляется между тремя альтернативами - малой задержкой, высокой достоверностью и высокой пропускной способностью. Установленный бит D (delay) говорит о том, что маршрут должен выбираться для минимизации задержки доставки данного пакета, бит T — для максимизации пропускной способности, а бит R — для максимизации надежности доставки. Во многих сетях улучшение одного из этих параметров связано с ухудшением другого,

кроме того, обработка каждого из них требует дополнительных вычислительных затрат. Зарезервированные биты имеют нулевое значение.

Поле «Общая длина» (Total Length) занимает 2 байта и означает общую длину пакета с учетом заголовка и поля данных. Максимальная длина пакета ограничена разрядностью поля, определяющего эту величину, и составляет 65535 байт, однако в большинстве хост-компьютеров и сетей столь большие пакеты не используются. При передаче по сетям различного типа длина пакета выбирается с учетом максимальной длины пакета протокола нижнего уровня, несущего IP - пакеты. Если это кадры Ethernet, то выбираются пакеты с максимальной длиной в 1500 байт, умещающиеся в поле данных кадра Ethernet. В стандарте предусматривается, что все хосты должны быть готовы принимать пакеты вплоть до 576 байт длиной (приходят ли они целиком или по фрагментам). Хостам рекомендуется отправлять пакеты размером более чем 576 байт, только если они уверены, что принимающий хост или промежуточная сеть готовы обслуживать пакеты такого размера.

Поле «Идентификатор пакета» (Identification) занимает 2 байта и используется для распознавания пакетов, образовавшихся путем фрагментации исходного пакета. Все фрагменты должны иметь одинаковое значение этого поля.

Поле «Флаги» (Flags) занимает 3 бита и содержит признаки, связанные с фрагментацией. Установленный бит DF (Do not Fragment) запрещает маршрутизатору фрагментировать данный пакет, а установленный бит MF (More Fragments) говорит о том, что данный пакет является промежуточным (не последний) фрагментом. Оставшийся бит зарезервирован.

Поле «Смещение фрагмента» (Fragment Offset) занимает 13 бит и задает смещение в байтах поля данных этого пакета от начала общего поля данных исходного пакета, подвергнутого фрагментации. Используется при сборке/разборке фрагментов пакетов при передачах их между сетями с различными величинами MTU. Смещение должно быть кратно 8 байт.

Поле «Время жизни» (Time to Live) занимает один байт и означает предельный срок, в течение которого пакет может перемещаться по сети. Время

жизни данного пакета измеряется в секундах и задается источником передачи. На маршрутизаторах и в других узлах сети по истечении каждой секунды из текущего времени жизни вычитается единица. Единица вычитается и в том случае, когда время задержки меньше секунды. Время жизни можно считать равным максимальному числу узлов, которые разрешено пройти данному пакету до того, как он достигнет места назначения. Если параметр времени жизни станет нулевым до того, как пакет достигнет получателя, этот пакет будет уничтожен.

Идентификатор протокола верхнего уровня (Protocol) занимает один байт и указывает, какому протоколу верхнего уровня принадлежит информация, размещенная в поле данных пакета (например, это могут быть сегменты протокола TCP, дейтаграммы UDP, пакеты ICMP или OSPF). Значения идентификаторов для различных протоколов приводятся в документе RFC «Assigned Numbers».

Контрольная сумма (Header Checksum) занимает 2 байта и рассчитывается только по заголовку. Поскольку некоторые поля заголовка меняют свое значение в процессе передачи пакета по сети (например, время жизни), контрольная сумма проверяется и повторно рассчитывается при каждой обработке IP - заголовка. Контрольная сумма - 16 бит подсчитывается как дополнение к сумме всех 16-битовых слов заголовка. При вычислении контрольной суммы значение самого поля «контрольная сумма» устанавливается в нуль. Если контрольная сумма неверна, то пакет будет отброшен, как только ошибка будет обнаружена.

Поля «IP - адрес источника» (Source IP Address) и «IP - адрес назначения» (Destination IP Address) имеют одинаковую длину - 32 бита и одинаковую структуру.

Поле «Опции» (IP Options) является необязательным и используется обычно только при отладке сети. Механизм опций предоставляет функции управления, которые необходимы или просто полезны при определенных ситуациях, однако он не нужен при обычных коммуникациях. Это поле состоит из нескольких подполей, каждое из которых может быть одного из восьми предопределенных типов. В этих подполях можно указывать точный маршрут прохождения маршрутизаторов, регистрировать проходимые пакетом маршрутизаторы, помещать данные системы безопасности, а также временные отметки. Так как число подполей может быть

произвольным, то в конце поля Опции должно быть добавлено несколько байт для выравнивания заголовка пакета по 32-битной границе.

Поле «Выравнивание» (Padding) используется для того, чтобы убедиться в том, что IP-заголовок заканчивается на 32-битной границе. Выравнивание осуществляется нулями.

Каждый интерфейс в объединенной сети должен иметь уникальный IP адрес. Эти адреса представляют из себя тридцатидвухбитовые числа. Существует определенная структура адреса Internet. Эти 32-битные адреса обычно записываются как 4 десятичных числа, по одному на каждый байт адреса. Такая форма записи называется "десятичной записью с точками" (dotted-decimal) [1].

Пример. Адрес сети класса B может быть записан как 140.252.13.33.

Определить класс адреса, или класс сети, можно по первому числу в адресе (таблица 2).

Таблица 2 - Пять классов адресов

Класс	Диапазон IP адресов в разных классах сетей
A	0.0.0.0 - 127.255.255.255
B	128.0.0.0 - 191.255.255.255
C	192.0.0.0 - 223.255.255.255
D	224.0.0.0 - 239.255.255.255
E	240.0.0.0 - 247.255.255.255

Так как каждый интерфейс, подключенный к сети, должен иметь уникальный адрес, встает вопрос распределения IP адресов в глобальной сети Internet. Этим занимается сетевой информационный центр (Internet Network Information Center или InterNIC). InterNIC назначает только сетевые идентификаторы (ID). Назначением идентификаторов хостов в сети занимаются системные администраторы.

Существует три типа IP адресов - персональный адрес (unicast) - указывает на один хост, широковещательный адрес (broadcast) - указывает на все хосты в указанной сети, и групповой адрес (multicast) - указывает на группу хостов, принадлежащей к группе адресации.

Порядок выполнения работы.

1. Изучить назначение протокола TCP/IP.
2. Изучить уровни протокола TCP/IP.
3. Изучить систему адресации протокола TCP/IP.
4. Изучить структуру заголовка пакета IP.
5. Изучить описание полей заголовка IP.
6. Оформить отчет.

Содержание отчета по лабораторной работе.

1. Название и цель работы.
2. Основные теоретические пункты общих сведений.
3. Выводы по выполненной работе.
4. Список использованных источников.

Контрольные вопросы.

1. Назначение протокола TCP/IP.
2. Какие стандарты поддерживает протокол TCP/IP ?
3. Какие уровни представлены в протоколе TCP/IP ?
4. Какую функциональную нагрузку несет канальный уровень ?
5. Какую функциональную нагрузку несет сетевой уровень ?
6. Какую функциональную нагрузку несет транспортный уровень ?
7. Какую функциональную нагрузку несет прикладной уровень ?
8. Как устроена система адресации в протоколе TCP/IP ?
9. Что понимается под локальным адресом в протоколе TCP/IP ?
10. Какое назначение IP - адреса ?
11. Что такое символьные доменные имена ?
12. Какую длину и структуру имеет IP - адрес ?
13. Что представляют собой маски классов сетей ?
14. Из чего состоит IP - пакет ?
15. Какие поля (и их назначение) используются в пакете ?

2 Лабораторная работа № 2. Работа с кабелем типа «витая пара»

Цель работы. Получение навыков работы с кабелем типа «витая пара».

Теоретическая справка.

Витая пара (UTP/STP, unshielded/shielded twisted pair) в настоящее время является распространенной средой передачи сигналов в локальных сетях. Кабели UTP/STP используются в сетях Ethernet, Token Ring и ARCnet. Они различаются по категориям (в зависимости от полосы пропускания) и типу проводников (гибкие или одножильные). В кабеле 5-й категории, как правило, находится восемь проводников, перевитых попарно.

Структурированная кабельная система, построенная на основе витой пары 5-й категории, имеет большую гибкость в использовании.

На каждое рабочее место устанавливается не менее двух (рекомендуется три) четырехпарных розеток RJ-45. Каждая из них отдельным кабелем 5-й категории соединяется с кроссом или патч-панелью, установленной в специальном помещении — серверной. В это помещение заводятся кабели со всех рабочих мест, а также городские телефонные вводы, выделенные линии для подключения к глобальным сетям и т.п.

Патч-панель (панель соединений) представляет собой группу розеток RJ-45, смонтированных на пластине шириной 19 дюймов. Это стандартный размер для универсальных коммуникационных шкафов — рэков (rack), в которых устанавливается оборудование (концентраторы, коммутаторы, маршрутизаторы, серверы, ИБП и т.п.) [7].

Кросс в отличие от патч-панели розеток не имеет. Вместо них он несет на себе специальные соединительные модули. В данном случае его преимущество перед патч-панелью в том, что при его использовании в телефонии вводы можно соединять между собой не специальными патч-кордами, а обычными проводами.

Кроме того, кросс мож но монтировать прямо на стену — наличия коммуникационного шкафа он не требует.

Кабели с многожильными гибкими проводниками используются в качестве патч-кордов, то есть соединительных кабелей между розеткой и сетевой платой, либо между розетками на панели соединений или кроссе. Кабели с одножильными проводниками — для прокладки собственно кабельной системы. Монтаж разъемов и розеток на эти кабели совершенно идентичен, но обычно кабели с одножильными проводниками монтируются на розетки рабочих мест пользователей, панели соединений и кроссы, а разъемы устанавливают на гибкие соединительные кабели.

Как правило, применяются следующие виды разъемов [7]:

– S110 — общее название разъемов для подключения кабеля к универсальному кроссу «110» или коммутации между вводами на кроссе;

– RJ-11 и RJ-12 — разъемы с шестью контактами (первый обычно применяются в телефонии общего назначения, второй обычно используется в телефонных аппаратах, предназначенных для работы с офисными мини-АТС, а также для подключения кабеля к сетевым платам ARCnet);

– RJ-45 — восьмиконтактный разъем, использующийся обычно для подключения кабеля к сетевым платам Ethernet либо для коммутации на панели соединений.

Разъем RJ-45.

В зависимости от того, что с чем нужно коммутировать, применяются различные патч-корды – «45-45» (с каждой стороны по разъему RJ-45), «110-45» (с одной стороны S110, с другой — RJ-45) или «110-110».

Для монтажа разъемов RJ-11, RJ-12 и RJ-45 используются специальные обжимочные приспособления, различающиеся между собой количеством ножей (6 или 8) и размерами гнезда для фиксации разъема. В качестве примера рассмотрим монтаж кабеля 5-й категории на разъем RJ-45 [7].

Порядок выполнения работы.

1. Обрежьте конец кабеля. Торец кабеля должен быть ровным.

2. Используя специальный инструмент, снимите с кабеля внешнюю изоляцию на длину примерно 30 мм и обрежьте нить, вмонтированную в кабель (нить предназначена для удобства снятия изоляции с кабеля на большую длину). Любые повреждения (надрезы) изоляции проводников абсолютно недопустимы — именно поэтому желательно использовать специальный инструмент, лезвие резака которого выступает ровно на толщину внешней изоляции.

3. Аккуратно разведите, расплетите и выровняйте проводники. Выровняйте их в один ряд, при этом соблюдая цветовую маркировку. Существует два наиболее распространенных стандарта по разводке цветов по парам: T568A (рекомендуемый компанией Siemon) и T568B (рекомендуемый компанией AT&T и фактически наиболее часто применяемый). На разъеме RJ-45 цвета проводников располагаются так показано в таблице 3.

Таблица 3 – Цвета проводников кабеля типа «витая пара» 5-й категории

Номер контакта	Цвет по T568B	Цвет по T568A
1	бело-оранжевый	бело-зеленый
2	Оранжевый	зеленый
3	бело-зеленый	бело-оранжевый
4	Синий	синий
5	бело-синий	бело-синий
6	Зеленый	оранжевый
7	бело-коричневый	бело-коричневый
8	Коричневый	коричневый

Проводники должны располагаться строго в один ряд без нахлестов друг на друга. Удерживая их одной рукой, другой ровно обрежьте проводники так, чтобы они выступали над внешней обмоткой на 8 - 10 мм.

4. Держа разъем защелкой вниз, вставьте в него кабель. Каждый проводник должен попасть на свое место в разъеме и упереться в ограничитель. Прежде чем обжимать разъем, убедитесь, что вы не ошиблись в разводке проводников. При неправильной разводке помимо отсутствия соответствия номерам контактов на концах кабеля, легко выявляемого с помощью простейшего тестера, возможна более неприятная вещь — появление “разбитых пар” (splitted pairs). Для выявления этого

брака обычного тестера недостаточно, так как электрический контакт между соответствующими контактами на концах кабеля обеспечивается и с виду все как будто бы нормально. Но такой кабель никогда не сможет обеспечить нормальное качество соединения даже в 10-мегабитной сети на расстояние более 40 - 50 метров.

5. Вставьте разъем в гнездо на обжимочном приспособлении и обожмите его до упора-ограничителя на приспособлении. В результате фиксатор на разъеме встанет на свое место, удерживая кабель в разъеме неподвижным. Контактные ножи разъема врежутся каждый в свой проводник, обеспечивая надежный контакт.

Аналогичным образом можно осуществить монтаж разъемов RJ-11 и RJ-12, используя соответствующий инструмент.

Для монтажа разъема S110 специального обжимочного инструмента не требуется. Сам разъем поставляется в разобранном виде. Кстати, в отличие от “одноразовых” разъемов типа RJ разъем S110 допускает многократную разборку и сборку, что очень удобно. Последовательность действий при монтаже.

1. Снимите внешнюю изоляцию кабеля на длину примерно 40 мм, разведите в стороны пары проводников, не расплетая их.

2. Закрепите кабель (в той половинке разъема, на которой нет контактной группы) с помощью пластмассовой стяжки и отрежьте получившийся “хвост”.

3. Аккуратно уложите каждый проводник в органайзер на разъеме. Не расплетайте пару на большую, чем требуется, длину — это ухудшит характеристики всего кабельного соединения. Последовательность укладки пар обычная — синяя-оранжевая - зеленая-коричневая; при этом светлый провод каждой пары укладывается первым.

4. Острым инструментом (бокорезами или ножом) обрежьте каждый проводник по краю разъема.

5. Установите на место вторую половинку разъема и руками обожмите ее до защелкивания всех фиксаторов. При этом ножи контактной группы врежутся в проводники, обеспечивая контакт.

Используя тестер – выполнить «прозвонку» контактов разъемов.

Содержание отчета по лабораторной работе.

1. Название и цель работы.
2. Основные теоретические пункты общий сведений;
3. Этапы проведения работы.
4. Макет выполненной работы (кабель с двумя установленными разъемами RJ-45 на концах).
5. Выводы по выполненной работе.
6. Список использованных источников.

Контрольные вопросы.

1. Какие типы кабелей существуют (типы, характеристики, параметры).
2. Алгоритм последовательность обжима кабеля.
3. Назначение и конструкции разъемов и розеток (обычных и экранированных).

3 Лабораторная работа № 3. Определение строительных длин кабелей связи и определение места обрыва жил

Цель работы. Исследовать метод определения расстояния до места повреждения кабеля при помощи электронного вольтметра в стационарных условиях. Определить расстояние до места повреждения кабеля при помощи электронного вольтметра в стационарных условиях. Дать сравнительный анализ точности измерений приборов ПКП-3 (прибор кабельный переносной), измерителя неоднородностей линий Р5 – 10 и электронного вольтметра В7 – 16А.

Оборудование. Электронный вольтметр В7 – 16А, учебные стенды – 3 комплекта.

Теоретическая справка.

Вольтметр универсальный В7 – 16А предназначен для измерения напряжений постоянного и переменного токов, активного сопротивления при регламентных, ремонтных и регулировочных работах в различных областях электроники, а также

для проверки приборов более низкого класса.

Технические данные.

Время измерения активного сопротивления не превышает 10 с.

В вольтметре предусмотрен индикатор перегрузки.

Предел допускаемой основной погрешности при измерении активного сопротивления при времени преобразования 20 мс и 100 мс:

$$\delta = \pm (0{,}15 + 0{,}05\ R_к / R_x)\ \%, \qquad (1)$$

где Rк – конечное значение установленного предела измерений;

Rx – показание вольтметра.

Предел допускаемой дополнительной погрешности от изменения температуры окружающей среды при всех видах измерений не превышает половины основной погрешности на каждые 10^0 С изменения температуры.

Выходной кодовый сигнал, соответствующий логическому нулю, имеет уровень напряжения не более ± 0,3 В. Выходной кодовый сигнал, соответствующий логической единице, имеет уровень напряжения не менее +2,4 В, на нагрузке не менее 10 кОм.

Задание к проведению лабораторной работы.

1. Ознакомиться с лабораторными установками и прибором электронный вольтметр В7 – 16А.

2. Провести измерение параметров кабельных цепей (Сопротивление шлейфа и расстояния до места обрыва жил).

3. Рассчитать длину кабельных цепей согласно указанию преподавателя.

4. Сравнить измеренные величины с нормами.

5.Составить отчёт по работе.

Подготовка оборудования к работе.

Установить тумблер **«Сеть»** в верхнее положение.

При включении вольтметра должно индицироваться:

– один из знаков "+ ", " –", " ~";

– один из символов "**mV**", "**V**", " **Ω**", "**кΩ**", "**П**", "**МΩ**";

– четырёх - или трёхразрядное число индикаторного табло.

Установить тумблер « » в положение « ».

Установить потенциометр **«ВР. ИНД.»** В положение, обеспечивающее удобное время индикации.

Установить переключатель **«ВР. ПРЕОБРАЗ.»** в положение **«20 mS»** или **«100 mS»**. При этом должны гореть 4 цифровые лампы индикаторного табло.

При установке переключателя **«ВР. ПРЕОБРАЗ.»** в положение **«1 mS»** на индикаторном табло должны гореть 3 индикаторные лампы.

Установить переключатель **«РОД РАБОТЫ»** в положение «**U–1S»**, а переключатель **«ПРЕДЕЛ ИЗМЕРЕНИЯ»** – в положение **«1»**. Закоротить вход **«=100VR»** и ручкой « » установить на индикаторном табло показания "0000" с равновесным изменением знака полярности.

Примечание. Допускается установка показаний «**+ 0000»** или «**- 0000»** как с изменением, так и без изменения знака полярности на противоположный. При установке нуля с преобладанием одного из знака полярности возможно появление на цифровом табло показание «**0001».**

Установить переключатель **«РОД РАБОТЫ»** в положение « » и ручкой « » (установка калибровки) установить на индикаторном табло показание, равное значению, указанному на шильдике вольтметра.

Примечание. Допускается установка на индикаторном табло показания, отличающегося от указанного на шильдике вольтметра на **± 1** знак младшего разряда.

Установить переключатель **«ПРЕДЕЛ ИЗМЕРЕНИЯ»** – в положение **«100».**

Соединить переключатель **«РОД РАБОТЫ»** в положение «**R»** и органами регулировки нуля установить нулевое показание вольтметра.

Соединить между собой гнёзда **«= 100 VR»** и "**89,8 кΩ**". Регулировочными органами калибровки установить показание величины сопротивления на индикаторном табло, равное 89,80 кОм.

Установить переключатель **«ПРЕДЕЛ ИЗМЕРЕНИЯ»** – в положение «**10**

MΩ» и закоротить вход **«= 100 VR»**. Регулировочными органами установки нуля установить нулевые показания индикаторного табло.

Соединить между собой гнёзда **«= 100 VR»** и **«89,8 MΩ»**. С помощью органов калибровки установить показание величины сопротивления на индикаторном табло, равное 8,980 МОм.

Следует помнить, что при большом уровне индустриальных помех в сети работоспособность вольтметра может нарушаться, что проявляется в виде сбоев при индикации показаний на индикаторном табло.

Выключение прибора производится тумблером «СЕТЬ». Положение всех остальных органов управления произвольное.

Порядок выполнения работы.

1. Измерение активного сопротивления.

2. Подготовить вольтметр к работе согласно методике измерений.

3. Установить потенциометр **«ВР. ИНД.»** в положение, обеспечивающее удобное время индикации.

4. Установить переключатель **«ПРЕДЕЛ ИЗМЕРЕНИЯ»** в положение, соответствующее величине измеряемого сопротивления.

5. Измеряемое сопротивление подключить к гнезду **«= 100 VR».**

6. Произвести отсчёт показаний и рассчитать длину кабелей согласно указанию преподавателя. Отсчёт показаний производить не менее, чем через 10 сек.

Содержание отчета по лабораторной работе.

1. Цель работы.

2. Результаты измерений и расчётов (таблицы).

3. Анализ результатов и выводы по лабораторной работе.

4. Возможности прибора В7 – 16А.

4 Лабораторная работа № 4. Установка сетевого адаптера для подключения рабочей станции в вычислительную сеть

Цель работы. Получить основные теоретические сведения и практические навыки по установке (настройки) сетевого адаптера для рабочей станции при ее подключении в локальную сеть.

Теоретическая справка.

Для нормального функционирования сети необходимо правильно настроить операционную систему. Для этого необходимо открыть панель управления, в которой находится апплет настройки сети. При вызове апплета необходимо нажать кнопку **«Добавить»** и установить следующие компоненты:

– клиент для сетей Microsoft;

– сетевая плата;

– сетевой протокол;

– сетевая служба.

Клиент для сетей Microsoft отвечает за взаимодействие системы с сервером, выбор клиента сети отображен на рисунке 3.

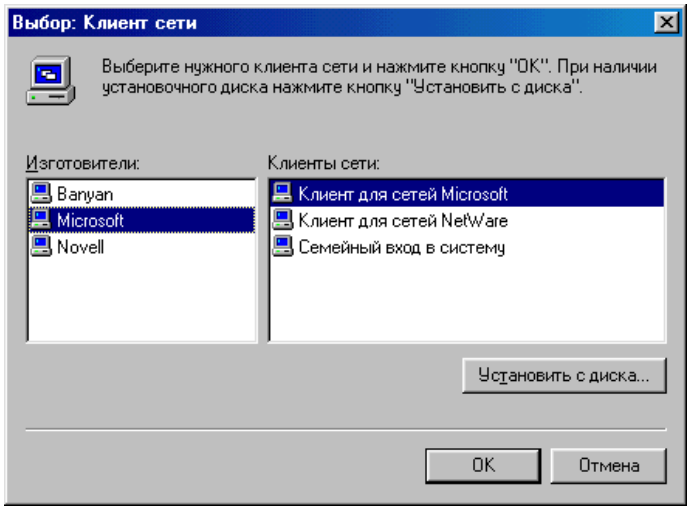


Рисунок 3 – Установка клиента для сетей Microsoft

Сетевая плата является компонентом, выполняющим взаимодействие с физической средой передачи данных. Его выбор производится в соответствии с производителем и названием устройства, пример показан на рисунке 4.

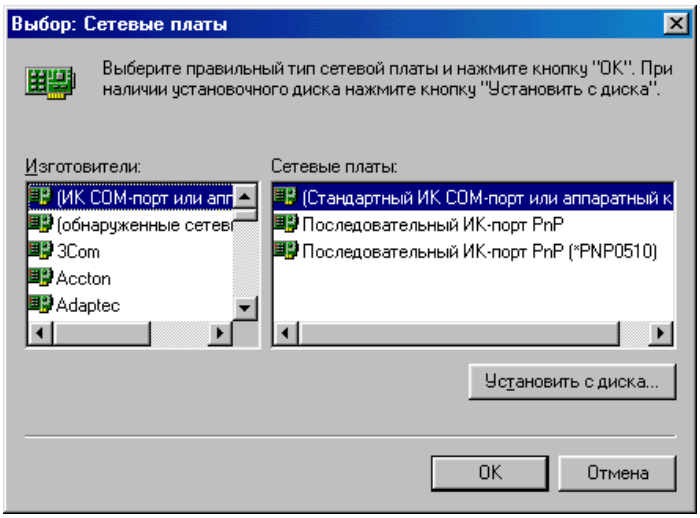


Рисунок 4 – Установка драйверов для сетевой платы

Установка сетевого протокола происходит аналогично (рисунок 5).

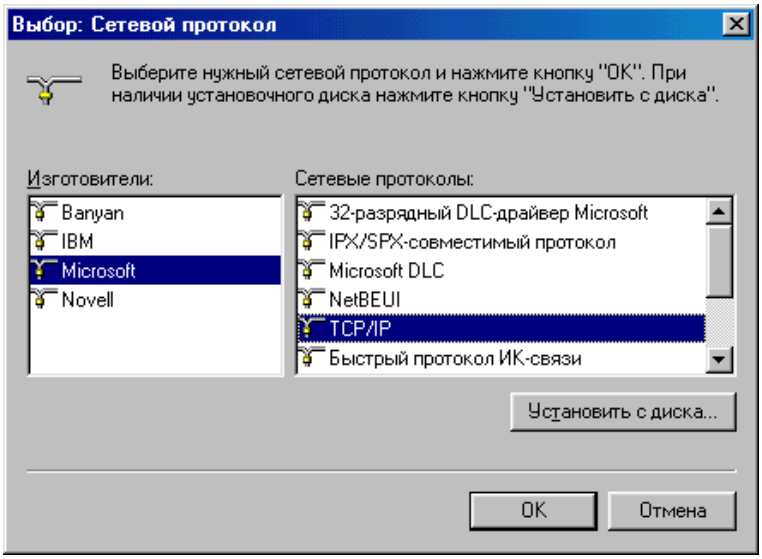


Рисунок 5 – Установка сетевого протокола

Последняя из устанавливаемых компонент – сетевая служба. Она выполняет функции разграничения доступа к общим ресурсам компьютера в сети (рисунок 6).

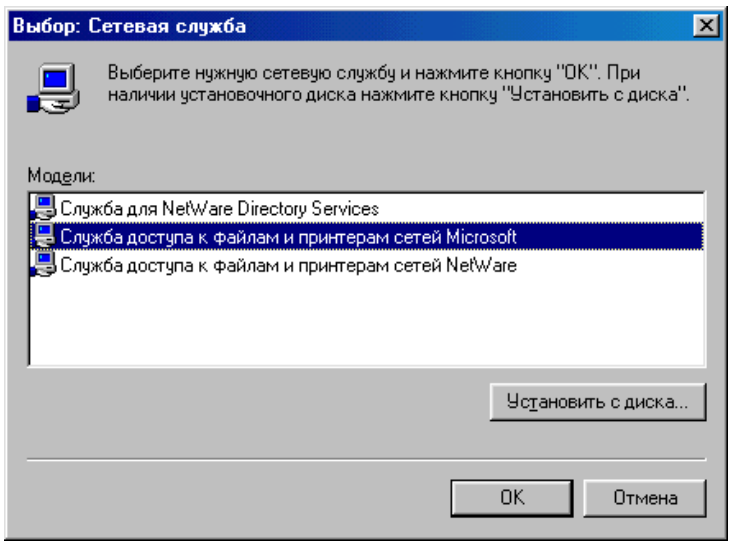


Рисунок 6 – Инсталляция сетевой службы

После установки всех компонент мы получим форму, отображенную на рисунке 7. Для дальнейшей настройки необходимо выделить компонент и нажать кнопку «Свойства», которая на рисунке неактивна, потому что не выбран компонент.

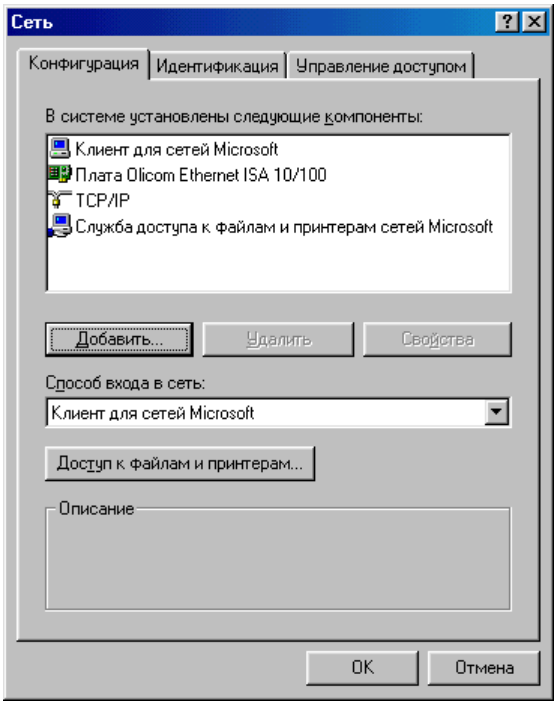


Рисунок 7 – Конфигурация сети после установки всех компонентов

Настройка клиента сетей сводится к прописыванию домена, к которому подключается компьютер при входе в сеть и установки параметра входа с

восстановлением сетевых подключений, как показано на рисунке 8.

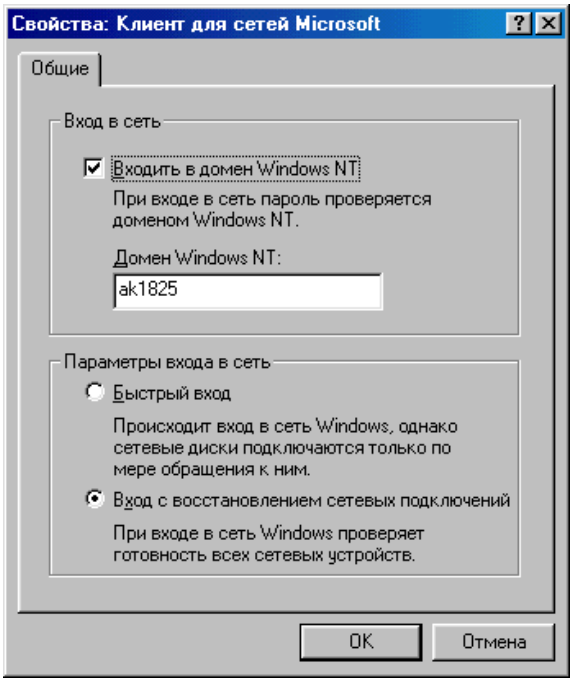


Рисунок 8 – Настройка свойств клиента сети

Настройка драйверов сетевой платы обычно не требуется, так как все установки по умолчанию не создают проблем. Для примера на рисунке 9 показаны вкладки сетевой карты Olicom Ethernet ISA 10/100.

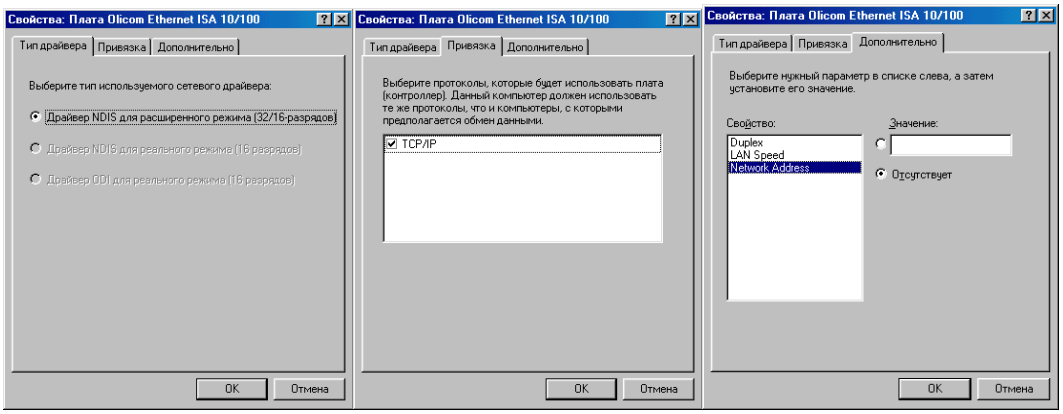


Рисунок 9 – Вкладки свойств сетевой карты

Настройка протокола TCP/IP не требуется, если нет необходимости настраивать сеть на соединение сети с интернетом. Оговоримся только, что если на сервере не работает служба DHCP, то на вкладке IP-адреса необходимо указать его и маску подсети, к которой принадлежит компьютер (рисунок 10) .

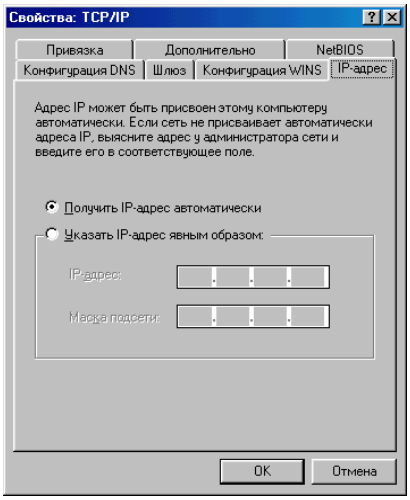


Рисунок 10 – Настройка IP адреса компьютера

На рисунке 11 изображена другая вкладка апплета «Сеть» -«Идентификация», которая позволяет задать имя компьютера и его принадлежность рабочей группе (домену). Имя компьютера и его описание позволяет легко ориентироваться в большом количестве компьютеров в домене.

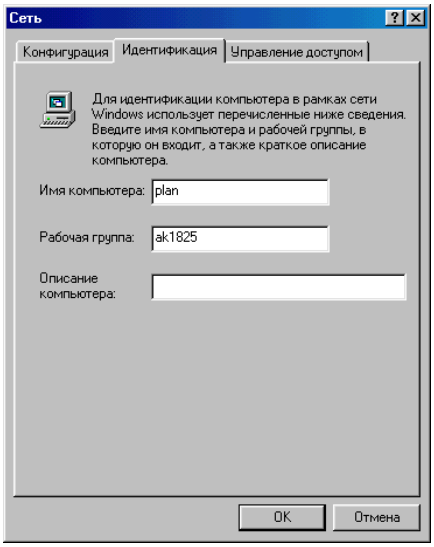


Рисунок 11 – Вкладка идентификации

Управление ресурсами в домене может осуществляться двумя способами: на уровне ресурсов и на уровне пользователей (рисунок 12). Каждый из способов имеет ряд преимуществ, но для облегчения администрирования домена наиболее предпочтительней управление на уровне пользователей, что позволяет указывать пользователей и группы, имеющих доступ к каждому общему ресурсу.

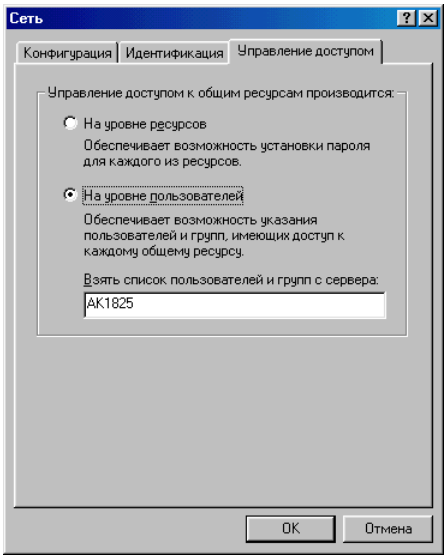


Рисунок 12 – Управление доступом

Порядок выполнения работы.

1. Установить компоненты сети, основываясь на рисунки 3 - 12**;**
2. Оформить отчет.

Содержание отчета.

1. Название и цель работы.
2. Основные теоретические пункты общий сведений.
3. Этапы (процедуру проведения) установки и активации сетевого адаптера.
4. Выводы по выполненной работе.
5. Список использованных источников.

5 Лабораторная работа № 5. Передача информации в вычислительной сети технологии Ethernet (Fast Ethernet)

Цель работы. Определение сетевых параметров рабочей станции вычислительной сети технологии Ethernet (Fast Ethernet).

Теоретическая справка.

Локальные вычислительные сети (ЛВС, LAN) – это распределенные вычислительные системы, объединяющие компьютеры, находящиеся в пределах

одного или нескольких зданий. Узлы локальной сети находятся, как правило, в пределах 3 км.

По масштабам и иерархии построения различают:

– сети рабочих групп (5 - 20 станций);

– сети отделов (20 - 100 станций);

– сети предприятий (корпоративные сети) [6].

Последние часто имеют развернутую структуру сетевых служб и по географии иногда выходят за рамки локальных сетей, образуя кампусные сети, сети с удаленным доступом, а также сети других масштабов, вплоть до корпоративных частных глобальных сетей. Количество станций в корпоративных сетях варьируется в широких пределах: от 20 компьютеров до десятков тысяч.

Одной из характеристик локальных сетей являются пропускная способность, диапазоны которых приведены в таблице 4 [6].

Таблица 4 – Сетевые технологии

Технологии	Топологии	Среды передачи	Активное оборудование	Пропускная способност	Протоколы
Локальные сети	Шина, звезда, кольцо, дерево, смешанн	витая пара, коаксиальный кабель	Сетевые карты, концентраторы, коммутаторы,	10 - 100 Мбит/с	1-2 уровни - Ether-net, Token Ring, FDDI, 3-7 уровни - TCP/IP, NetBIOS /SMB,
Глобальные сети	Ячеистая, смешанная	волоконно-оптические,	Маршрутизаторы, спутники связи, антенны	10 Мбит/с - 100 Гбит/с	1-2 уровни - X.25, ISDN, TM, frame relay 3-7 уровни TCP/IP, ATM
Удаленный доступ	Точка-точка	ТфОП	Модемы	33,6 кбит/с -10 Мбит/с	1-2 уровни - ISDN, SLIP, PPP, RS-232, V.34, V.90; 3-7 уровни - TCP/

Для понимания принципов Ethernet необходимо общее представление о принципах работы компьютерных сетей и разбиения задачи сетевой связи на

уровни, изложенных выше.

Международный стандарт технологии Ethernet - IEEE 802.3.

Технология Ethernet используется для описания всех локальных сетей, использующих метод коллективного доступа к среде передачи данных с опознанием несущей и обнаружением коллизий.

Физическая топология сети – это реальное соединение ее узлов и линий связи. Физическая топология может отличаться от логической.

Логическая топология – это схема соединения, связанная с методом доступа к передающей среде. Поскольку при технологии Ethernet все компьютеры локальной сети имеют возможность одновременного доступа к передающей среде, логическая топология является «шиной». Несмотря на изменение физической топологии в **Fast Ethernet**, при этом не изменился метод доступа к среде, следовательно, логическая топология также не изменилась.

Метод коллективного доступа с опознанием несущей и обнаружением коллизий.

В **Ethernet** все компьютеры сети имеют возможность одновременно получать данные, которые любой из компьютеров начал передавать на общую шину. Кабель, к которому подключены все компьютеры, работает в режиме коллективного доступа. В конкретный момент времени передавать данные на общую шину может только один компьютер в сети. При этом все компьютеры сети обладают равными правами доступа к среде [6].

Принцип коллективного доступа к среде передачи данных.

Когда какая-либо станция А в Ethernet хочет передать кадр станции Б, она пытается вначале определить, что никакая другая станция в это время ничего не передает. В стандарте Ethernet признаком свободной линии является «тишина», то есть напряжение 0 В. В стандарте Fast Ethernet признаком свободного состояния среды является не отсутствие сигналов на шине, а передача по ней специального Idle-символа. Если рабочая станция обнаруживает несущий сигнал, то для нее это является признаком занятости шины и передача данных откладывается, то есть станция переходит в режим ожидания.

В случае если кабель свободен, станция начинает передачу. По окончании передачи кадра все узлы сети обязаны выдержать паузу, называемую **межкадровым интервалом** (Inter Packet Gap, IPG). Эта пауза необходима для приведения сетевых адаптеров в исходное состояние и для обеспечения равных прав всем станциям на передачу данных, то есть для предотвращения монопольного захвата одной станцией общей шины. По окончании паузы станции сети определяют среду как свободную и могут снова начать передачу данных.

Длительность межкадрового интервала для 10-мегабитного Ethernet составляет 9,6 мкс, а для 100-мегабитного Fast Ethernet – в 10 раз меньше, то есть 0,96 мкс. Межкадровый интервал равен времени, необходимому для передачи 12 байт или 96 бит. Если определить в качестве единицы измерения временного интервала время, необходимое для передачи одного бита — битовый интервал (bt), то межкадровый интервал равен 96 bt. Такой способ определения временных интервалов не зависит от скорости передачи данных и часто используется в стандарте Ethernet.

Вторая часть метода описывает способ разрешения конфликтов, возникающих в разделяемой среде передачи. Если две станции начинают передачу одновременно, то происходит конфликт (коллизия). Все узлы сети должны быть способны распознать возникающую коллизию. Четкое распознавание коллизий всеми станциями сети является необходимым условием корректной работы сети Ethernet. Если какая-либо передающая станция не распознает коллизию и решит, что кадр данных передан ею верно, то этот кадр будет утерян. Из-за наложения сигналов при коллизии информация кадра исказится, и он будет отбракован принимающей станцией из-за несовпадения контрольной суммы.

Искаженная информация будет повторно передана каким-либо протоколом верхнего уровня, например транспортным или прикладным, работающим с установлением соединения. Но повторная передача сообщения протоколами верхних уровней произойдет через значительно более длительный интервал времени по сравнению с микросекундными интервалами, которыми оперирует протокол Ethernet. Поэтому если коллизии не будут надежно распознаваться узлами сети

Ethernet, то это приведет к заметному снижению полезной пропускной способности данной сети [6].

Для того чтобы иметь возможность распознать коллизию, каждая станция прослушивает сеть во время и после передачи кадра. Обнаружение коллизии основано на сравнении посылаемого станцией сигнала и регистрируемого сигнала. Если регистрируемый сигнал отличается от передаваемого, то станция определяет эту ситуацию как коллизию.

Пусть первая станция, решив, что шина свободна, начинает передачу кадра. До самой удаленной от нее второй станции этот кадр дойдет не мгновенно, а через некоторый промежуток времени t. Если немного раньше вторая станция, также решит, что шина свободна, и начинает передачу своего кадра, то возникает коллизия. Искаженная информация дойдет обратно до первой станции также через время t. Поэтому коллизия будет обнаружена первой станцией через время $2t$ после начала передачи ею кадра.

Данная характеристика - время разрешения конфликта (время двойного оборота) - имеет огромное значение для эффективности протокола, в частности во многом именно она определяет ограничения на максимальный диаметр сети Ethernet и количество концентраторов на пути распространения сигнала.

Обнаружение коллизии должно произойти до окончания передачи кадра. Отсюда получается простое соотношение между временем, необходимым для передачи кадра минимальной длины $T_{\min}$ и задержкой сигнала при распространении в сети:

$$T_{\min} = 2\,t, \qquad (2)$$

где t – время распространения сигнала по сети Ethernet.

Алгоритм отката [6].

После возникновения коллизии станция, ее обнаружившая, делает паузу, после которой предпринимает следующую попытку передать кадр. Пауза Δt после коллизии является случайной и выбирается по следующему правилу:

$$\Delta t = L \cdot \tau, \quad (3)$$

где τ - интервал отсрочки равный 512 bt, что при скорости 100 Мбит/с составит 5,12 мкс;

L - целое случайное число, выбранное из диапазона $[0; 2^N]$;

N - номер повторной попытки передачи данного кадра.

После первой попытки пауза может либо отсутствовать, либо составлять один или два интервала отсрочки. После второй попытки пауза может либо отсутствовать, либо быть равной одному, двум, трем или четырем интервалам отсрочки и т.д. После 10-й попытки интервал, из которого выбирается пауза, не увеличивается. Таким образом, после десятой попытки передачи кадра случайная пауза может принимать значения от *0* до *1024·512 bt = 524288 bt*. Для Ethernet и Fast Ethernet это соответствует временному диапазону от 0 до 52,4 мс и 5,24 мс соответственно.

Передатчик предпринимает всего 16 последовательных попыток передачи кадра. После 16 конфликтов контроллер отказывается от дальнейших попыток передать кадр и сообщает об этом компьютеру. Все дальнейшие действия по исправлению ситуации должны осуществляться высокоуровневыми протоколами.

Такой алгоритм позволяет разрешить коллизии, когда конфликтующих станций немного, а также ликвидировать их за приемлемое время, когда множество станций пытается передавать одновременно.

Формат кадра Ethernet.

Максимальный размер кадра Ethernet (таблица 5) составляет 1526 байт (12208 бит), а минимальный - 72 байт (576 бит). При частоте передачи 10 МГц время передачи пакета минимальной длины составляет 57,6 мс. Это время несколько больше, чем удвоенное время распространения сигнала, равное 51,2 мс, следовательно условие (1) выполняется. Последняя цифра получена исходя из максимально допустимого в Ethernet расстояния между узлами в 2500 м.

Каждый кадр начинается с преамбулы длиной 7 байт, причем каждый байт преамбулы представляет собой чередующуюся последовательность единиц и нулей.

Преамбула позволяет принимающей стороне подстроиться под передающую станцию, т. е. синхронизироваться с ней. Следом за преамбулой идет стартовый байт (10101011), сигнализирующий о начале кадра [6].

Таблица 5 - Формат кадра Ethernet

Байт	7	1	6	6	2	0-1500	4
Поле	Преамбула	Начало	Адрес получате	Адрес отправите	Длина поля	Данные	Контрольная

Далее кадр содержит два 6-байтных поля адреса – получателя и отправителя. Если сетевая плата Ethernet определяет, что адрес получателя совпадает с ее собственным, то, считав кадр, она передает его для дальнейшей обработки на более высокие уровни. Если адреса не совпадают, то кадр игнорируется. Адреса Ethernet могут быть обычными, групповыми и широковещательными. Если все биты адреса равны единице, то это широковещательный адрес, и такой пакет предназначен всем станциям.

Поле длины кадра состоит из двух байтов и определяет длину поля данных (от 0 до 1500 бит). Однако, ввиду ограничений на минимальную длину кадра, поле данных не может быть короче 46 байт. Если же объем передаваемых данных меньше, то поле данных дополняется заполняющими битами.

Заканчивается кадр концевиком – контрольной последовательностью. Она служит для проверки кадра на наличие ошибок.

Развитие спецификации Ethernet Технологии Fast Ethernet и Gigabit Ethernet являются дальнейшим развитием Ethernet. Сети Fast Ethernet имеют номинальную пропускную способность в полудуплексном режиме 100 Мбит/с, сети Gigabit Ethernet – 1 Гбит/с. В полнодуплексном режиме при использовании двух пар проводов эти значению удваиваются.

Fast Ethernet и Gigabit Ethernet имеют другое коммуникационное оборудование, сетевые карты, но часто обратно совместимы с Ethernet. Качественные принципы работы Fast и Gigabit Ethernet в общих чертах сходны с Ethernet, различия в основном в количественных характеристиках.

В таблице 6 приведены физические характеристики различных спецификаций Ethernet [6].

Таблица 6 - Разновидности Ethernet и их физические характеристики

Стандарт	Физическая специфика-ция	Кабели, разъемы	Ограничения на длину физ. сегмента, м	Макс. число повторителей	макс. число станций	D сети, м
Ethernet (IEEE 802.3i)	10BaseTX	2ВП UTP3-4-5, RJ-45	100	4	1024	500
	10BaseF	ОМ ОВ / ММ ОВ 62.5, разъемы ST	1000/5000	-	2	1000/ 5000
Fast Ethernet (IEEE 802.3u)	100BaseTX	2ВП UTP, STP Type 1, разъемы RJ-45	100	1 класса I / 2 класса II (кабель между повторит. – до 5 м)	1024	200-320
	100BaseFX	мм ОВ 62,5, 125 мкм, разъемы ST, SC	160 (rep) / 412 (полудуплекс)/ 2000 (полнодуплекс н.)			200-320
	100BaseT4	4ВП UTP3 -4-5, RJ-45	100			200-320
Gigabit Ethernet (802.3z)	1000BaseLX	мм ОВ/ ОМ ОВ, разъемы ST, SC	316 (550/3000)	-	2	300/ 550
	1000BaseSX	ММ ОВ 62.5/50 мкм разъемы ST, SC	275 (300/550)	-		300/ 550
	1000BaseCX	коаксиал, (ВП STP), RJ-45	25	-		25
(802.3ab)	1000BaseT	ВП STP5-6 RJ-45	100	-		100

Условные сокращения. 2ВП – 2-жильная, 4ВП – 4-жильная витая пара; ОВ – оптоволокно, ММ – многомодовое, ОМ – одномодовое.

Задание к проведению лабораторной работы.

Определение сетевых параметров рабочей станции.

1. Определите параметры протокола TCP/IP вашего компьютера. Для этого проделайте следующие действия. Выберете меню ***Пуск - Настройка – Панель управления – Сеть – Устройства и протоколы – TCP/IP***. Нажмите кнопку ***Свойства***.

2. Определите следующие параметры протокола:

- IP адрес сетевого адаптера;
- сетевую маску;
- адрес шлюза по умолчанию;
- адрес основного и вспомогательного сервера DNS.

3. Определите физический адрес сетевого адаптера вашего компьютера и его доменное имя. Для этого нужно в командной строке (меню ***Пуск – Программы – Стандартные – Сеанс MSDOS*** или в меню ***Пуск – Выполнить*****)** ввести команду ***Winipcfg*** (для операционных систем Windows 2000/XP/2003 – ввести команду **>ipconfig –all**).

4. Определите, открыт ли сетевой доступ к диску вашего компьютера. Для этого щелкните правой клавишей на значке диска и в открывшемся контекстном меню выберите значение ***Доступ***. Определите также емкость диска (меню ***Свойства***).

5. Определите быстродействие и память вашего компьютера. Для этого щелкните правой кнопкой мыши на значке ***Мой компьютер*** и в открывшемся контекстном меню выберите значение ***Свойства***, а затем вкладку ***Общие.***

6. Создайте файл в текстовом редакторе WORD и занесите в него следующие сведения:

- ваше Ф.И.О.;
- группа;
- имя компьютера по протоколу NetBIOS (имя в сетевом окружении);

– физический адрес сетевого адаптера и его тип;

– IP адрес и маску;

– адрес шлюза по умолчанию;

– адрес серверов DNS;

– параметры вашего компьютера - тактовая частота процессора, оперативная память, размер диска, параметры сетевого доступа к диску.

Содержание отчета по лабораторной работе.

1. Название и цель работы.
2. Дать характеристики локальных сетей.
3. Представить форматы кадров технологии Ethernet.
4. Представить характеристики сетевых параметров рабочей станции.
5. Выводы по выполненной работе.
6. Список использованных источников.

Контрольные вопросы.

1. Каким уровням ISO/OSI соответствует спецификация Ethernet.
2. Отличие физической топологии от логической. Примеры.
3. В чем заключается метод коллективного доступа к среде с опознанием несущей и обнаружением коллизий? Опишите алгоритм отката.
4. Сигналами какого типа и формы передается информация в сетях Ethernet.
5. Размер кадра, номинальная битовая скорость передачи данных, величина адресного пространства, совместимость различных поколений Ethernet.
6. Какой вид сегментации – физическая или логическая – эффективней с точки зрения скорости работы сети.
7. По какой причине в сетях не используется только этот вид сегментации, а используются оба.
8. Основные физические ограничения на оборудование различных поколений Ethernet.

6 Лабораторная работа № 6. Исследование вычислительной сети топологии «шина»

Цель работы. Целью работы является исследование характеристик шинной локальной вычислительной сети (ЛВС), использующей множественный доступ с контролем несущей и обнаружением коллизий (МДКН/ОК). Для исследования используется имитационная модель ЛВС.

Теоретическая справка.

Метод множественного доступа с контролем несущей частоты и обнаружением коллизий (МДКН/ОК) является наиболее совершенным и применяется в ЛВС технологии Ethernet.

При методе МДКН/ОК каждый из абонентов прослушивает канал (шину) до того, как приступит к передаче. После освобождения канала абоненты могут приступить к передаче своего пакета. Из-за разницы в задержке распространения (наличия «окна конфликтов») два и более абонентов могут начать передачу в одно время, что приводит к наложению пакетов в шине и их искажению. При МДКН/ОК вводится прослушивание шины как до начала передачи (контроль несущей), так и во время передачи (обнаружение наложения). Каждый из отправителей, обнаружив наложение, сразу же прекращает передачу. При этом потерянное на конфликт время будет сравнительно небольшим по сравнению с общим временем передачи пакета.

После прекращения передачи, конфликтующие абоненты повторяют попытку передачи своих пакетов через случайным образом сформированные тайм-ауты (для минимизации возможности нового конфликта). Если при повторной попытке снова возникает конфликт, то снова выбирается случайный период ожидания, но большего размера и т.д. После определенного числа попыток передать пакет устройство прекращает передачу и сообщает пользователю о невозможности передачи [2].

Наиболее распространенным является экспоненциальный алгоритм отсрочки передачи. Время задержки определяется в условных единицах равных 51,2 мкс. (эта величина больше типичной круговой задержки в шине ЛВС). Диапазон

генерируемых случайных чисел в течение первых 10 попыток изменяется экспоненциально от (0,1) до (0,1023). С 11-й по 16-ю попытку диапазон остается неизменным (0,1023). Если 16-я попытка заканчивается неудачно, то канальный уровень отказывается от передачи пакета и оповещает об этом верхний протокольный уровень. Обычно это связано с нарушением кабеля.

В данной лабораторной работе используется имитационная модель шинной ЛВС с методом МДКН/ОК с изменяющимся числом (N) абонентских станций.

Модель имитирует поступление пакетов в соответствии с заданными законом и средней интенсивностью от абонентов для передачи их по ЛВС.

Модель позволяет собирать статистические данные о буферных накопителях как отдельных абонентских станций (АС), так и о суммарном объеме накопившейся не обслуженной нагрузки [2].

На модели могут быть получены характеристики загрузки шины ЛВС. Определяются также временные характеристики процесса доставки пакетов по ЛВС.

Входными переменными модели являются:

– число абонентских станций ЛВС;

– среднее время между моментами поступления пакетов от абонента (в модели принят экспоненциальный закон для потока входящей нагрузки);

– среднее время передачи пакета по шине (предполагается, что длительность передачи распределена экспоненциально);

– длительность интервала конфликтов в шине;

– время моделирования.

Порядок выполнения работы.

1. В таблице исходных данных приведены основные параметры исследуемой кольцевой ЛВС. Исходные данные вводятся в модель в интерактивном режиме в начале прогона.

2. Производится серия имитационных экспериментов с целью получения основных характеристик ЛВС при различном числе подключенных АС. Количество абонентов меняется в пределах 3 - 50.

Содержание отчета по лабораторной работе.

Отчет по лабораторной работе должен содержать следующие сведения.

1. Основные параметры ЛВС при каждом из экспериментов:

– загрузка шины ЛВС (смотри параметр AVE.C. в статистике STORAGE BUS);

– среднее время доставки пакета (см. параметр MEAN в статистике TABLE TPAC);

– средняя длина очереди у абонента (смотри параметр X9/1000);

– среднее число попыток на одну успешную передачу пакета (смотри параметр X8/1000);

– число отказов в передаче пакетов (смотри параметр X3).

2. Зависимости времени доставки пакета ($T_{дост}$), коэффициента загрузки шины ($K_{загр}$), среднего объема данных в буферном накопителе абонентской станции ($V_{ср.\ бн}$) и числа попыток передачи ($K_{поп}$) от текущего числа АС в ЛВС (N_{AC}), т.е.:

– $T_{дост} = F\{ N_{AC}\}$;

– $K_{загр} = F\{N_{AC}\}$;

– $V_{ср.\ бн} = F\{N_{AC}\}$;

– $K_{поп} = F\{ N_{AC}\}$.

3. Выводы по результатам моделирования.

Исходные данные для проведения лабораторной работы.

Таблица 7 - Исходные данные для проведения имитационных экспериментов

№ варианта	Средняя интенсивность	Интервал конфликтов	Среднее время передачи пакета	$T_{мод}$, мин
1	2	3	4	5
1	2000	2	14	0,5
2	3000	3	20	0,7
3	2500	2	10	0,6
4	1500	2	25	0,5
5	5000	3	18	0,8
6	4000	2	21	0,7
7	2000	2	10	0,6
8	3000	3	25	0,5

Продолжение таблицы 7

1	2	3	4	5
9	2500	2	18	0,8
10	1500	2	21	0,7
11	5000	3	25	0,5
12	4000	2	18	0,8
13	1500	2	21	0,7
14	5000	3	10	0,6
15	4000	2	25	0,5

7 Лабораторная работа № 7. Исследование вычислительной сети топологии «кольцо»

Цель работы. Целью работы является исследование характеристик кольцевой вычислительной сети с маркерным методом передачи информации. Для исследования используется имитационная модель ЛВС.

Теоретическая справка.

Кольцевые ЛВС являются одним из самых распространенных типов локальных сетей. В кольцевых ЛВС применяются три различных метода доступа: метод вставки регистра, метод тактируемого доступа и передача маркера.

При методе передачи маркера используется специальная последовательность символов, передаваемых по кольцу – маркер (рисунок 13) [2].

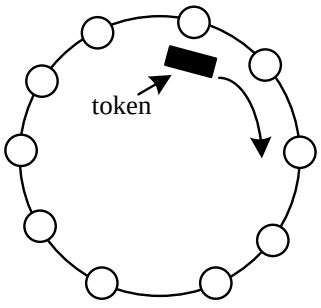


Рисунок 13 - ЛВС с маркерным методом доступа

В случае необходимости передачи данных абонентская станция (АС) ожидает прихода по кольцу к ней маркера. Получив маркер, АС удаляет его из кольца и посылает данные в кольцо. Затем АС ждет поступления обратно своего пакета, удаляет его из кольца и отправляет маркер следующему устройству в кольце. Получившая маркер станция может при необходимости отправить свой пакет по сети и т.д.

Описание работы.

В данной лабораторной работе используется имитационная модель локальной кольцевой маркерной сети с переменным числом (N) абонентских станций.

Модель имитирует поступление сообщений в соответствии с заданными законом и средней интенсивностью от абонентов для передачи их по ЛВС. Поступившие от абонентов сообщения могут иметь различную длину и, следовательно, формируются как некоторое количество подготовленных для передачи по сети пакетов. Станция, имеющая подготовленные пакеты ждет возможности занятия канала ЛВС – поступления маркера. Получив маркер, станция отправляет очередной пакет, дожидается его прихода обратно по кольцу, проверяет правильность передачи, и отсылает маркер следующей станции кольца.

Модель позволяет собирать статистические данные о буферных накопителях как отдельных АС, так и о суммарном объеме накопившейся не обслуженной нагрузки. На модели могут быть получены характеристики загрузки канала ЛВС. Определяются также временные характеристики процесса доставки пакетов по ЛВС.

Входными переменными модели являются:

– число абонентских станций ЛВС;

– среднее время меду моментами поступления сообщений от абонента;

– закон распределения моментов поступления сообщений от абонентов;

– закон распределения длин поступающих сообщений (в пакетах);

– длительность цикла передачи пакета по ЛВС;

– время моделирования.

Порядок выполнения работы.

1. В таблице исходных данных приведены основные параметры исследуемой

кольцевой ЛВС.

2. Исходный закон распределения количества пакетов для категорий сообщений меняется первоначально в тексте моделирующей программы (функция 110 РАК).

3. Остальные исходные данные вводятся в модель в интерактивном режиме в начале прогона.

4. Производится серия имитационных экспериментов с целью получения основных характеристик ЛВС при различном числе подключенных АС. Количество абонентов меняется в пределах 3 - 50.

Содержание отчета по лабораторной работе.

Отчет по лабораторной работе должен содержать следующие сведения.

1. Название и цель работы.

2. Основные параметры ЛВС при каждом из экспериментов: загрузка канала ЛВС, среднее и максимальное число ожидающих отправки пакетов у абонентов, среднее время доставки пакета адресату по сети и его распределение.

3. Зависимости времени доставки пакета ($T_{дост}$), коэффициента загрузки канала ($K_{загр}$), среднего и максимального объема данных в буферном накопителе абонентской станции ($V_{ср.бн}$ и $V_{max.\ бн}$) от текущего числа АС в ЛВС (N_{AC}), т.е.:

- $T_{дост} = F\{N_{AC}\}$;
- $K_{загр} = F\{N_{AC}\}$;
- $V_{ср.\ бн} = F\{N_{AC}\}$;
- $V_{max.\ бн} = F\{N_{AC}\}$;

4. Выводы по результатам моделирования.

5. Список использованных источников.

Исходные данные для проведения лабораторной работы.

Таблица 8 - Исходные данные для проведения имитационных экспериментов

№ варианта	Средняя интенсивность	Категории сообщений	Время передачи (мс)	$T_{мод}$, (мин)
1	2000	1 — 20 % 2 — 60 % 3 — 20 %	14	1
2	3000	1 — 10 % 2 — 50 % 3 — 40 %	20	0,8
3	2500	1 — 30 % 2 — 60 % 3 — 10 %	10	1,2
4	1500	1 — 20 % 2 — 30 % 3 — 50 %	25	1,3
5	5000	1 — 40 % 2 — 50 % 3 — 10 %	18	1,1
6	4000	1 — 20 % 2 — 40 % 3 — 40 %	21	0,9

8 Лабораторная работа № 8. Расчет параметров сети Ethernet

Цель работы. Проектирование сети Ethernet и Fast Ethernet.

Теоретическая справка.

Задача анализа сети Ethernet возникает при большой протяженности сети (диаметр > 2,5 км) и числе последовательно установленных повторителей больше двух.

Рассматривают две модели проектирования - *Модель 1* и *Модель 2*.

Если сеть удовлетворяет *Модели 1*, то сеть спроектирована верно. Если сеть не удовлетворяет *Модели 1*, то следует применить *Модель 2*, и если сеть будет удовлетворять *Модели 2*, то считают, что сеть спроектирована верно.

В *Модели 1* выделяют три условия, которым должна удовлетворять проектируемая сеть. Согласно *первому условию* (рисунок 14) путь между двумя узлами может содержать:

- до пяти сегментов;
- до четырех повторителей;
- два трансивера;
- два трансиверных кабеля [6].

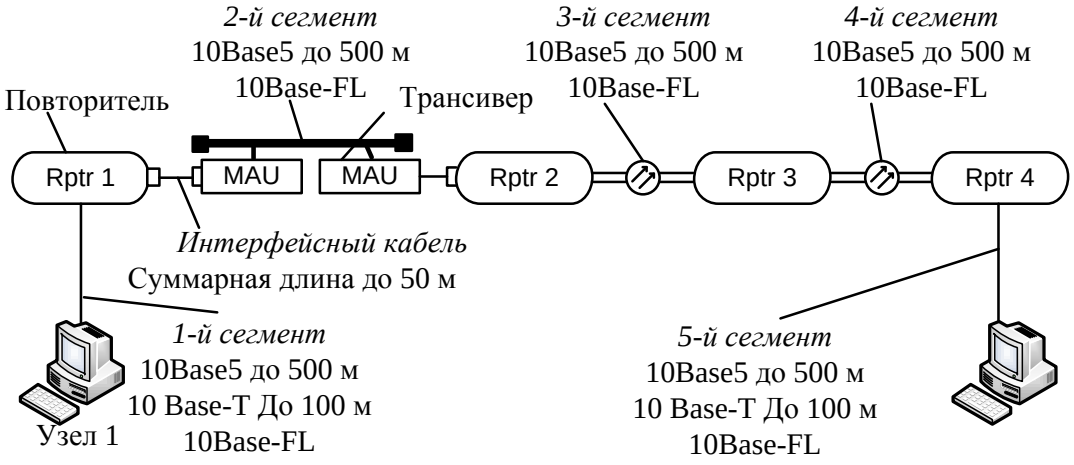


Рисунок 14 - Графическая интерпретация первого условия по *Модели 1*

Второе условие. Если путь содержит пять сегментов и четыре повторителя, то в нем может быть до трех коаксиальных сегментов. Тогда длина волоконно-оптических сегментов может достигать 500 м.

Третье условие. Если путь между двумя узлами состоит из трех повторителей и четырех сегментов, то длина волоконно-оптического участка между повторителями не должна превышать 1000 м (рисунок 15, а), а между повторителем и узлом – 400 м (рисунок 15, б). При этом число коаксиальных сегментов может достигать четырех.

10Base-FL до 1000 м
а)

10Base-FL до 400 м
б)

Рисунок 15 - Максимальная длина волоконно-оптического участка по третьему условию

Повторитель *(RPTR)* необходим для объединения сегментов сети, восстановления формы сигналов, временных характеристик и регенерации преамбулы.

Трансивер *(MAU)* служит для подключения толстого коаксиального кабеля или волоконно-оптического кабеля.

При использовании *Модели 2* проверяются.

1. Задержка распространения сигнала на двойном пробеге *RTD,* которая не должна превышать максимально допустимой величины $RTD_{max}=575$ ВТ при запасе надежности $SF = 5$ ВТ. Расчет *RTD* выполняют отдельно от узла *i* к узлу *j* и от узла *j* к узлу *i* по формуле:

$$RTD = base + RTDM \cdot L, \qquad (4)$$

где *base* - задержка в сетевом элементе (узле, повторителе);

RTDM - задержка на двойном пробеге в кабельном сегменте в 1 м;

L - длина сегмента (следует отметить, что длина трансиверного кабеля уменьшается на 2 м от действительного значения).

2. Уменьшение межкадрового интервала *SVV*, которое должно быть не более 49 ВТ. Уменьшение межкадрового интервала происходит в повторителях в процессе регенерации преамбулы, ретрансляции кадров. Уменьшение задержки учитывается на начальном и среднем сегментах сети [6].

Таблица 9 - Максимально допустимые задержки на устройствах Ethernet и кабельных сегментах

Тип сегмента	Max длина, м	base, BT			RTDM , BT/м
		Начальный сегмент	Средний сегмент	Конечный сегмент	
10Base5	500	11,75	46,5	169,5	0,0866
10Base2	185	11,75	46,5	169,5	0,1026
10Base–T	100	15,25	42	165	0,113
10Base–FL	2000	12,25	33,5	156,5	0,1
Трансиверный кабель	48 (+2)	–	–	–	0,102

Методику расчета рассмотрим на примере сети, приведенной на рисунке 16. Для участка между первым и вторым узлами имеем:

- число повторителей - четыре;
- число сегментов - пять;
- число трансиверов - два;
- длина трансиверного кабеля - 50 м;
- смешанных сегментов - три;

Следовательно, данный участок удовлетворяет *Модели 1.*

Для участка между первым и третьим узлами число трансиверов больше двух, следовательно, этот участок не удовлетворяет *Модели 1.* Поэтому необходимо воспользоваться *Моделью 2* для принятия решения о правильности проектирования сети.

Выполним расчет задержки на двойном пробеге *RTD* между первым и вторым узлами. Результирующая формула будет содержать пять слагаемых (по числу сегментов), вычисляемых по формуле (4). Для первого сегмента *base* = 15,25 BT (поскольку он начальный, а тип сегмента 10Base– T), *RTDM = 0,113* BT/м и *L* = 100.

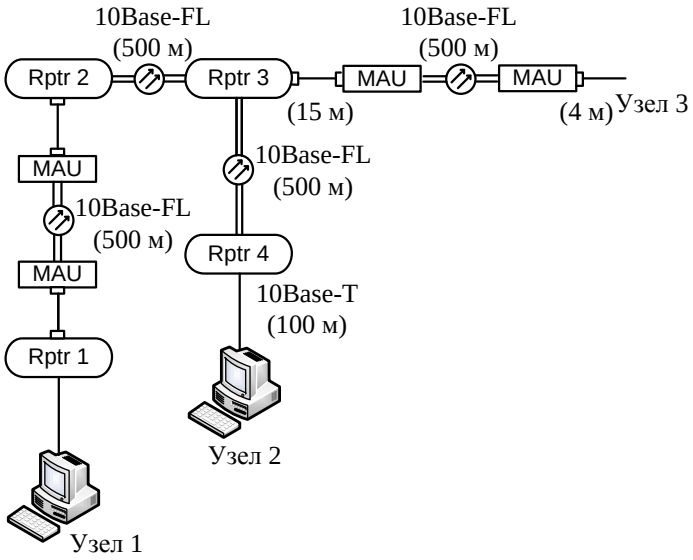


Рисунок 16 - Проектируемая сеть Ethernet

Используя формулу (4), следует выполнить аналогичный расчет для остальных четырех сегментов. Результаты вычислений всех возможных значений *RTD* приведены ниже [6].

$$RTD(1,2) = (15{,}25 + 0{,}113 \cdot 100) + ((50\text{-}2) \cdot 0{,}1026 + 33{,}5 + 0{,}1 \cdot 500) +$$

$$+ (33{,}5 + 0{,}1 \cdot 500) + (33{,}5 + 0{,}1 \cdot 500) + (165 + 0{,}113 \cdot 100) = 458 \text{ BT}$$

$$5{,}25 + 0{,}1\ 13 \cdot 100) + (33{,}5 + 0{,}1 \cdot 500) + (33{,}5 + 0{,}1 \cdot 500) +$$

$$+ ((50 - 2) \cdot 0{,}1026 + 33{,}5 + 0{,}1 \cdot 500) + (165 + 0{,}113 \cdot 100) = 458 \text{ BT}$$

$$RTD(1,3) = (15{,}25 + 0{,}113 \cdot 100) + ((50\text{-}2) \cdot 0{,}1026 + 33{,}5 + 0{,}1 \cdot 500) +$$

$$+ (33{,}5 + 0{,}1 \cdot 500) + ((19 - 2) \cdot 0{,}1026 + 156{,}5 + 0{,}1 \cdot 500) = 407 \text{ BT}$$

$$RTD(3,1) = ((19\text{-}2) \cdot 0{,}1026 + 12{,}25 + 0{,}1 \cdot 500) + (33{,}5 + 0{,}1 \cdot 500) +$$

$$+ ((50 - 2) \cdot 0{,}1026 + 33{,}5 + 0{,}1 \cdot 500) + (165 + 0{,}113 \cdot 100) = 412 \text{ BT}$$

$$RTD(2,3) = (1\ 5{,}25 + 0{,}113 \cdot 100) + (33{,}5 + 0{,}1 \cdot 500) +$$

$$+ ((19 - 2) \cdot 0{,}1026 + 156{,}5 + 0{,}1 \cdot 500) = 318 \text{ BT}$$

$$RTD(3,2) = ((19\text{-}2) \cdot 0{,}1026 + 12{,}25 + 0{,}1 \cdot 500) + (33{,}5 + 0{,}1 \cdot 500) +$$

$$+ (165 + 0{,}113 \cdot 100) = 324 \text{ BT}$$

Видно, что *RTD* не превышает 575 BT. Для окончательного принятия решения о правильности проектирования сети рассчитаем величину уменьшения межкадрового интервала *SVV* с учетом таблицы 10.

Таблица 10 - Вносимое уменьшение межкадрового интервала

Тип сегмента	Начальный сегмент, BT	Средний сегмент, BT
10Base2, 10Base5	16	11
10Base–FL, 10Base–T	10,5	8

Расчет *SVV* выполняется путем суммирования значений вносимого уменьшения межкадрового интервала от начального и средних сегментов. При этом учитываются тип сегмента и его местоположение. Для участка сети между первым и вторым узлами имеется начальный сегмент 10Base–T, который вносит уменьшение межкадрового интервала в 10,5 BT и три сегмента 10Base-FL, дающие вклад по 8 BT. Поэтому результирующее значение $SVV(1,2) = 10,5 + 8 + 8 + 8 = 34,5$ BT . Аналогично выполним расчет для остальных.

$$SVV(2,1) = 10,5 + 8 + 8 + 8 = 34,5 \text{ BT},$$

$$SVV(1,3) = 10,5 + 8 + 8 = 26,5 \text{ BT},$$

$$SVV(3,1) = 10,5 + 8 + 8 = 26,5 \text{ BT},$$

$$SVV(2,3) = 10,5 + 8 = 18,5 \text{ BT},$$

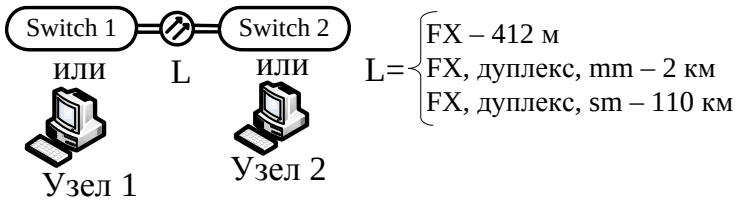
$$SVV(3,2) = 10,5 + 8 = 18,5 \text{ BT}.$$

Анализ показывает, что *SVV* не превышает 49 BT. Следовательно, рассматриваемая сеть спроектирована верно.

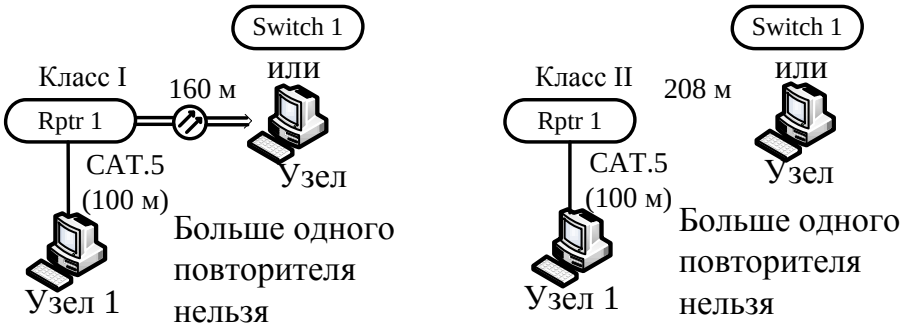
Расчет сети Fast Ethernet.

Для сети Fast Ethernet также приняты две модели, которым должна соответствовать проектируемая сеть.

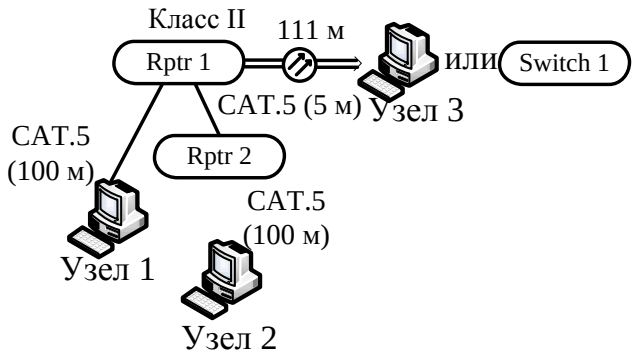
Модель 1 определяет четыре типовых схемы (рисунок 17), на которых указываются типы сегментов и их предельные длины:



а) *модель 1A*



б) *модель 1B* в) *модель 1C*



г) *модель 1D*

Рисунок 17 - Графическая интерпретация первого условия по *Модели 1* для сети Fast Ethernet

Все приведенные длины сегментов предельны. При установке на удаленном конце коммутатора с подключенными к нему рабочими станциями суммарная длина кабелей должна соответствовать приведенной в *Модели 1*.

Использование *Модели 2* заключается в вычислении задержки распространения сигнала на двойном пробеге *RTD* , последнее не должно превышать 512 BT .

Уменьшение межкадрового интервала *SSV* не рассчитывается, поскольку в сети имеется небольшое число повторителей.

Таблица 11 - Максимально допустимые задержки на устройствах Fast Ethernet и кабельных сегментах

Устройство/ кабельный сегмент	Максимальная задержка на двойном пробеге, RTD_{max}, BT
Узел/ коммутатор	100 (суммарно для двух оконечных устройств)
Повторитель *класса I*	140
Повторитель *класса II*	92
Витая пара *CAT*.5 для 1 м	1,112
Оптоволокно для 1 м	1,0

Проверим с помощью *Модели 2* справедливость *Модели 1.*

Так в *Модели 1A* (рисунок 17, а) имеется два узла с суммарной задержкой в 100 BT и оптоволоконный сегмент длиной 412 м с удельной задержкой в 1 BT. Кроме того, *RTD*(1,2) = *RTD*(2,1) = 100 + 412·1 = 512 BT. Аналогично для остальных моделей:

– *Модель 1 B*: *RTD* = 100 + 100·1,112 + 140 + 160·1 = 511,2 BT;

– *Модель 1C: RTD* = 100 + 100·1,112 + 92 + 208·1 = 511,2 BT;

– *Модель 1D*: *RTD*(1,2) = 100 + (100 + 100 + 5)·1,112 + 2·92 = 511,96 BT.

$$RTD(1,3) = 100 + 100 \cdot 1{,}112 + 92 + 111 \cdot 1 = 414{,}2 \text{ BT},$$

$$RTD(2,3) = 100 + (100 + 5) \cdot 1{,}112 + 2 \cdot 92 + 111 \cdot 1 = 511{,}76 \text{ BT}.$$

Из приведенных расчетов видно, что значение *RTD* не превышает 512 BT.

Рассмотрим пример вычисления максимальной длины волоконно-оптического кабеля для сети (рисунок 18).

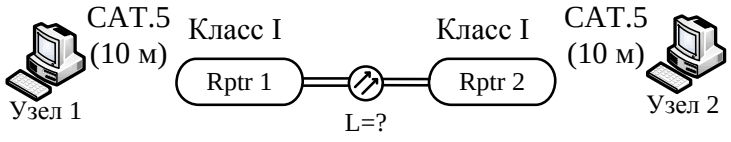


Рисунок 18 - Сеть Fast Ethernet

Составим по формуле выражение для вычисления максимальной задержки распространения сигнала на двойном пробеге с учетом всех элементов сети, приведенной на рисунке 18. Имеем:

$$RTD_{max}=100 + (10 + 10)\cdot 1{,}112 + 2\cdot 140 + L\cdot 1 = 512 \text{ BT}$$

Из данного уравнения находим максимальную длину волоконно-оптического участка *L*, которая будет равна 109,8 м.

Задание к проведению лабораторной работы.

Самостоятельно составить сеть Ethernet, в которой должно быть не меньше пяти сегментов. Выполнить анализ сети по *Модели 1* и *Модели 2.*

Содержание отчета по лабораторной работе.

1. Название и цель работы.
2. Представит логическую схему сети.
3. Представить анализ сети по *Модели 1* и *Модели 2.*
4. Привести расчет сети Fast Ethernet.
5. Список использованных источников.

Контрольные вопросы.

1. Дайте понятие коллизионного домена.
2. Объясните принцип протокола CSMA/CD.
3. Объясните назначение концентратора.
4. Объясните назначение коммутатора.
5. Приведите и поясните формат кадра Ethernet.

9 Лабораторная работа № 9. Расчет конфигурации сети Ethernet

Цель работы. Изучение принципов построения сетей по стандарту Ethernet и приобретение практических навыков оценки корректности их конфигурации.

Необходимое оборудование: калькулятор.

Теоретическая справка.

Основные характеристики и ограничения технологии Ethernet приведены в таблицах 12 и 13.

Таблица 12 – Общие ограничения для всех стандартов Ethernet

Характеристика	Значение
Номинальная пропускная способность	10 Мбит/с
Максимальное число станций в сети	1024
Максимальное расстояние между узлами в сети	2500 м (в 10Base-FB –2750 м)
Максимальное число коаксиальных сегментов в сети	5

Таблица 13 – Параметры спецификаций физического уровня для стандарта Ethernet

Параметр	10Base-5	10Base-2	10Base-T	10Base-F
Кабель	Толстый коаксиальный кабель RG-8 или RG-11	Тонкий коаксиачьный кабель RG-58	Неэкраннро-ванная витая пара категорий 3, 4, 5	Многомодовый ВОК
Максимальная длина сегмента, м	500	185	100	2000
Максимальное расстояние между узлами сети (при использовании повторителей), м	2500	925	500	2500 (2740 для 10Base-FB)
Максимальное число станций в сегменте	100	30	1024	1024
Максимальное число повторителей между любыми станциями сети	4	4	4	4 (5 для 10Base-FB)

Наиболее часто приходится проверять ограничения, связанные с длиной отдельного сегмента кабеля, а также количеством повторителей и общей длиной сети.

Правила «5-4-3» для коаксиальных сетей и «4-х концентраторов» для сетей на основе витой пары и оптоволокна не только дают гарантии работоспособности сети, но и оставляют большой «запас прочности» сети. Например, если посчитать время двойного оборота в сети, состоящей из 4 повторителей 10Base-5 и 5 сегментов максимальный длины 500 м, то окажется, что оно составляет 537 битовых интервала. А так как время передачи кадра минимальной длины (вместе с преамбулой), составляющей 72 байт, равно 575 битовым интервалам, то видно, что разработчики стандарта Ethernet оставили 38 битовых интервала в качестве запаса для обеспечения надежности. Тем не менее в документах комитета IEEE 802.3

утверждается, что и 4 дополнительных битовых интервала создают достаточный запас надежности.

В таблицах 14 и 15 приводятся исходные данные о задержках, вносимых повторителями и различными средами передачи данных, для самостоятельного расчёта для максимального количества повторителей и максимальной общей длины сети [6].

Таблица 14 – Данные для расчета значения PDV

Тип сегмента	База левого сегмента, bt	База промежуточного сегмента, bt	База правого сегмента, bt	Задержка среды на 1 м, bt	Максимальная длина сегмента, м
10Base-5	11,8	46.5	169.5	0.0866	500
10Base-2	11.8	46,5	169,5	0,1026	185
10Base-T	15,3	42.0	165.0	0,113	100
10Base-FB	-	24.0	-	0.1	2000
10Base-FL	12.3	33.5	156.5	0.1	2000
FOIRL	7,8	29,0	152,0	0,1	1000
AUI (>2 м)	0	0	0	0,1026	2+48

Таблица 15 – Уменьшение межкадрового интервала повторителями

Тип сегмента	Передающий сегмент. bt	Промежуточный сегмент, bt
10Base-5 или 10Base-2	16	11
10Base-FB	-	2
10Base-FL	10,5	8
10Base-T	10.5	8

Чтобы сеть Ethernet, состоящая из сегментов различной физической природы, работала корректно, необходимо выполнение четырех основных условий:

– количество станций в сети – не более 1024;

– максимальная длина каждого физического сегмента – не более величины, определенной в соответствующем стандарте физического уровня;

– время двойного оборота сигнала (Path Delay Value, PDV) между двумя самыми удаленными друг от друга станциями сети – не более 575 битовых ин-

тервала;

– сокращение межкадрового интервала (Path Variability Value, PVV) при прохождении последовательности кадров через все повторители – не больше, чем 49 битовых интервала (так как при отправке кадров конечные узлы обеспечивают начальное межкадровое расстояние в 96 битовых интервала, то после прохождения повторителя оно должно быть не меньше, чем 96 - 49 = 47 битовых интервала).

Соблюдение этих требований обеспечивает корректность работы сети даже в случаях, когда нарушаются простые правила конфигурирования, определяющие максимальное количество повторителей и общую длину сети в 2500 м.

Методика расчета времени двойного оборота и уменьшения межкадрового интервала.

Для упрощения расчетов обычно используются справочные данные IEEE, содержащие значения задержек распространения сигналов в повторителях, приемопередатчиках и различных физических средах. Битовый интервал обозначен как bt.

Задержки, вносимые повторителем, состоят из задержки входного трансивера, задержки блока повторения и задержки выходного трансивера. В таблице все эти задержки представлены одной величиной, названной базой сегмента. В таблице даются удвоенные величины задержек для каждого типа кабеля [6].

В таблице используются также такие понятия, как левый сегмент, правый сегмент и промежуточный сегмент. Поясним эти термины на примере сети, приведенной на рисунке 19. Левым сегментом называется сегмент, в котором начинается путь сигнала от выхода передатчика конечного узла. На рисунке 19 это сегмент 1. Затем сигнал проходит через промежуточные сегменты 2-5 и доходит до приемника наиболее удаленного узла наиболее удаленного сегмента 6, который называется правым. Именно здесь в худшем случае происходит столкновение кадров и возникает коллизия.

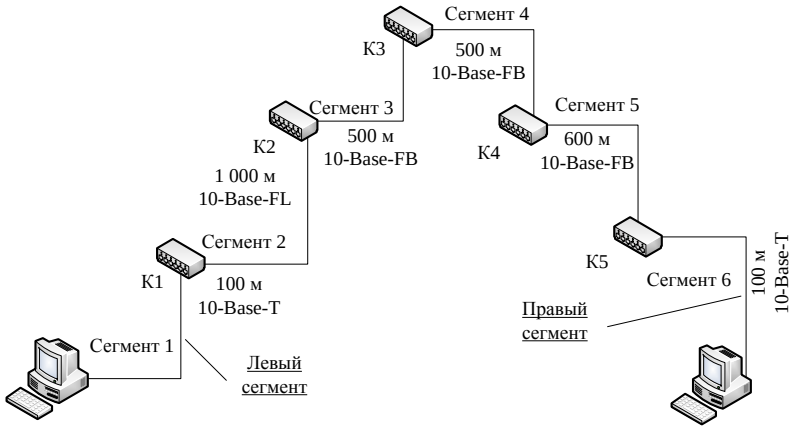


Рисунок 19 – Пример сети Ethernet, состоящей из сегментов различных физических стандартов

С каждым сегментом связана постоянная задержка, названная базой, которая зависит только от типа сегмента и от положения сегмента на пути сигнала (левый, промежуточный или правый). База правого сегмента, в котором возникает коллизия, намного превышает базу левого и промежуточных сегментов [6].

Кроме этого, с каждым сегментом связана задержка распространения сигнала вдоль кабеля сегмента, которая зависит от длины сегмента и вычисляется путем умножения времени распространения сигнала по одному метру кабеля (в битовых интервалах) на длину кабеля в метрах.

Расчет PDV заключается в вычислении задержек, вносимых каждым отрезком кабеля (приведенная в таблице задержка сигнала на 1 м кабеля умножается на длину сегмента), а затем суммировании этих задержек с базами левого, промежуточных и правого сегментов. Общее значение PDV не должно превышать 575.

Так как левый и правый сегменты имеют разные величины базовой задержки, то в случае различных типов сегментов на удаленных краях сети необходимо выполнить расчеты дважды: один раз принять в качестве левого сегмента сегмент одного типа, а во второй – сегмент другого типа. Результатом можно считать максимальное значение PDV.

Чтобы признать конфигурацию сети корректной, нужно рассчитать также уменьшение межкадрового интервала повторителями, то есть величину PVV.

Для расчета PVV также можно воспользоваться значениями максимальных

величин уменьшения межкадрового интервала при прохождении повторителей различных физических сред, приведены в таблице 15 [6].

Пример расчета конфигурации сети.

Приведенная на рисунке 19 сеть в соответствии с правилом «4 хабов» не является корректной – в сети между узлами сегментов 1 и 6 имеются 5 хабов, хотя не все сегменты являются сегментами 10Base-FB. Кроме того, общая длина сети равна 2800 м, что нарушает правило 2500 м. Рассчитаем значение PDV.

Левый сегмент 1: 15,3 (база) + 100·0,113 = 26,6.

Промежуточный сегмент 2: 33,5 + 1000·0,1 = 133,5.

Промежуточный сегмент 3: 24 + 500·0,1 = 74,0.

Промежуточный сегмент 4: 24 + 500·0,1 = 74,0.

Промежуточный сегмент 5: 24 + 600·0,1 = 84,0.

Правый сегмент 6: 165 + 100·0,113 = 176,3.

Сумма всех составляющих дает значение PDV, равное 568,4.

Так как значение PDV меньше максимально допустимой величины 575, то эта сеть проходит по критерию времени двойного оборота сигнала несмотря на то, что ее общая длина превышает 2500 м, а количество повторителей больше 4.

Рассчитаем значение PVV.

Левый сегмент 1 10Base-T: сокращение в 10,5 bt.

Промежуточный сегмент 2 10Base-FL: 8.

Промежуточный сегмент 3 10Base-FB: 2.

Промежуточный сегмент 4 10Base-FB: 2.

Промежуточный сегмент 5 10Base-FB: 2.

Сумма этих величин дает значение PVV, равное 24,5, что меньше предельного значения в 49 битовых интервала.

В результате сеть соответствует стандартам Ethernet по всем параметрам.

Задание к проведению лабораторной работы.

1. Ознакомиться с теоретическим материалом.

Произвести оценку конфигурации сети в соответствии с вариантом:

− по физическим ограничениям: на длину сегмента, на длину сети, правило «4

хаба» («5 хабов» для 10Base-FB);

– по времени двойного оборота сигнала в сети;

– по уменьшению межкадрового интервала.

3. По результатам расчетов сделать вывод о корректности конфигурации сети Ethernet.

4. По результатам работы оформить отчет.

Задание к проведению лабораторной работы.

Исходные данные для заданий к лабораторным работам показаны в таблицах 16 - 28.

Таблица 16 - Задание к проведению лабораторной работы для варианта № 1

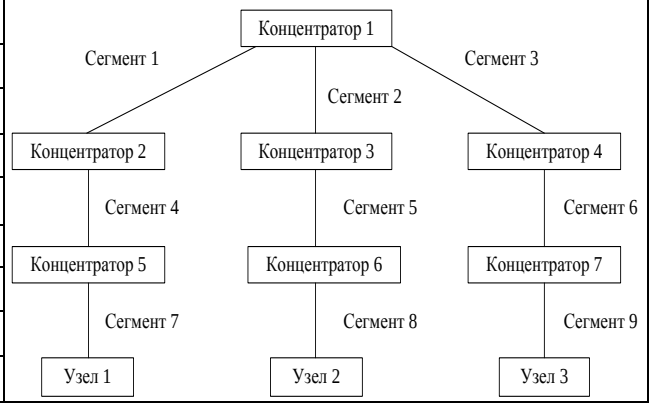
	10Base-FB	10Base-FL	10Base-T	Длина, м	Топология сети
Сегмент 1	+			500	
Сегмент 2	+			300	
Сегмент 3	+			400	
Сегмент 4		+		1000	
Сегмент 5		+		300	
Сегмент 6		+		400	
Сегмент 7			+	100	
Сегмент 8			+	50	
Сегмент 9			+	100	

Таблица 17 - Задание к проведению лабораторной работы для варианта № 2

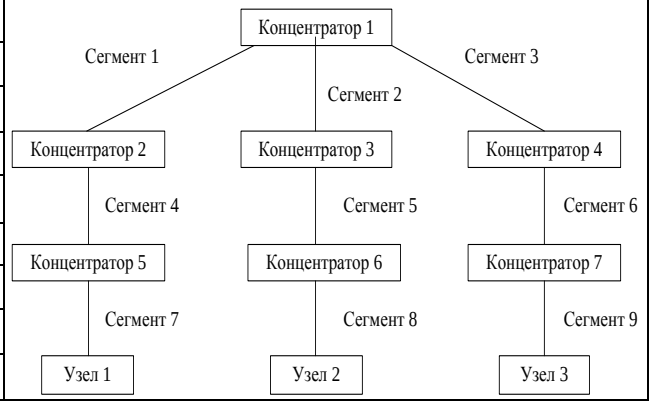
	10Base-FB	10Base-FL	10Base-T	Длина, м	Топология сети
Сегмент 1		+		700	
Сегмент 2	+			400	
Сегмент 3	+			400	
Сегмент 4		+		700	
Сегмент 5		+		200	
Сегмент 6	+			500	
Сегмент 7			+	80	
Сегмент 8			+	100	
Сегмент 9			+	80	

Таблица 18 - Задание к проведению лабораторной работы для варианта № 3

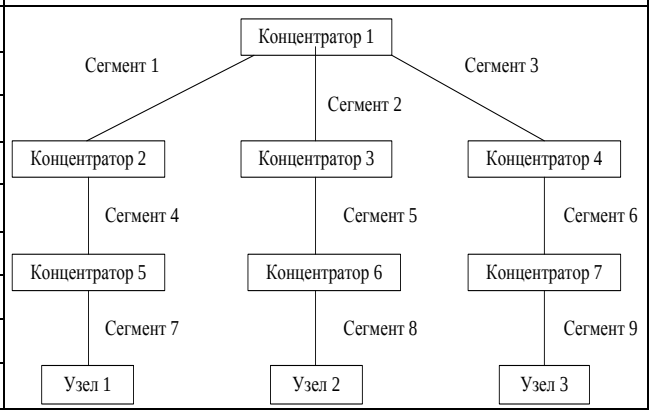
	10Base-FB	10Base-FL	10Base-T	Длина, м	Топология сети
Сегмент 1	+			1000	
Сегмент 2		+		200	
Сегмент 3		+		200	
Сегмент 4		+		400	
Сегмент 5	+			300	
Сегмент 6		+		200	
Сегмент 7			+	100	
Сегмент 8			+	100	
Сегмент 9			+	40	

Таблица 19 - Задание к проведению лабораторной работы для варианта № 4

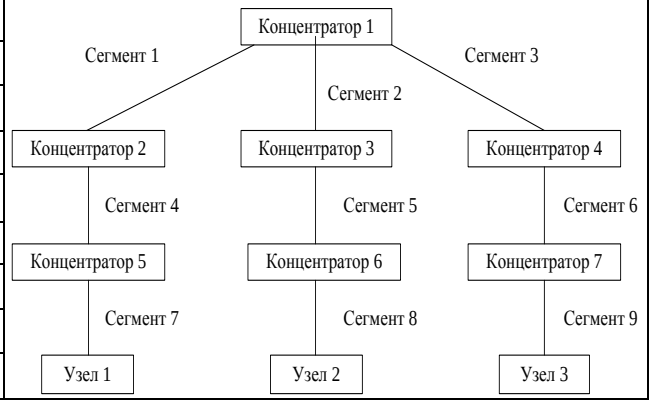
	10Base-FB	10Base-FL	10Base-T	Длина, м	Топология сети
Сегмент 1		+		600	
Сегмент 2		+		400	
Сегмент 3		+		200	
Сегмент 4	+			800	
Сегмент 5	+			500	
Сегмент 6	+			800	
Сегмент 7			+	50	
Сегмент 8			+	100	
Сегмент 9			+	50	

Таблица 20 - Задание к проведению лабораторной работы для варианта № 5

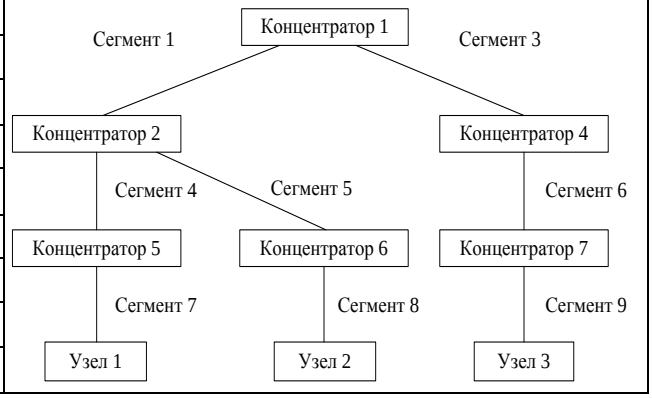
	10Base-FB	10Base-FL	10Base-T	Длина, м	Топология сети
Сегмент 1	+			400	
Сегмент 3	+			500	
Сегмент 4		+		1100	
Сегмент 5		+		1100	
Сегмент 6		+		600	
Сегмент 7			+	100	
Сегмент 8			+	100	
Сегмент 9			+	100	
Сегмент 9			+	50	

Таблица 21 - Задание к проведению лабораторной работы для варианта № 6

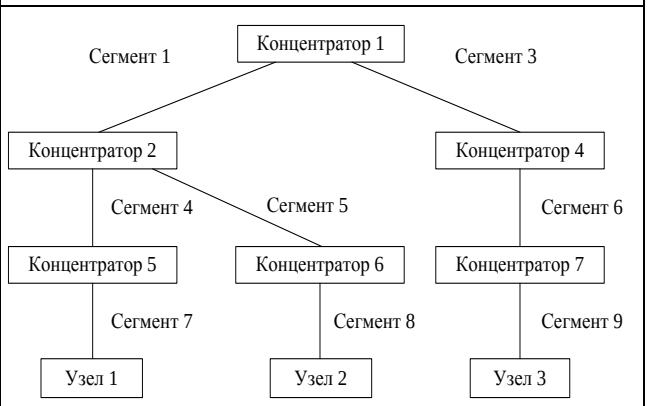
	10Base-FB	10Base-FL	10Base-T	Длина, м	Топология сети
Сегмент 1	+			500	
Сегмент 3		+		500	
Сегмент 4	+			1000	
Сегмент 5	+			1000	
Сегмент 6		+		500	
Сегмент 7			+	80	
Сегмент 8			+	80	
Сегмент 9			+	100	
Сегмент 9			+	50	

Таблица 22 - Задание к проведению лабораторной работы для варианта № 7

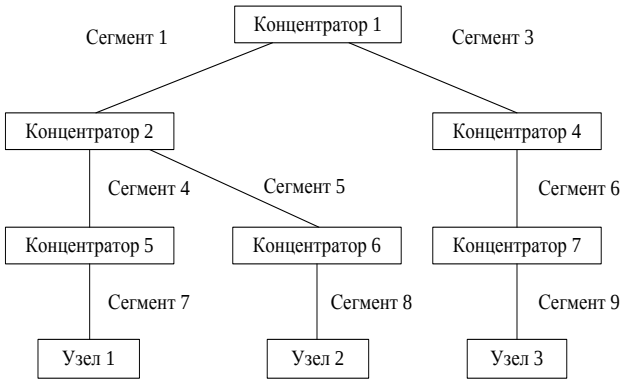
	10Base-FB	10Base-FL	10Base-T	Длина, м	Топология сети
Сегмент 1		+		1000	
Сегмент 3	+			1000	
Сегмент 4		+		600	
Сегмент 5		+		600	
Сегмент 6	+			400	
Сегмент 7			+	60	
Сегмент 8			+	60	
Сегмент 9			+	90	
Сегмент 9			+	50	

Таблица 23 - Задание к проведению лабораторной работы для варианта № 8

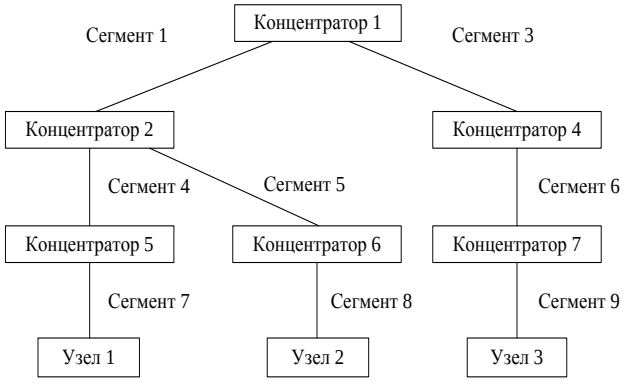
	10Base-FB	10Base-FL	10Base-T	Длина, м	Топология сети
Сегмент 1		+		900	
Сегмент 3		+		900	
Сегмент 4	+			700	
Сегмент 5	+			700	
Сегмент 6	+			500	
Сегмент 7			+	70	
Сегмент 8			+	70	
Сегмент 9			+	100	
Сегмент 9			+	50	

Таблица 24 - Задание к проведению лабораторной работы для варианта № 9

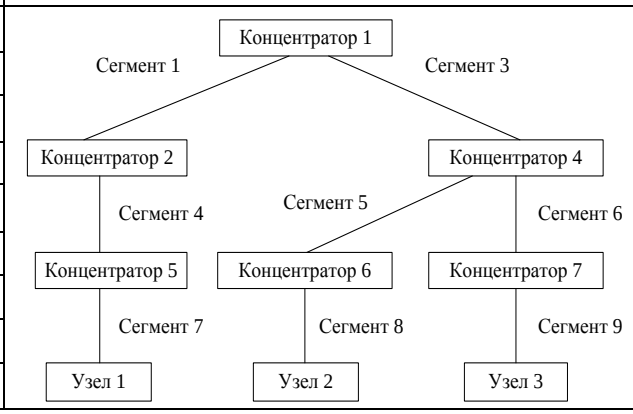
	10Base-FB	10Base-FL	10Base-T	Длина, м	Топология сети
Сегмент 1		+		900	
Сегмент 3		+		900	
Сегмент 4	+			700	
Сегмент 5	+			700	
Сегмент 6	+			500	
Сегмент 7			+	70	
Сегмент 8			+	70	
Сегмент 9			+	100	
Сегмент 9			+	50	

Таблица 25 - Задание к проведению лабораторной работы для варианта № 9

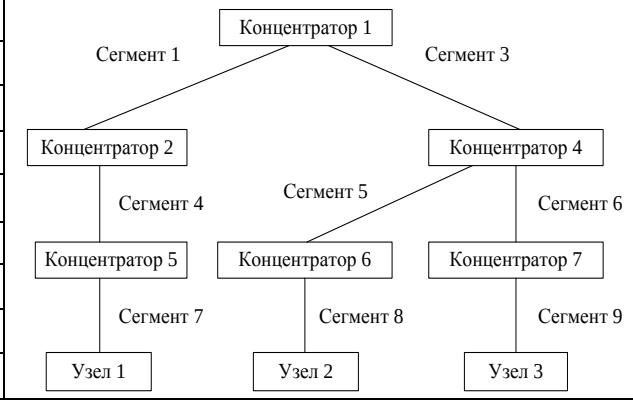
	10Base-FB	10Base-FL	10Base-T	Длина, м	Топология сети
Сегмент 1	+			400	
Сегмент 3	+			500	
Сегмент 4		+		1100	
Сегмент 5		+		1100	
Сегмент 6		+		600	
Сегмент 7			+	100	
Сегмент 8			+	100	
Сегмент 9			+	100	
Сегмент 9			+	50	

Таблица 26 - Задание к проведению лабораторной работы для варианта № 10

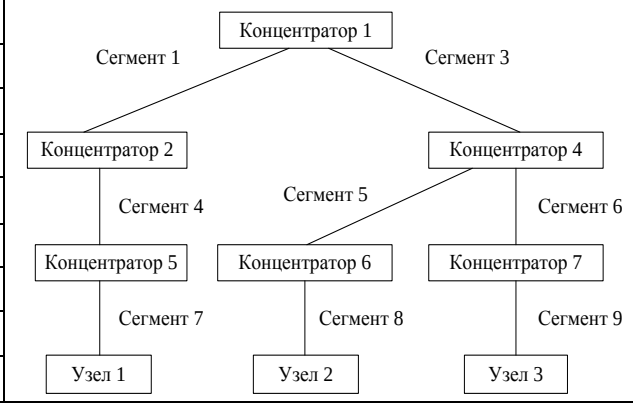
	10Base-FB	10Base-FL	10Base-T	Длина, м	Топология сети
Сегмент 1	+			500	
Сегмент 3		+		500	
Сегмент 4	+			1000	
Сегмент 5	+			1000	
Сегмент 6		+		500	
Сегмент 7			+	80	
Сегмент 8			+	80	
Сегмент 9			+	100	
Сегмент 9			+	50	

Таблица 27 - Задание к проведению лабораторной работы для варианта № 11

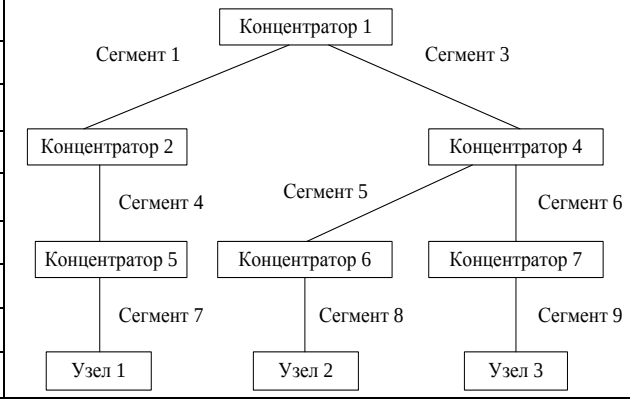
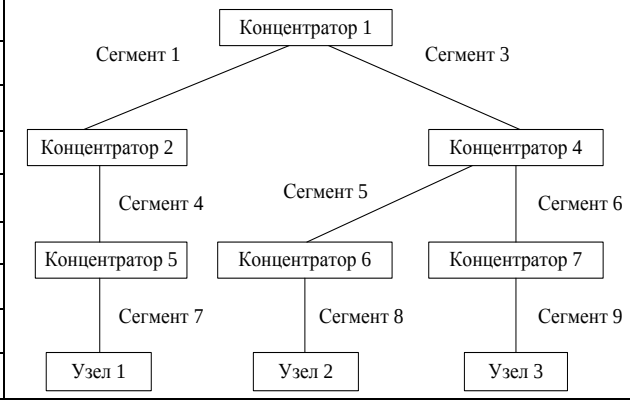
	10Base-FB	10Base-FL	10Base-T	Длина, м	Топология сети
Сегмент 1	+			500	
Сегмент 1		+		1000	
Сегмент 3	+			1000	
Сегмент 4		+		600	
Сегмент 5		+		600	
Сегмент 6	+			400	
Сегмент 7			+	60	
Сегмент 8			+	60	
Сегмент 9			+	90	

Таблица 28 - Задание к проведению лабораторной работы для варианта № 12

	10Base-FB	10Base-FL	10Base-T	Длина, м	Топология сети
Сегмент 1		+		600	
Сегмент 3		+		600	
Сегмент 4	+			900	
Сегмент 5	+			1000	
Сегмент 6	+			500	
Сегмент 7			+	70	
Сегмент 8			+	80	
Сегмент 9			+	90	
Сегмент 9			+	90	

Содержание отчета по лабораторной работе.

1. Название и цель работы.

2. Исходные данные.

3. Представит логическую схему сети.

4. Расчеты указанных параметров.

5. Выводы по выполненной работе.

6. Список использованных источников.

Контрольные вопросы.

1. Поясните механизм доступа к разделяемой среде в технологии Ethernet.

2. В каких случаях возможна оценка корректности конфигурации по физическим ограничениям?

3. Сформулируйте условие надежного распознавания коллизий.

4. С какой целью вводится ограничение на уменьшение межкадрового интервала?

5. В каком случае и почему для самого длинного пути проводятся два расчета?

10 Лабораторная работа № 10. Логическое и физическое проектирование сети

Цель работы. Используя программное обеспечение NetWizard.exe провести логическое и физическое проектирование ЛВС. Провести расчет спроектированной ЛВС.

Задание на выполнение лабораторной работы.

В соответствии с заданным вариантом, разработать структурную схему ЛВС Ethernet отдела предприятия, включающую общий сервер при следующих данных:

- M - количество групп;
- N - количество компьютеров в группе;
- L - расстояние между компьютерами в группе;
- S - расстояние между группами;
- k - средняя интенсивность трафика, генерируемого одним компьютером.

Соотношение трафиков внутригруппового и меж группового 50 % на 50 %.

Примечание. Работа выполняется по вариантам согласно таблице.

Этапы выполнения лабораторной работы.

1. Логическое проектирование - логическая структуризация и определение логической схемы ЛВС.

2. Физическое проектирование. Выбор физических спецификаций базовых технологий, обеспечивающих построение ЛВС в соответствии с заданными максимальными расстояниями. Построение структурной схемы ЛВС. Определение типов и требуемых характеристик структурообразующего оборудования.

3. Проверочный расчет спроектированной ЛВС на корректность: Производится расчет времени двойного оборота PDV (Path Delay Value) для доменов коллизий спроектированной сети.

Этап №1. Логическое проектирование.

1. Определение трафика одного компьютера:

$$C_i = k * C, \tag{5}$$

где С - принять равным 10 (100) Мбит/с;

k - коэффициент из варианта задания.

2. Определение суммарного трафика неструктурированной ЛВС:

$$C_{\Sigma}^{H} = N * M * C_i, \tag{6}$$

где N, M – величины, взятые из задания;

C_i – трафик, одного компьютера.

3. Определение коэффициента нагрузки неструктурированной ЛВС:

$$\rho_H = \frac{C_{\Sigma}^{н}}{C_{макс}}, \tag{7}$$

где $C_{мах}$ – пропускная способность выбранной технологии; первоначально выбрать 10 (100) Мбит/с.

4. Проверка выполнения условия допустимой нагрузки ЛВС (домена коллизий):

$$\rho_н \le \rho_{Ethernet} = 0{,}35, \tag{8}$$

где $\rho_н$ - коэффициент нагрузки неструктурированной сети, или коэффициент нагрузки домена коллизий $\rho_{Д.К}$.

Если условие не выполняется необходимо произвести логическую структуризацию проектируемой ЛВС:

– разбить сеть на домены коллизий (логические сегменты) по числу групп и количеству компьютеров согласно варианту задания (порядок разбиения смотри в помощи, «F1»);

– определить коэффициент нагрузки домена коллизий:

$$\rho_{Д.К} = \frac{N_{л.с} * C_i}{C_{макс}}, \quad (9)$$

где $N_{л.с}$ - количество компьютеров в логическом сегменте.

Проверить выполнение условия п.4. Если условие не выполняется, повторить пункт 5, разбив каждый домен коллизий на два.

Повторять пункт 1.6 до тех пор пока не будет выполнено условие п.4.

5. Определение межгруппового трафика

$$C^{H}_{меж.гр} = 0{,}5C^{H}{}_{\Sigma}, \quad (10)$$

где C^{H}_{Σ} - суммарный трафик неструктурированной ЛВС.

6. Определение коэффициента нагрузки ЛВС по межгрупповому обмену:

$$\rho^{н}_{меж.гр} = \frac{C^{H}{}_{меж.гр}}{C_{макс}}, \quad (11)$$

где $C^{H}{}_{меж.гр}$ - межгрупповой трафик;

$C_{макс}$ - выбранная максимальная производительность базовой технологии Ethernet.

Для достижения нормальной нагрузки межгруппового соединения, последовательно увеличивайте производительность базовой технологии до тех пор пока не будет выполнятся условие 4.

7. Произвести контрольную проверку при помощи кнопки «Проверить».

Этап № 2. Физическое проектирование ЛВС.

1. Подобрать тип кабеля для соединения компьютеров. Выбор кабельной системы производится исходя из параметров физического уровня для стандартов Ethernet, Fast Ethernet, Gigabit Ethernet.

2. Подобрать коммутационное оборудование. Выбор коммутационного оборудования производится исходя из списка заложенного в программу. Выберите оптимально подходящее оборудование.

3. Собрать схему, согласно проведенному расчету в окне «Логическое проектирование ЛВС». При соединении кабелем, не забывайте указывать его длину.

4. Произвести проверочный расчет спроектированной ЛВС на корректность: Производится расчет времени двойного оборота PDV (Path Delay Value) для доменов коллизий спроектированной сети.

$$PDV = 2 * \left[\left(\frac{bt}{M} * L \right) + \left(bt_r + bt_t \right) + \left(\frac{bt}{M} * L \right) ..... \right], \quad (12)$$

где $\frac{bt}{M}$ - задержка в кабеле на 1 метр;

L - длина кабеля в метрах;

bt_r - задержка на входном порту концентратора;

bt_t - задержка на выходном порту концентратора.

5. Проверка допустимых размеров сети.

$$PDV < 512bt. \quad (13)$$

6. Если условие не выполняется необходимо произвести физическую структуризацию проектируемой ЛВС.

Пример выполнения работы.

Этап №1. Логическое проектирование сети.

Таблица 29 – Задание для пятого варианта

Вариант	M	N	S	L	K	Количество рабочих станций в сети
5	3	7	400,00	3,00	0.070	21

Определение трафика одного компьютера:

$C_i = k*C = 0.070*10(\text{Мбит/с}) = 0{,}7$ Мб/с.

Определение суммарного трафика неструктурированной ЛВС:

$C^H_\Sigma = N*M*C_i = 7*3*0.7 = 14.70$ Мб/с.

Определение коэффициента нагрузки неструктурированной ЛВС:

$$\rho_h <= \rho_{Ethernet} = 0.35$$

$$\rho_H = C_\Sigma^H / C_{max} = 14.70/10(\text{Мбит/с}) = 1.47$$

Коэффициент нагрузки превышает допустимое значение. Для обеспечения работоспособности сети необходимо произвести логическую структуризацию проектируемой ЛВС путем увеличения доменов коллизий (логических сегментов) и

уменьшения количества рабочих станций в каждой группе. Оптимальный вариант получается при установке пяти групп. Первая группа содержит пять рабочих станций, остальные - по четыре.

Определение суммарного трафика неструктурированной ЛВС:

$$C^{H}_{\Sigma}=N*M*C_i=5*1*0.7=3.50 \text{ МБ/с}$$

Определение коэффициента нагрузки неструктурированной ЛВС:

$$\rho_h <= \rho_{Ethernet}=0.35$$

$$\rho_H= C_{\Sigma}^{H}/C^{max}=3.50/10(\text{Мбит/с})=0.35$$

Определение коэффициента нагрузки домена коллизий:

$$\rho_{д.к.}=N_{л.с.}*C_i/C_{max}=5*0.7/10=0.35$$

Определение межгруппового трафика:

$$C^{H}_{Меж.гр.}=0.5\ C_{\Sigma}^{H}=0.5*3.50=1.75\text{МБ/с}$$

Определение коэффициента нагрузки ЛВС по межгрупповому обмену:

$$\rho^{H}_{Меж.гр.}=C^{H}_{Меж.гр.}/C_{max}=1.75/10(\text{Мбит/с})=0.18$$

Междоменный трафик превышает допустимое значение. Необходимо увеличить пропускную способность сети до 100Мбит/с.

$$\rho^{H}_{Меж.гр.}=C^{H}_{Меж.гр.}/C_{max}=1.75/100(\text{Мбит/с})=0.02$$

После проведения логической структуризации и выбора необходимых параметров, данная сеть является представлением рабочей и полнофункциональной сети.

Этап № 2. Физическое проектирование сети.

Как уже упоминалось, оптимальный вариант получается при установке пяти групп, первая из которых содержит пять рабочих станций, остальные - по четыре.

Для объединения компьютеров внутри каждой группы использовался коммутатор, именуемый Compex SwitchHub 10/100 с восемью UTP-портами.

Центральным звеном сети является коммутатор 3COM OfficeConnect с пятью UTP-портами.

Кабель – неэкранированная пара 5-ой категории.

В данном задании расстояние между группами должно составлять 400м. По стандарту для кабеля 5-й категории, длина одного сегмента не должна превышать

100м. Для увеличения расстояния используются повторители. Число активных устройств на всей протяженности канала не должно превышать четырех.

Чтобы достичь расстояния в 400 метров, между группой и центральным коммутатором используется повторитель типа Compex SwitchHub 10/100.

Таким образом, расстояние между группами делится на четыре сегмента по сто метров каждый.

Комплексная проверка модели сети указала на ее корректность.

Расчет ЛВС на корректность.

Для проверки корректности сети производится расчет времени двойного оборота PDV для доменов коллизий спроектированной сети.

$$PDV=2*[(bt/M)*L+(bt_r+bt_t)+(bt/M)*L \ldots]. \quad (14)$$

где bt/M – задержка в кабеле на 1 метр;

L – длина кабеля в метрах;

bt_r - задержка на входном порту концентратора;

bt_t - задержка на выходном порту концентратора.

Допустимый параметр сети – PDV<575.

В данном случае:

– bt/M=0.113 (для типа сегмента 10Base-T);

– L=608 (600м между группами и по 4 метра от компьютера группы до ближайшего коммутатора).

Значения bt_r и bt_t согласно брать максимально приближенными к нулю (т.е. не учитывать).

$$PDV=2*0,113*608=137,408 < 575$$

Данная модель удовлетворяет условиям PDV и является рабочей.

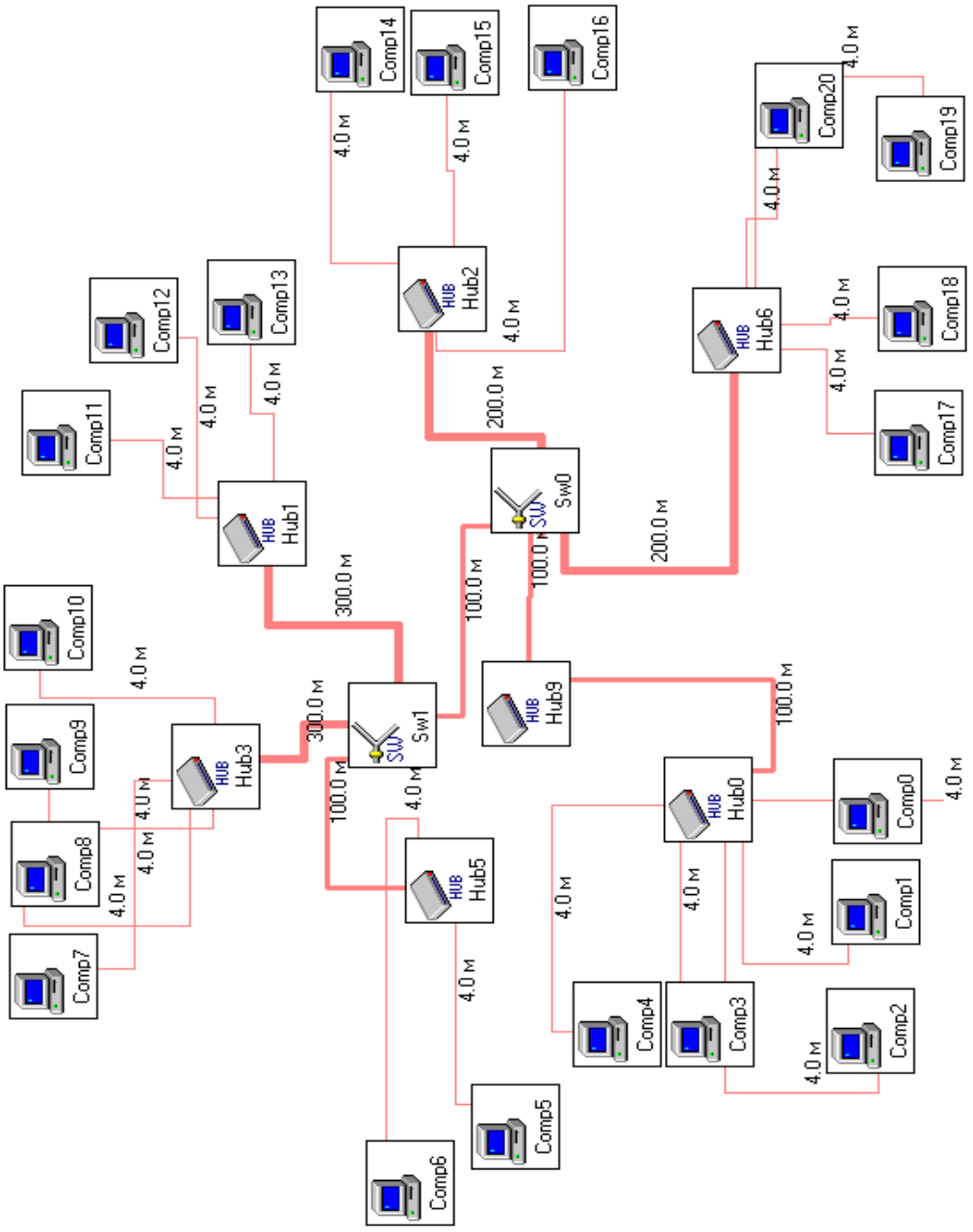


Рисунок 20 – Логическая схема разработанной ЛВС

Содержание отчета по лабораторной работе.

1. Название и цель работы.
2. Исходные данные.
3. Представит логическую схему сети.
4. Расчеты параметров ЛВС.
5. Обосновать принятые инженерные решения.
6. Выводы по выполненной работе.
7. Список использованных источников.

11 Лабораторная работа № 11. Структуризация внутренней сети с помощью маски постоянной длины на примере IP-адреса класса B

Цель работы. Научиться организовывать заданное число подсетей с учетом вырожденной сети, используя маску подсети постоянной длины.

Решить задачу в соответствии с вариантом, используя изложенную ниже методику.

1. В Основном меню курсором выбрать закладку **"Задание № 1"**. В появившейся таблице исходных данных выбрать строку с соответствующим номером варианта, дважды щелкнув мышью по строке таблицы. Например – вариант - 1, число подсетей - 3. Это значит, что необходимо будет организовать 3 подсети + 1 вырожденная (соединяющая маршрутизаторы **M1** и **M2**).

2. В появившейся форме ввод во всех окнах ввода десятичной записи заканчивается щелчком "мыши" в соответствующем окна двоичной записи и наоборот. Ввести в окне **"Номер внутренней сети"** любой номер сети из диапазона IP-адресов класса B, например - 128.0.0.0. Щелкнуть в окне двоичной записи номера сети - в данном окне отобразится двоичная запись номера сети. Для получения информации о классах IP - адресов щелкнуть правой кнопкой "мыши" на окне ввода, в появившемся меню выбрать **"Класс B"**. Появится диапазон номеров сетей данного класса (Internet - адрес) и диапазон адресов, не обрабатываемых Internet - маршрутизаторами (локальных), которые нельзя использовать при выполнении задания.

3. Ввести в окне **"Маска подсети"** десятичное значение стандартной маски класса B. Информацию о стандартных масках можно получить, щелкнув правой кнопкой мыши на окне десятичной записи **"Маска подсети".**

Определить маску, необходимую для образования заданного количества подсетей (в нашем примере 4-х).

Для этого:

– определить количество старших разрядов в поле номера узла, подлежащих

маскированию, исходя из заданного количества подсетей (в этих разрядах маски должны быть установлены «1»);

– расписать в отчете в двоичном исчислении последовательный перебор номеров подсетей, подлежащих образованию, учитывая, что номер не может содержать все «0» или все «1».

В данном примере - это сочетание 001 010 011 100 101 110.

В нашем примере - это 3 единицы в старших разрядах поля номера узла, и маска должна иметь все единицы в поле номера сети и 3 единицы в старших разрядах поля номера узла.

Теоретическая справка.

Варианты значения масок подсетей, применяемых в данной работе:

- 255.255.0.0 11111111.11111111.00000000.00000000;
- 255.255.128.0 11111111.11111111.10000000.00000000;
- 255.255.192.0 11111111.11111111.11000000.00000000;
- 255.255.224.0 11111111.11111111.11100000.00000000;
- 255.255.240.0 11111111.11111111.11110000.00000000;
- 255.255.248.0 11111111.11111111.11111000.00000000;
- 255.255.252.0 11111111.11111111.11111100.00000000;
- 255.255.254.0 11111111.11111111.11111110.00000000;
- 255.255.255.0 11111111.11111111.11111111.00000000.

В окне ввода маски в двоичном изображении отделить в старшем октете слева определенное количество разрядов (в примере - 3 разряда) и установить в выделенных разрядах «1». Для этого:

– щелкнуть «мышью» в окне двоичного ввода маски;

– используя клавиши управления курсором и клавишу **«BackSpace»** удалить «0» в соответствующих разрядах;

– используя клавишу «1» цифровой клавиатуры установить в разрядах единицы.

Щелкнуть «мышью» в окне ввода десятичной записи маски.

В окне **«Маска подсети»** появится десятичная запись маски. В нашем

примере это - 255.255.224.0.

Заполнение таблицы маршрутизации маршрутизатора **М2.**

1. Щелкнуть по кнопке **«Запись»**. В окне **«Номера подсетей»** будут отображены номера организуемых подсетей кроме номера подсети, соединяющей маршрутизаторы (в нашем примере это - 128.1.64.0).

2. Щелкнуть по кнопке **«Таблица маршрутизации»**. Раскроется таблица маршрутизации. Щелкая «мышью» на соответствующей записи в окне **«Номера подсетей»**, а затем в пустой ячейке таблицы маршрутизации и, таким образом, подключить подсети к портам маршрутизатора (IP - адреса портов маршрутизатора назначаются программой автоматически). По мере заполнения таблицы маршрутизации записи в окне **«Номера подсетей»** удаляются. При ошибочном вводе значений в таблицу маршрутизации дважды щелкнуть «мышью» в редактируемой строке таблицы - строка в таблице пропадет и переместится на последнюю строчку окна **«Номера подсетей».** При записи значений в таблицу маршрутизации поверх уже существующей строки последняя также удаляется из таблицы и добавляется в окно **«Номера подсетей».** Для восстановления содержимого окна **«Номера подсетей»** щелкнуть по кнопке «**Запись».** Подключение подсетей заканчивается щелчком по кнопке **«Таблица маршрутизации».** Для удобства вырожденная сеть, соединяющая маршрутизаторы **М1** и **М2** всегда подключена к 1-му порту **М2** и имеет номер X.X.64.0, где X.X.0.0 - номер сети (в нашем примере - 128.1.0.0). Порт 2 маршрутизатора **М1** имеет IP-адрес X.X.64.2, порт 1 **М2** - X.X.64.1. Остальным портам **М2** автоматически будут присвоены IP-адреса Y.Y.Y.1, где Y.Y.Y.0 - номер подключаемой к порту подсети (**пример** - подсеть 128.1.32.0 подключается к порту 2, IP-адрес порта 2 - 128.1.32.1).

Пошаговая отработка алгоритма работы маршрутизатора **М2** при продвижении произвольного IP-пакета.

1. В окне «Внешний IP – адрес» записать любой IP - адрес или выбрать из предложенных в выпадающем списке.

Примечание. При вводе IP - адресов учесть, что в программе недопустимы следующие их значения:

– X.X.X.255;

– X.X.X.0;

– X.X.X.1 - (адреса портов маршрутизатора M2, X.X.X - номер подсети).

IP - адреса вырожденной сети:

– X.X.64.1;

– X.X.64.2.

Запрещены адреса в диапазоне - 172.16.0.0 - 172.31.255.255 (локальные адреса, не обрабатываемые Internet - маршрутизаторами).

2. Мысленно наложить маску на IP - адрес, ориентируясь по записям в окнах двоичного представления адреса и маски. Замаскированные разряды 3-го октета покажут номер подсети.

В описываемом примере введем IP - адрес 128.1.192.2.

Маска подсети 11111111.11111111.11100000.00000000.

IP - адрес 10000000.00000001.11000000.00000010.

3. В выпадающем окне двоичной записи номеров подсетей выбрать строку с ранее определенным адресом подсети - 10000000.00000001.11000000.00000000.

В окне **«Номер подсети»** появится десятичная запись номера подсети: 128.1.192.0.

Определить, имеется ли данная подсеть во внутренней сети, сравнив с номерами подсетей на мнемосхеме. Если внешний IP-адрес адресует пакет в одну из образованных подсетей, то определенный номер подсети, сформировавшийся в окне **«Номер подсети»** будет совпадать с одним из адресов, подключенных к порту маршрутизатора **M2**. Щелкнуть по кнопке **«Test»** для удостоверения в правильности определения номера подсети.

4. Вычесть значения полей окна двоичного представления номера подсети из соответствующих значений окна двоичного представления IP-адреса - полученное значение будет номером узла, который необходимо записать в окне двоичного представления номера узла. Щелкнуть по кнопке **«Test».**

Пример 1:

– IP - адрес 128.1.192.2;

– номер подсети 128.1.192.0;

– номер узла 0.2.

Пример 2:

– IP - адрес 128.1.197.2;

– Номер подсети 128.1.192.0;

– Номер узла 5.2.

5. Определить широковещательный адрес подсети – «**BROADCAST»**.

Адрес позволяет обращаться ко всем узлам подсети и содержит '1' во всех разрядах номера узла (двоичное представление).

Для определения адреса необходимо:

– используя клавиши управления курсором, **«BackSpace»**, цифровую клавиатуру, записываем номер определенной подсети в окне двоичной записи широковещательного адреса - 10000000.00000001.1100000.00000000;

– используя клавиши управления курсором, **«BackSpace»**, цифровую клавиатуру, записываем единицы в немаскируемых разрядах поля номера узла, получаем - 10000000.00000001.11011111.11111111;

– после щелчка "мыши" вне окна ввода в окне **«BROADCAST»** появится запись широковещательного адреса подсети - 128.1.223.255.

Проверка результатов и имитация продвижения IP - пакета в сети.

1. Щелкнуть по кнопке **«Test».** На элементах, составляющих изображение структурной схемы сети появятся кнопки-индикаторы в случае успешного продвижения пакета зеленого, иначе - красного цветов. Щелкнув по кнопке индикатору можно получить краткое сообщение о результате прохождения IP - пакета через данный элемент сети.

2. Если номер подсети в которую направляется IP-пакет определен правильно, то на схеме напротив соответствующего номера подсети появится сообщение **«CONNECT».** Кроме того, о правильности выполнения этапов задания выдаются сообщение в информационном окне.

В случае правильного выполнения задания повторить подпункты разделов 5 и 6 не менее чем для 3 IP - адресов.

Ввести IP-адрес не принадлежащий ни одной из подсетей. Для этого в окне **«Внешний IP-адрес»** установить произвольный IP-адрес, отличный от адреса отображенного в окне **«Номер внутренней сети»** в нашем примере это может быть адрес 190.1.192.2. Убедиться, что он не подходит ни для одной из подсетей.

Возврат в основное меню производится щелчком «мыши» по кнопке **«Выход».**

Содержание отчета по лабораторной работе.

1. Название и цель работы.
2. Определить маску, необходимую для образования заданного количества подсетей.
3. Заполнить таблицу маршрутизации маршрутизатора.
4. Представить пошаговую отработку алгоритма работы маршрутизатора при продвижении произвольного IP-пакета.
5. Провести проверку результатов продвижения IP - пакета в сети.
6. Эскизно представить пути продвижения IP - пакета в сети.
7. Представить в двоичном исчислении последовательный перебор номеров подсетей.
8. Обосновать принятые инженерные решения.
9. Выводы по выполненной работе.
10. Список использованных источников.

12 Лабораторная работа № 12. Определение адресов продвижения IP пакета в гетеродинной сети

Цель работы. Изучение функционирование служб разрешения адресов и продвижение IP пакета в гетерогенной сети.

Порядок выполнения работы.

В соответствии с вариантом задания, определить адреса продвижения IP пакета в гетерогенной сети, используя ниже приведенный пример.

Дано. IP адрес (один адрес).

Разрешение IP адресов, используя просмотр файла hosts.

Разрешение MAC адресов, используя ARP запрос.

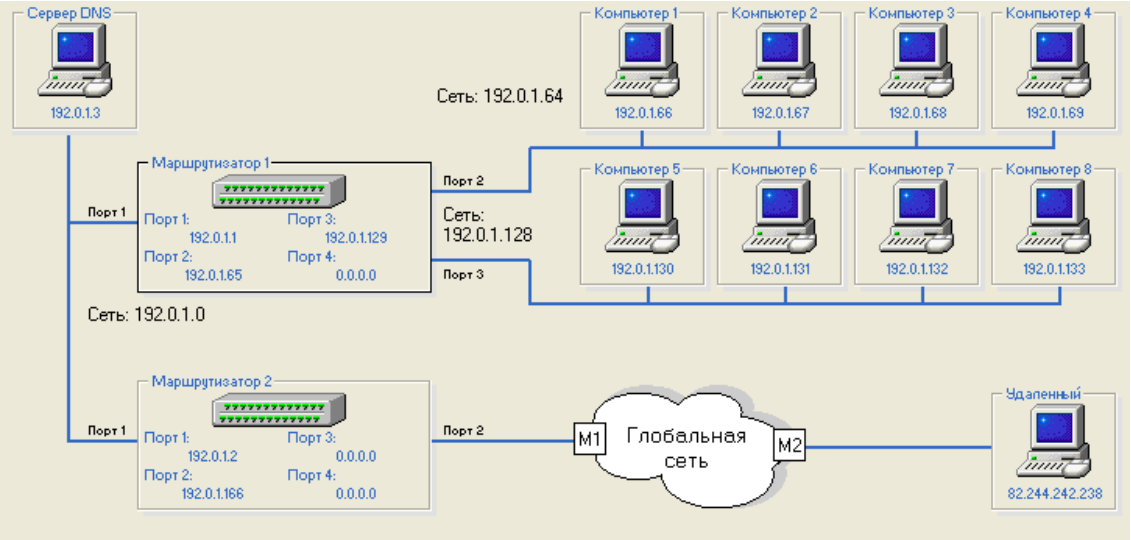


Рисунок 21 – Логическая схема сети

1. Передача данных между компьютерами находящимися в одной подсети.

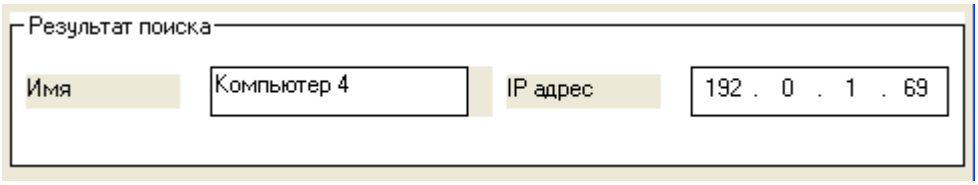
Передатчик – Компьютер 1 (IP адрес: 192.0.1.66).

Приемник – Компьютер 4 (IP адрес: 192.0.1.69).

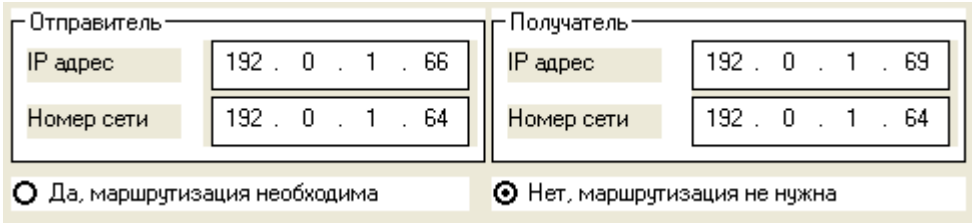
Этапы выполнения:

– отправить данные;

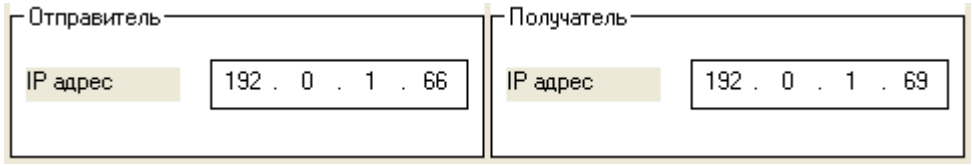
– просмотр hosts;



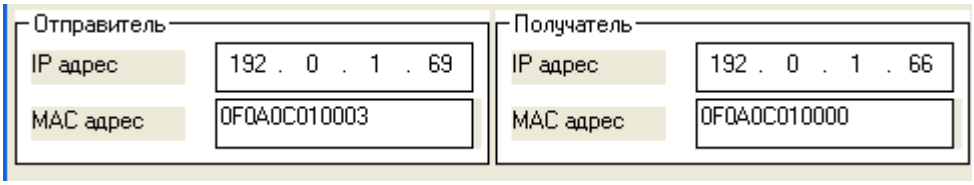
– определение необходимости маршрутизации;



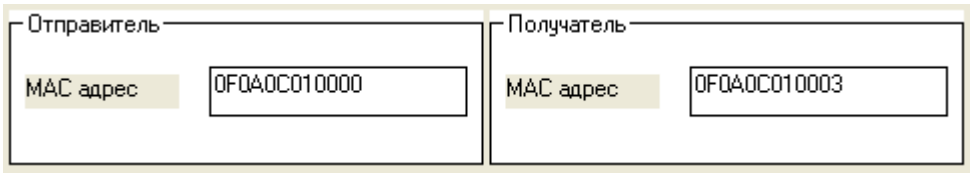
– формирование пакета IP;



– широковещательный ARP запрос;



– формирование кадра Ethernet.



2. Передача данных между компьютерами находящимися в разных подсетях

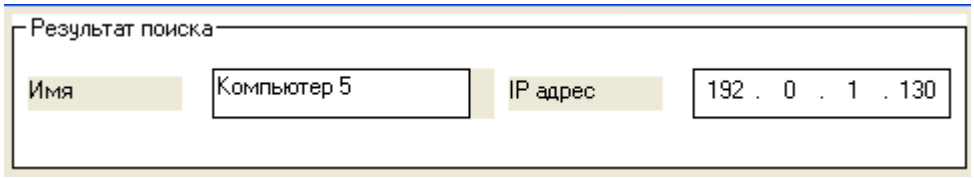
Передатчик – компьютер 1 (IP адрес: 192.0.1.66).

Приемник – компьютер 5 (IP адрес: 192.0.1.130).

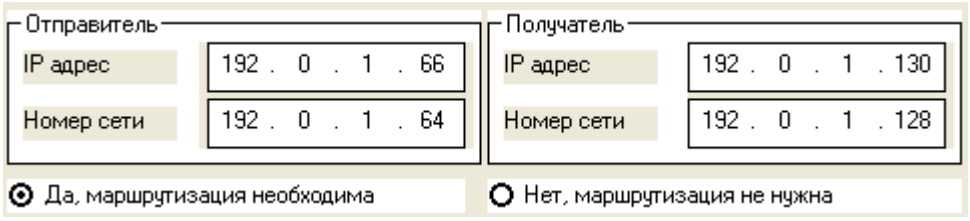
Этапы выполнения:

– отправить данные;

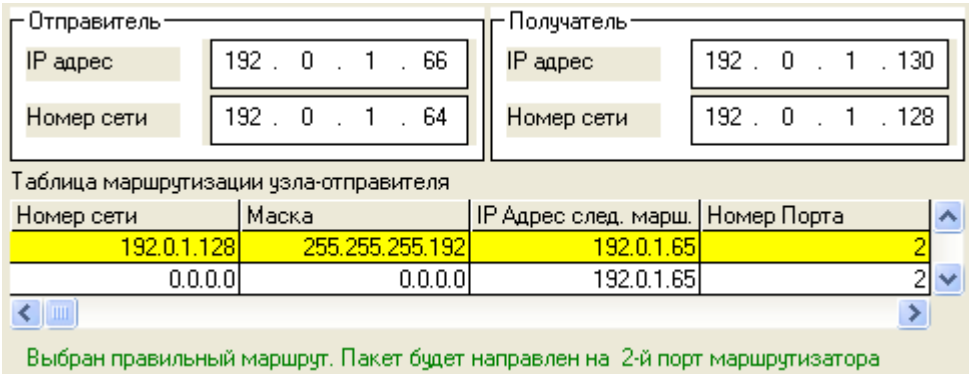
– просмотр hosts;



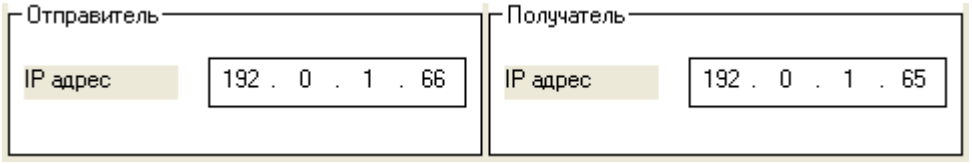
– определение необходимости маршрутизации;



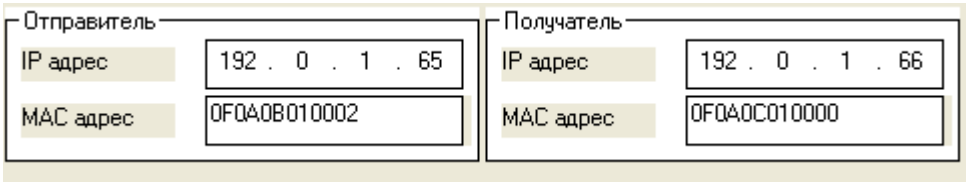
– определение маршрута;



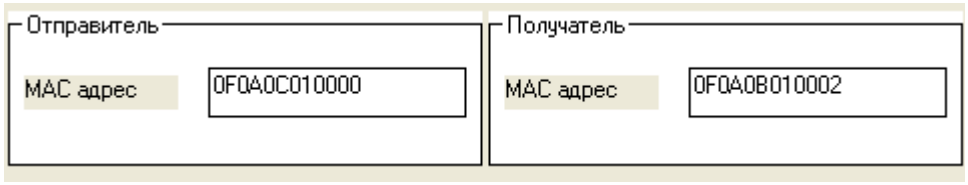
– формирование пакета IP;



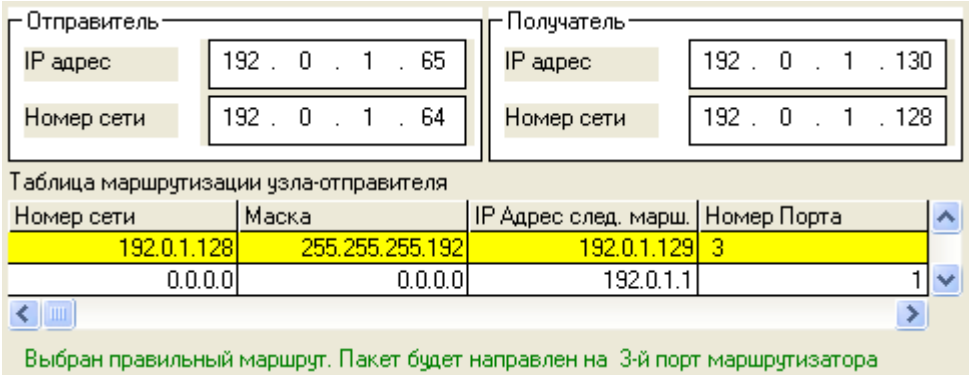
– широковещательный ARP запрос;



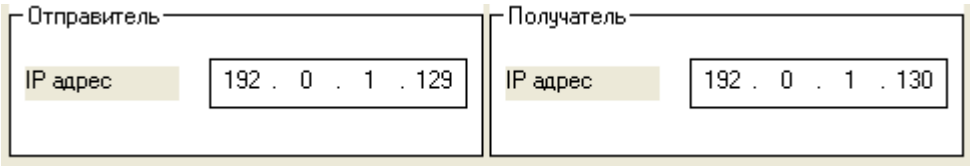
– формирование кадра Ethernet;



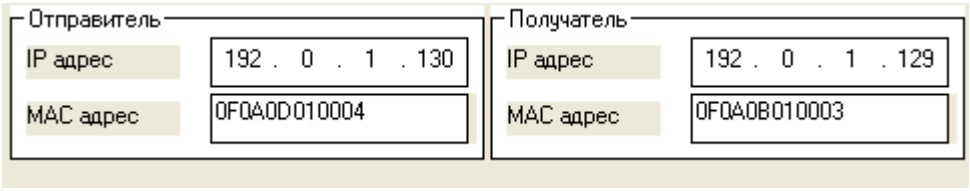
– определение маршрута;



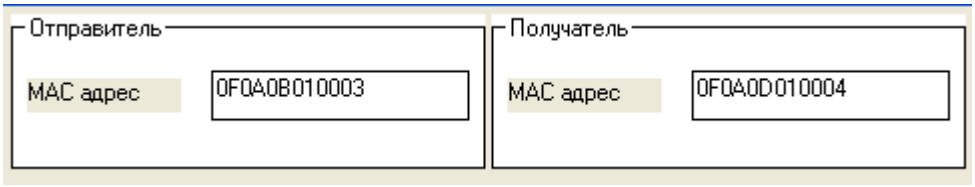
– формирование пакета IP;



– широковещательный ARP запрос;



– формирование кадра Ethernet.



3. Передача данных на удаленный компьютер.

Передатчик – компьютер 1 (IP адрес: 192.0.1.66).

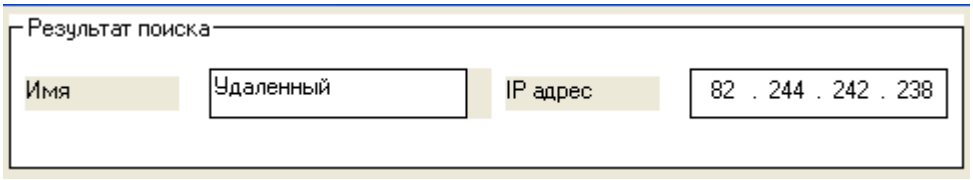
Приемник – удаленный компьютер (IP адрес: 82.244.242.238).

Для передачи данных на удаленный компьютер данные о нем необходимо внести в таблицы маршрутизации, ARP, hosts узла-отправителя (компьютеры, маршрутизаторы, сервер DNS) вручную.

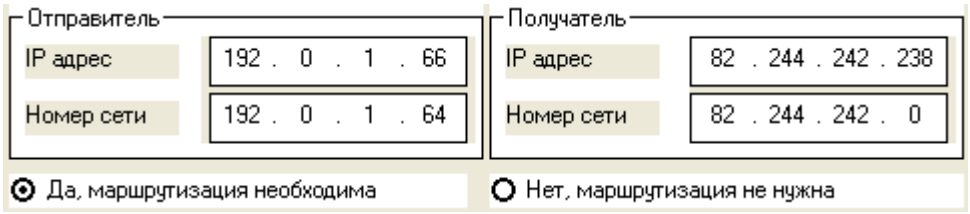
Этапы выполнения:

– оправить данные;

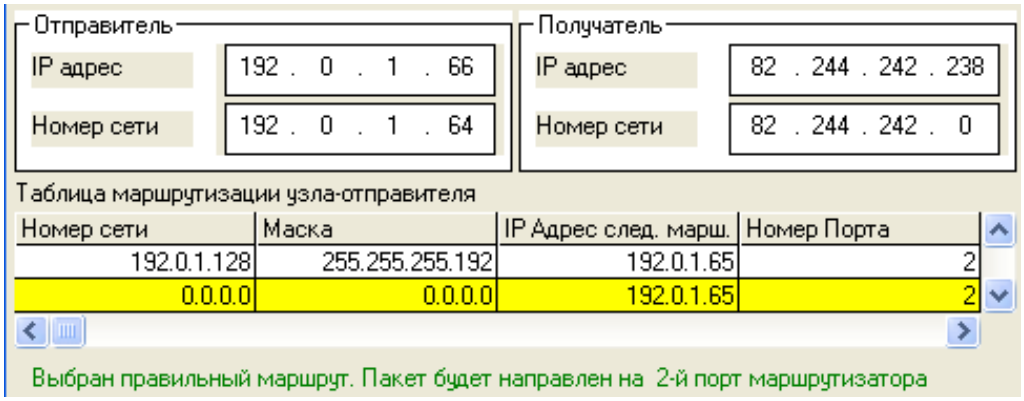
– посмотр hosts;



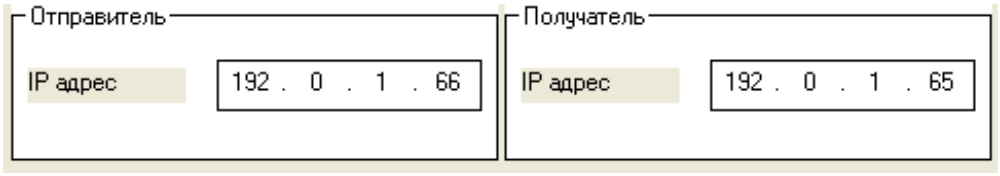
– ределение необходимости маршрутизации;



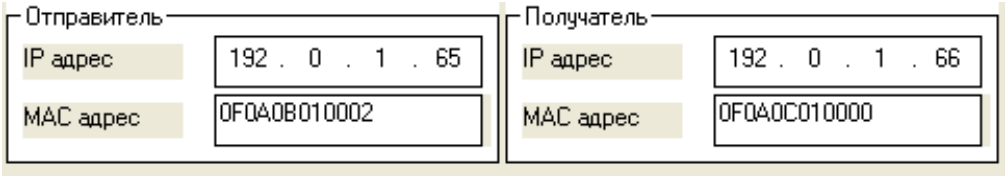
– ределение маршрута;



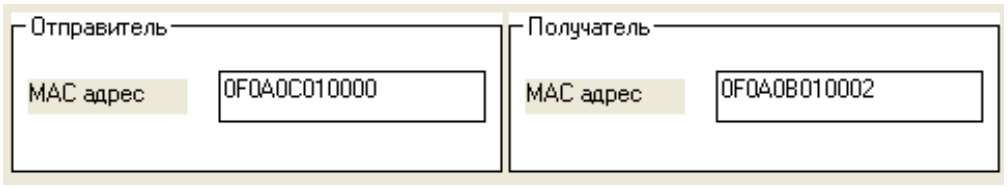
– фрмирование пакета IP;



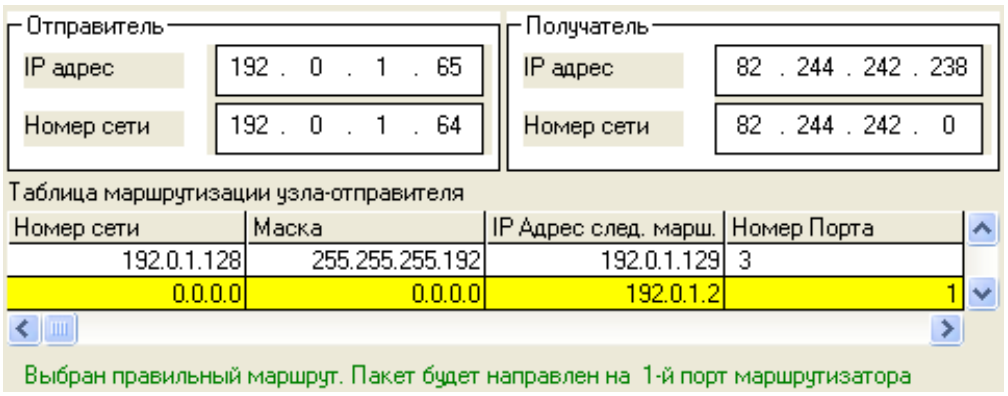
– шроковещательный ARP запрос;



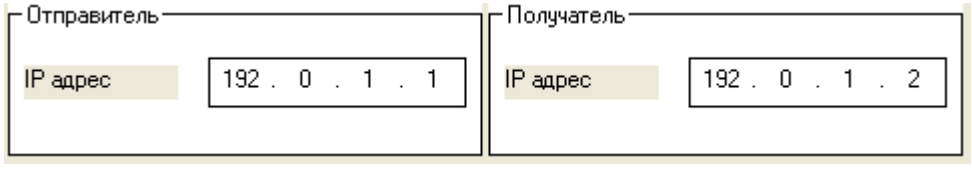
– фрмирование кадра Ethernet;



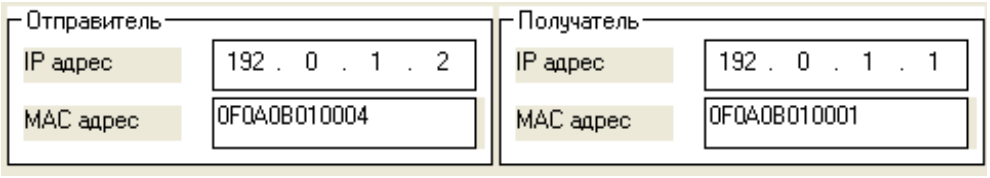
– ределение маршрута;



– фрмирование пакета IP;

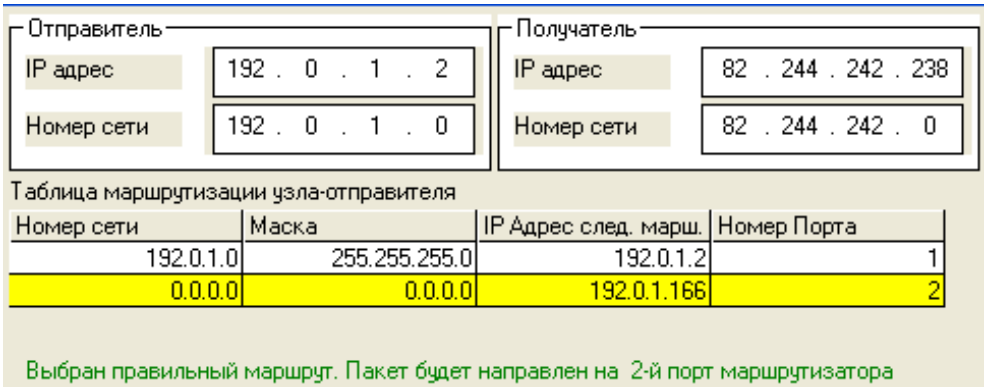


– шроковещательный ARP запрос;

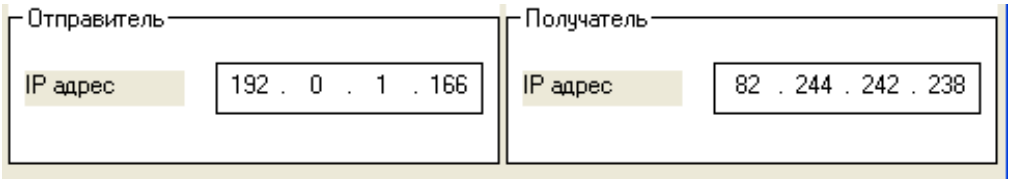


– фрмирование кадра Ethernet;

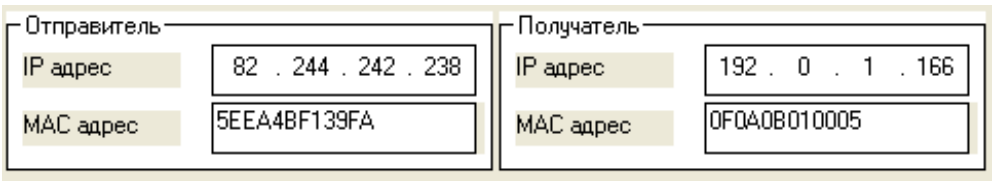
– ределение маршрута;



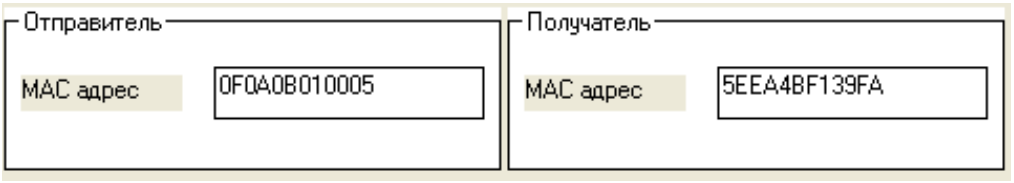
– фрмирование пакета IP;



– широковещательный ARP запрос;



– фрмирование кадра Ethernet.



Содержание отчета по лабораторной работе.

1. Название и цель работы.

2. *Передача данных между компьютерами находящимися в одной подсети.* Представить алгоритм (шаги) выполнения (экранные формы работы с описанием привести по этапно) задания для своего варианта.

3. *Передача данных между компьютерами находящимися в разных подсетях.* Представить алгоритм (шаги) выполнения (экранные формы работы с описанием привести по этапно) задания для своего варианта.

4. *Передача данных на удаленный компьютер.* Представить алгоритм (шаги) выполнения (экранные формы работы с описанием привести по этапно) задания для своего варианта.

5. Обосновать принятые инженерные решения.

6. Выводы по выполненной работе.

7. Список использованных источников.

13 Лабораторная работа № 13. Изучение пакета NetCracker Pro

Цель работы. Изучить основные возможности программного пакета NetCracker Pro и получить навыки построения компьютерных вычислительных сетей.

Теоретическая справка по использованию программного обеспечения NetCracker.

Программа NetCracker предназначена для проектирования и моделирования компьютерных сетей. Для проектирования структуры сети программа предоставляет возможность выбора необходимого оборудования из встроенной базы данных, а также добавления в базу данных и конфигурирования нового оборудования различных типов.

Пользователь размещает выбранные компоненты на наборном поле, задает структуру и тип связей между ними, определяет тип программного обеспечения и характер трафика между узлами сети. В дальнейшем имеется возможность указать перечень анализируемых характеристик и вид отображения статистической информации и выполнить имитационное моделирование спроектированной сети.

На рисунке 22 приведен типичный вид окна программы NetCracker. Панель просмотра компонент, имеющихся в базе данных, располагается обычно в левой части окна и включается с помощью команды **View->Bars->Browser Pane**. Панель содержит несколько закладок.

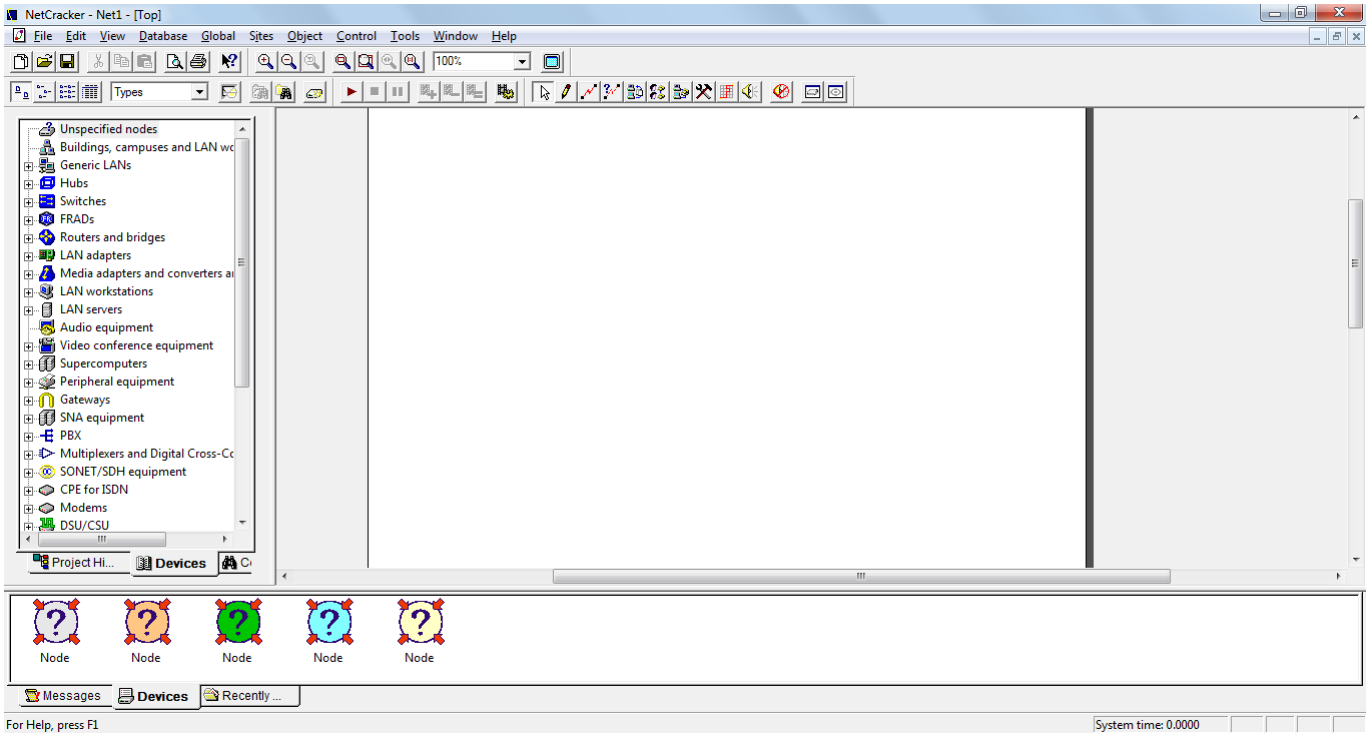


Рисунок 22 - Окна программы NetCracker

Закладка **Project Hierarchy** предназначена для отображения структуры документов создаваемого проекта сети.

Закладка **Devices** предназначена для отображения базы данных устройств. Список устройств имеет несколько видов отображения.

1. **Types** (Типы) – устройства в списке группируются по типам. Затем в каждой группе могут выделяться подтипы устройств по функциональным признакам. После этого устройства разделяются по изготовителям.

2. **Vendors** (Изготовители) – устройства в списке группируются по изготовителям. Затем в каждой группе выделяются подгруппы, соответствующие типу устройств.

3. **User** (Пользовательские) – устройства, определяемые пользователем. В свою очередь также могут группироваться по типам или изготовителям.

Закладка **Compatible Devices** предназначена для отображения списка совместимых устройств.

В нижней части окна программы обычно располагается панель устройств, которая может быть отображена с помощью команды **View->Bars->Image Pane**. Данная панель предназначена для отображения устройств из выбранной группы.

В правой верхней части главного окна программы располагается основное окно, представляющее собой наборное поле. В нем необходимо размещать используемые компоненты при проектировании структуры сети.

Для задания структуры сети необходимо разместить в наборном поле используемые устройства и соединить их линиями связи. Для размещения устройства в наборном поле необходимо, пользуясь панелью просмотра списка устройств, выбрать соответствующий класс и тип устройства. После этого нужно выбрать устройство в панели устройств, перетащить его в наборное поле и разместить в нужном месте. Для дублирования размещенного устройства нужно выбрать требуемое устройство и выполнить команду **Edit->Duplicate**. Команда **Edit->Replicate** позволяет разместить в наборном поле нужное количество устройств. Для этого в диалоговом окне требуется указать количество устройств и нажать кнопку **Replicate**. Переключатель **Organize** позволяет выбрать наиболее удобный вариант размещения устройств в наборном поле. Для удаления устройства из наборного поля необходимо выбрать устройство и выполнить команду **Delete** из меню **Edit** или из контекстного меню.

Для отображения реальной структуры сети организации желательно использовать такие классы компонент, как City (Город), Building (Здание), Campus (Университет), Floor (Этаж) и Room (Комната). Каждый из этих объектов имеет свое наборное поле, раскрыть которое можно при помощи команды **Expand** из меню

Object или из контекстного меню. В новом окне, открывающемся при выполнении данной команды, можно построить ту часть сети, которая соответствует данному объекту.

При выборе различных устройств, используемых для построения сети, прежде всего следует учитывать такие параметры:

- требуемое количество портов;
- требуемый тип портов;
- пропускную способность;
- поддерживаемые транспортные протоколы;
- поддерживаемые протоколы маршрутизации;
- количество слотов.

Для просмотра и редактирования параметров устройств используются команды **Properties**, **Open**, **Configuration**, **Configure Ports** из меню **Object** или команды **Configuration** и **Properties** из контекстного меню. Для задания связей между устройствами (а необходимо выбрать устройство и выполнить команду **Delete** из меню **Edit** или из контекстного меню, точнее между их интерфейсами или портами) необходимо воспользоваться кнопкой **Link**.

Devices на панели режимов указателя мыши (включается командой **View->Bars->Modes**).

После выбора данной кнопки необходимо указать одно из соединяемых устройств и, не отпуская кнопку мыши, растянуть связь до второго устройства. После этого появляется диалоговое окно **Link Assistant**, в котором производится дальнейшее конфигурирование параметров соединения. Первоначально предоставляется возможность определить соединяемые порты устройств и связать их, выполнив щелчок по кнопке **Link**. После этого становится доступной секция **Link Settings**, в которой настраиваются параметры данного соединения, например, используемый протокол (Ethernet), тип среды передачи (Twisted Pair – витая пара), пропускная способность среды (10 Мб/с), длина соединения (до 100 м). В большинстве случаев эти параметры фиксированы и изменяться не могут, хотя иногда имеется возможность выбора из нескольких значений. Например, при

соединении двух оптоволоконных модемов пропускная способность может быть выбрана из списка значений: T3, E3, DSn, Ocn, STSn, STMn (для аналогового модема: 2400, 9600, 14400, 28800 и т. д.). Тип соединения в данном случае единственный – frame relay (ретрансляция кадров).

Передающая среда тоже фиксирована – fiber-optic cable (оптоволокно). При соединении устройств, имеющих порт ISDN, список типов соединений несколько шире – ISDN BRI, ISDN PRI, point-to-point leased line (выделенная линия), dial-up analog line (аналоговая телефонная линия).

После задания структуры сети и топологии связей определяется состав и расположение используемого программного обеспечения. Для этого необходимо в списке компонентов выбрать категорию Network and enterprise software, в ней найти необходимое программное обеспечение и поместить его на соответствующий объект в сети.

В дальнейшем определяется трафик между узлами сети. Для задания трафика необходимо воспользоваться кнопкой Set Traffic на панели режимов указателя мыши для перехода в соответствующий режим. После этого необходимо последовательно выбирать пары абонентских станций (АС) сети, между которыми будет задан трафик. Порядок щелчков на АС определяет направление передачи – сначала отмечается источник, потом приемник. В результате появляется диалоговое окно Profiles, позволяющее задать тип и основные характеристики трафика. Тип трафика выбирается из списка Profiles List, причем указывается по принципу "запросы клиента к серверу", т.е. в качестве приемника может выступать только та АС, на которой функционирует соответствующее программное обеспечение (HTTP/FTP Server, SQL server, File Server). Некоторые типы трафика составляют исключение из данного правила, например, Small office, LAN peer-to-peer traffic, InterLAN traffic и др. Для задания характеристик трафика между указанными АС необходимо нажать кнопку Advanced. В появившемся диалоговом окне Traffic from (АС-источник) to (АС - приемник) задаются закон распределения и диапазон значений для размера запроса (Transaction Size) и интервала между запросами (Time Between Transactions), а также тип протокола прикладного уровня (Application Layer

Protocol). При необходимости добавить новый тип трафика, удалить или изменить параметры существующих типов следует воспользоваться кнопками Add, Remove, Edit и Rename диалогового окна Profiles. Цвет, которым при моделировании будут отображаться пакеты, принадлежащие данному типу трафика, отображается ц゛ルンソスタフハв столбце Color. Характеристики типов трафика, используемые по умолчанию, можно изменить также и с помощью команды Global->Profiles. Также имеется возможность использовать команду Global->Data Flow для конфигурирования потоков данных в сети.

Для указания анализируемых характеристик следует воспользоваться командой «**Statistics**» из контекстного меню или Define Statistics из меню Object. В диалоговом окне Statistical Items можно задать тип характеристики и способ отображения статистической информации в процессе моделирования. Это диалоговое окно будет различным для различных типов объектов, т.е. может изменяться перечень характеристик, а также некоторые способы отображения информации могут быть недоступны. Для одного объекта (выбранного устройства, соединения или потока данных) можно выбирать несколько способов отображения информации (индикатор, число, график).

После этого уже можно производить имитационное моделирование работы сети. Для управления процессом моделирования используются команды пункта меню Control. Команда Start используется для запуска, Pause для приостановки и Stop для полной остановки процесса моделирования. Команды Simulation Faster и Simulation Slower предназначены для изменения скорости моделирования, тогда как команды Animation Faster, Animation Slower, Animation Default предназначены для изменения скорости визуализации процесса. Команда Animation Setup позволяет в диалоговом режиме выбрать наиболее подходящие параметры для интенсивности, скорости и размера пакетов и звонков. Некоторые из перечисленных команд можно выполнить с помощью кнопок, располагающихся на панелях инструментов Zoom и Control.

Для просмотра обобщенных результатов моделирования используется команда Associated Data Flow из меню Object или из контекстного меню. В

результате выполнения данной команды в окне отображается статистика по процентному соотношению количества пакетов для входящих (Incoming Traffic) и исходящих соединений (Outgoing Traffic).

Посредством команд, содержащихся в пункте меню Tools->Reports, можно создать отчеты, обобщающие результаты выполненной работы.

Задание на работу.

1. Используя пакет NetCracker, построить локальную сеть технологии Ethernet со следующими параметрами:

– количество рабочих станций $N_{PC} = MOD_4(NB) + 2$;

– количество серверов $NC = MOD_3(NB) + 1$;

– тип среды передачи $T_{СП} = MOD_3(NB)$;

– тип трафика - LAN peer-to-peer traffic, FTP, E-Mail (SMTP), HTTP, где NB - вариант (порядковый номер студента в журнале группы).

2. Произвести имитационное моделирование работы сети и собрать статистику - средняя загрузка узлов, каналов передачи данных, средняя задержка, количество принятых, отброшенных пакетов.

Таблица 30 – Исходные данные для работы

$T_{СП}$	0	1	2
Среда передачи	Витая пара	Коаксиальный кабель	Оптоволокно

Содержание отчета по лабораторной работе.

1. Название и цель работы.

2. Отобразить в экранной форме работу закладок меню программного пакета NetCracker и теоретически описать его работу и функции по соответсвующим вкладкам меню.

3. Используя пакет NetCracker, представить логическую схему вычислительной сети (экранные формы работы программы), с заданными параметрами.

4. Представить результаты моделирования работы сети

5. Представить результаты расчетов и поведенной статистики - среднюю загрузку узлов, каналов передачи данных, среднюю задержку, количество принятых, отброшенных пакетов.

6. Обосновать принятые инженерные решения.

7. Выводы по выполненной работе.

8. Список использованных источников.

14 Лабораторная работа № 14. Построение локальных вычислительных сетей с использованием технологии Ethernet (ПО NetCracker)

Цель работы. Получить навыки выбора оборудования, кабельной системы для построения инфраструктуры локальной вычислительной сети уровня предприятия по технологии Ethernet с использованием пакета NetCracker.

Теоретическая справка.

Физический уровень Fast Ethernet имеет состоит из трех подуровней:

- подуровень согласования (reconciliation sublayer);
- независимый от среды интерфейс (Media Independent Interface, MII);
- устройство физического уровня (Physical layer device, PHY).

Устройство физического уровня (PHY) обеспечивает кодирование данных, поступающих от MAC-подуровня, для передачи их в среду определенного типа, синхронизацию передаваемых данных, а также прием и декодирование данных в узле-приемнике. Интерфейс MII поддерживает независимый от используемой физической среды способ обмена данными между MAC-подуровнем и подуровнем PHY. Подуровень согласования предназначен для согласования работы подуровня MAC с интерфейсом MII [3, 4].

В таблице 31 приведены основные характеристики сетей Fast Ethernet.

Таблица 31 - Основные характеристики сетей Fast Ethernet

Спецификация физического уровня	100BASE-TX	100BASE-FX	100BASE-T4
Скорость передачи данных, Мб/с	100	100	100
Передача данных	Узкополосная	Узкополосная	Узкополосная
Тип среды передачи	UTP cat. 5	ВОК	UTP cat. 3, 4, 5
Топология	Звезда	Звезда	Звезда
Максимальная длина сегмента, м	100	412 (полудуплекс) 2000 (дуплекс)	100
Максимальное количество абонентов в сегменте	1024	1024	1024
Максимальный диаметр сети, м	205	272	200

Спецификации 100Base-TX и 100Base-FX определяют в качестве среды передачи соответственно двухпарную витую пару и многомодовое оптоволокно. При этом одна витая пара (или одно оптическое волокно) используется для передачи сигнала (T_x), а вторая пара (или второе оптическое волокно) – для приема (R_x). Узлы, поддерживающие эти спецификации, могут осуществлять обмен данными как в полудуплексном (half-duplex), так и в полнодуплексном режиме (full-duplex). Отличительной особенностью полнодуплексного режима является то, что в этом режиме не используется метод доступа к среде CSMA/CD, и отсутствует понятие коллизий – каждый узел одновременно передает и принимает кадры данных по каналам T_x и R_x. Использование полнодуплексного режима возможно только при соединении сетевого адаптера с коммутатором или же при непосредственном соединении коммутаторов. Номинальная скорость обмена данными между узлами при этом увеличивается до 200 Мб/с. Полнодуплексный режим обмена для сетей 100Base-X окончательно не утвержден комитетом IEEE в качестве стандарта. Тем не менее, большинство производителей выпускают как сетевые адаптеры, так и коммутаторы с поддержкой этого режима. Из-за отсутствия стандарта оборудование разных производителей может взаимодействовать некорректно. Прежде всего для полнодуплексного режима должны быть определены процедуры управления потоком кадров, поскольку без этого механизма возможна потеря кадров в

результате переполнения буферного пространства коммутаторов, что всегда крайне нежелательно, так как восстановление информации будет осуществляться более медленными протоколами сетевого и транспортного уровней. Ввиду отсутствия стандарта каждый производитель сам определяет способы управления потоком кадров в коммутаторах и сетевых адаптерах. Обычно, при заполнении буфера устройства до определенного предела, это устройство посылает передающему устройству сообщение о временном прекращении передачи (XOFF). При освобождении буфера посылается сообщение о возможности возобновить передачу (XON) [3, 4].

Спецификация 100Base-T4 была разработана для обеспечения возможности использования для высокоскоростного Fast Ethernet имеющихся кабельных систем на базе витой пары категории 5. Эта спецификация использует все 4 витые пары для того, чтобы можно было повысить общую пропускную способность за счет одновременной передачи потоков бит по нескольким парам. В этом случае используется способ кодирования 8B/6T вместо классического 4B/5B. Каждые 8 бит информации MAC-уровня кодируются 6-ю троичными цифрами, то есть цифрами, имеющими три состояния. Группа из 6-ти троичных цифр затем передается по трем передающим витым парам. Четвертая пара при передаче используется для обнаружения коллизии. Скорость передачи данных по каждой из трех передающих пар равна 33.3 Мб/с, поэтому общая скорость протокола 100Base-T4 составляет 100 Мб/с.

Технология Fast Ethernet рассчитана на подключение конечных узлов – компьютеров с соответствующими сетевыми адаптерами – к многопортовым концентраторам (повторителям) или коммутаторам.

Правила корректного построения сегментов сетей Fast Ethernet определяют:

– ограничения на максимальные длины сегментов, соединяющих терминальное оборудование между собой;

– ограничения на максимальные длины сегментов, соединяющих терминальное оборудование с портом повторителя;

– ограничения на максимальный диаметр сети;

– ограничения на максимальное число повторителей и максимальную длину сегмента, соединяющего повторители [3, 4].

В качестве терминального оборудования (Data Terminal Equipment, DTE) может выступать любой источник кадров данных для сети - сетевой адаптер, порт моста, порт маршрутизатора, модуль управления сетью и другие подобные устройства.

Повторители Fast Ethernet делятся на два класса. Повторители класса I поддерживают все типы систем кодирования физического уровня - 100Base-TX/FX и 100Base-T4. Повторители класса II поддерживают только один тип системы кодирования физического уровня - 100Base-TX/FX или 100Base-T4. В одном домене коллизий допускается наличие только одного повторителя класса I. Это связано с тем, что такой повторитель вносит большую задержку при распространении сигналов из-за необходимости трансляции различных систем сигналов. Максимальное число повторителей класса II в домене коллизий - 2, причем они должны быть соединены между собой кабелем длиной до 5 метров. Небольшое количество повторителей Fast Ethernet не является серьезным препятствием при построении сетей. Во-первых, наличие стековых повторителей снимает проблемы ограниченного числа портов - все каскадируемые повторители представляют собой один повторитель с достаточным числом портов - до нескольких сотен. Во-вторых, применение коммутаторов и маршрутизаторов позволяет разделить сеть на несколько доменов коллизий [3, 4].

Задание на выполнение лабораторной работы.

1. Используя пакет NetCracker, изучить состав и функциональные характеристики типового оборудования локальных сетей на основе технологии Ethernet.

2. В соответствии с вариантом задания построить сеть предприятия с использованием технологий Ethernet и Fast Ethernet, исходя из расчета минимизации стоимости проектируемой сети.

3. Для полученной модели сети задать необходимые типы потоков данных между рабочими станциями и серверами и произвести имитационное

моделирование работы сети.

4. Проанализировать среднюю загрузку сетевого оборудования и среды передачи данных и время ответа для потока данных. Указать участки сети, уязвимые к перегрузкам, и определить средства повышения надежности функционирования сети.

Таблица 32 - Варианты заданий

№ варианта	Тип инфраструктуры	Тип трафика
1	1	2
2	2	3
3	3	4
4	4	1
5	1	3
6	2	4
7	3	1
8	4	2
9	1	4
10	2	1
11	3	2
12	4	3
13	1	1
14	2	2
15	3	3

Таблица 33 - Тип инфраструктуры

№ варианта	Количество зданий	Расстояние между зданиями	Количество этажей	Количество комнат на этаже
1	2	300	4	3
2	2	250	3	3
3	3	200	3	3
4	3	150	2	3
5	2	300	4	3
6	2	250	3	3
7	3	200	3	3
8	2	300	4	3
9	2	250	3	3

Таблица 34 - Тип моделируемого трафика

№ варианта	Количество файло-вых серверов	Количество HTTP-серверов	Количество FTP-серверов	Количество серве-ров базданных
1	3	1	2	2
2	3	2	1	2
3	2	1	2	3

4	2	2	1	3
5	3	1	2	2
6	3	2	1	2
7	2	1	2	3
8	2	2	1	3
9	3	1	2	2

Содержание отчета по лабораторной работе.

1. Название и цель работы.
2. Исходные данные.
3. Характеристики типового оборудования локальных сетей на основе технологии Ethernet.
4. Сеть предприятия с использованием технологий Fast Ethernet.
5. Логическая схема сети.
6. Расчеты параметров ЛВС.
7. Обосновать принятые инженерные решения.
8. Выводы по выполненной работе.
9. Список использованных источников.

Контрольные вопросы.

1. Краткая характеристика технологии Fast Ethernet.
2. Средства логической структуризации сети.
3. Основные функции сетевых адаптеров.
4. Правила построения сетей Fast Ethernet.
5. Понятие среды передачи, основные типы.

15 Лабораторная работа № 15. Объединении сетей Ethernet с помощью маршрутизатора (ПО NetCracker)

Цель работы. Используя программное обеспечение NetCracker разработать локальную сеть по технологии Ethernet на базе маршрутизатора.

Задание к проведению лабораторной работы.

Построить локальную вычислительную сеть (ЛВС) следующей топологии - пять персональных компьютеров (**PC**) образуют сегмент **10BASE-T** на базе концентратора. Другие пять **PC** и сервер объединены коммутатором по технологии **100BASE-T4**. Концентратор и коммутатор соединить маршрутизатором. Сервер может обслуживать **SQL**-клиентов базы данных, клиентов **HTTP**-приложений и предоставлять **FTP** доступ к файлам. Рабочие станции сегмента **10BASE-T** являются клиентами **HTTP** приложений, рабочие станции сегмента **100BASE-T4** являются **SQL**-клиентами базы данных. Кроме этого, все рабочие станции обращаются на сервер за файлами по протоколу **FTP**, а внутри каждого сегмента взаимодействуют друг с другом по трафику **Small office peer-to-peer.**

Размер ответа сервера на запрос (**Reply Size**) рассчитывается по нормальному закону. Математическое ожидание – 1000, дисперсия – 800 (размер в байтах). Задержка ответа на запрос (**Reply Delay**) рассчитывается по экспоненциальному закону, математическое ожидание – 5 (время в секундах).

Вывести статистику:

– для сервера – текущую нагрузку (**current workload**) и количество полученных пакетов;

– для сегмента **100BASE-T4** – процент использования (**average utilization**).

Порядок выполнения работы.

1. Запустите приложение NetCracker (путь - **меню Пуск → Программы → NetCracker Professional 4.0 → NetCracker Professional**). На экране отобразится основное прикладное окно **NetCracker**, которое в дополнение к области заголовка, главному меню и инструментальным панелям, включает в себя три области: окно навигатора устройств или браузер базы данных устройств (**Browser Pane**), рабочее пространство или окно проекта (**Project Pane**) и область окна изображения или окна устройств (**Image Pane**). При запуске NetCracker рабочее пространство содержит

пустой сетевой проект **Net1**. Область окна изображения заполняется выбранными из базы данных изображениями устройств и приложений (здания, рабочие станции, маршрутизаторы и прочие устройства).

2. На активной вкладке **Devices** (устройства) браузера выберите группу **LAN workstations** (рабочие станции локальных сетей). Далее, сделайте щелчок на символе расширения ⊞ для этой группы устройств (рисунок 23). При этом в раскрывшемся списке **LAN workstations** появляются различные группы этих устройств.

3. Разверните список далее, нажимая на символ расширения для **PCs** (персональные компьютеры), затем разверните список, чтобы отобразить стандартные устройства (**Generic Devices**). В области окна изображений выберите устройство **PC.**

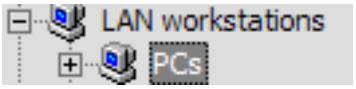


Рисунок 23 - Список устройств

4. Поместите станцию в окно проекта, выполнив следующие действия: нажмите на устройство **PC** в области окна изображения и при помощи технологии **Drag & Drop** переместите его в рабочее пространство.

5. Следующим шагом требуется обеспечить рабочую станцию сетевым интерфейсом. Для этого необходимо при помощи мыши выбрать в браузере группу устройств **LAN adapters** (адаптеры локальной сети, **LAN – Local Area Network** – локальная вычислительная сеть). Разверните первоначально список этой группы, а затем **Ethernet**, и нажмите на папку с названием фирмы - производителя **3Com Corp**. (**Ethernet** – передающая среда ЛВС, стандарт адаптеров ЛВС). В области окна изображения будут представлены сетевые платы категории **Ethernet** корпорации **3Com Corp**. Полоса прокрутки области окна изображения дает возможность просмотреть все доступные устройства выбранной фирмы – производителя.

6. Выберите устройство **Fast EtherLink 10/100 PCI**, представленное в области окна изображения (**PCI** – **Peripheral Component Interconnect**, спецификация на

локальную шину для системных плат, предложенная фирмой **Intel**, промышленный стандарт). В формате **Details (**подробности) изображение платы показано на рисунке 24.

Рисунок 24 - Сетевая плата Fast EtherLink 10/100 PCI

7. Поместите выбранное устройство в рабочую станцию **PC** на рабочем пространстве: наведите указатель мыши на изображение сетевой платы **Fast Ethernet 10/100 PCI**, нажмите левую кнопку и, не отпуская ее, перемещайте плату в рабочее пространство так, чтобы в форме курсора мыши, оказавшегося над изображением рабочей станции, появился символ ⊡. Отпустите левую кнопку мыши.

8. Создайте необходимое количество рабочих станций последовательно выполняя следующие действия:

– выделите в рабочем пространстве станцию с уже установленной в нее сетевой платой и скопируйте эту рабочую станецию, используя команды основного меню **Edit → Copy (**сочетание клавиш **Ctrl+C**);

– вставьте **PC** на пустое место рабочего пространства, используя команды **Edit → Paste** (сочетание клавиш **Ctrl+V**);

– создайте таким способом 10 станций в рабочем пространстве.

При построении сетевого проекта с использованием множественных копий устройства можно выбирать устройства, используя содержимое либо вкладки **Devices** браузера, либо – вкладки **Recently used** в области окна изображения (рисунок 3.3).

9. Поместите в окно проекта сервер, разворачивая в браузере элементы **LAN servers → Generic Devices**. В области окна изображения следует воспользоваться стандартным устройством **Server**.

10. Установите в сервер ранее описанным способом сетевую плату, которую была использована для рабочих станций (путь - **LAN adapters → Ethernet → 3Com**

Corp. → Fast EtherLink 10/100 PCI).

11. Для объединения пяти персональных компьютеров (**PC**) в сегмент **10BASE-T** поместите в рабочее пространство концентратор (hub)одного из ведущих производителей сетевого оборудования – компании **Cisco Systems (**путь - **Hubs → Shared media → Ethernet → Cisco Systems)**. Выделите в браузере устройств hub **MicroHub 1502**. Устройство **MicroHub 1502** от выбранной компании – производителя поддерживает стандарт **10BASE-T** (Спецификация **IEEE 802.3i** для сетей **Ethernet** с использованием неэкранированного кабеля на основе скрученных пар ("витая пара") со скоростью передачи данных по сети 10 Мбит/с). В случае отсутствия указанного устройства, воспользуйтесь методикой поиска в базе данных (лабораторная работа №3), найдите это устройство и поместите в рабочую область.

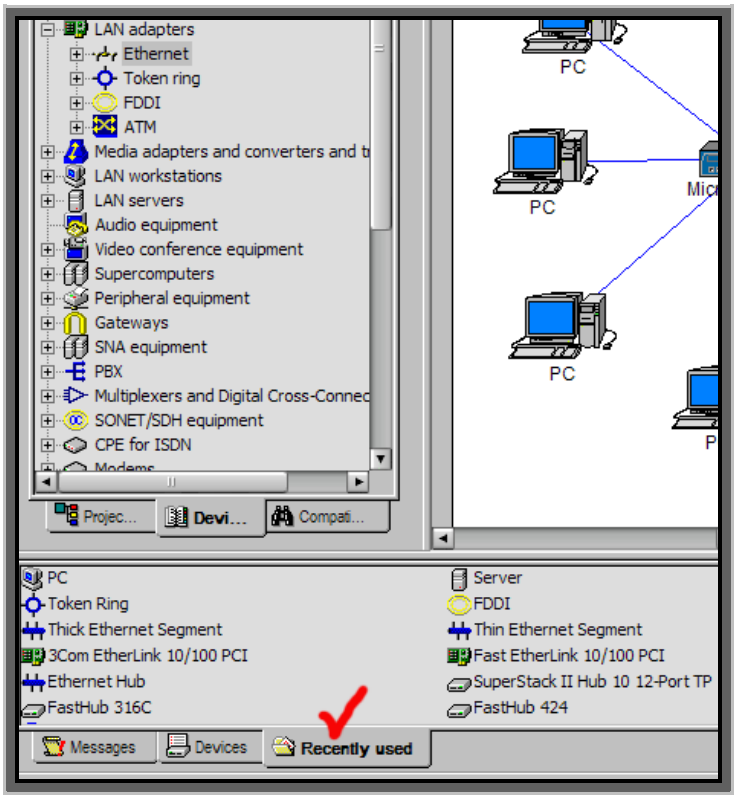


Рисунок 25 - Вкладка браузера **Recently Used** – уже использованные устройства

12. Оставшиеся пять компьютеров и сервер объедините по технологии **100BASE-T4** при помощи коммутатора (**Switch)** от производителя **Cisco Systems**. **(**путь - **Switches → Workgroup → Ethernet → Cisco Systems)**. Найдите устройство **Cisco 1548M Micro Switch 10/100** и поместите его в рабочую область проекта.

13. В соответствии с заданием требуется соединить устройства концентратор (hub) и коммутатор (swith) при помощи маршрутизатора (router – устройство, соединяющее две или несколько физических сетей (подсетей) и передающее пакеты из одной сети (подсети) в другую). Для этого необходимо найти устройство **NETBuilder II Chassis, 4-Slot** и поместить его в рабочую область (путь - **Routers and bridges → Backbone → 3Com Corp. → NETBuilder II Chassis, 4-Slot**).

14. Выбранный маршрутизатор имеет модульную структуру (в данном случае «**4-Slot**» означает возможность расширения четырьмя модулями), поэтому в него можно добавлять соответствующие платы расширения. Сделайте двойной щелчок на устройстве **NETBuilder II Chassis, 4-Slot**, расположенном в окне проекта. Откроется окно диалога конфигурации, которое включает в себя изображение устройства, конфигурационную панель выбора, а также кнопки - **[Device Setup]** (установка устройства), **[Plug-in Setup]** (установка в разъем), **[Close]**, **[Help]**. Щелчком по кнопке **[Close]** закройте диалог конфигурации.

15. В области окна изображения устройств найдите **Ethernet**–модуль протокола **10BASE-T** с 6-ю портами (**NETBuilder II MP Ethernet 10BASE-T Module, 6-Port**) и вставьте его прямо в устройство **NETBuilder II Chassis, 4-Slot** в рабочем пространстве при помощи технологии **Drag & Drop**.

Получите информацию относительно технических характеристик сменного блока: сделайте двойной щелчок на устройстве **NETBuilder II Chassis, 4-Slot**, расположенном в окне проекта → в конфигурационной панели выбора выделите сменный блок **NETBuilder II MP Ethernet 10BASE-T Module, 6-Port** левой клавишей мыши и щелкните по кнопке **[Plug-in Setup]**.

Просмотрите, какие порты допускает этот сменный блок. Для этого в диалоге свойств нажмите на вкладку **Ports.** Закройте диалог свойств, нажав кнопку [**OK].**

Просмотрите конфигурацию **NETBuilder II Chassis, 4-Slot**, для чего в диалоге конфигурации, нажмите кнопку **[Device Setup]**. На вкладке **Ports** видно, какие из портов используются (**Used**), а какие свободны (**Unused**). Кнопкой [**OK]** закройте диалог свойств, а кнопкой **[Close]** – диалог конфигурации.

16. Для соединения созданных сетевых устройств необходимо проделать

следующие операции:

– перейдите в режим физического соединения устройств, щелкнув на панели инструментов **Modes Bar** (рисунок 26) по кнопке **[Link devices]**;

Рисунок 26 - Панель инструментов **Modes Bar**

– используя левую кнопку мыши соедините одну из рабочих станций с концентратором (рисунок 27);

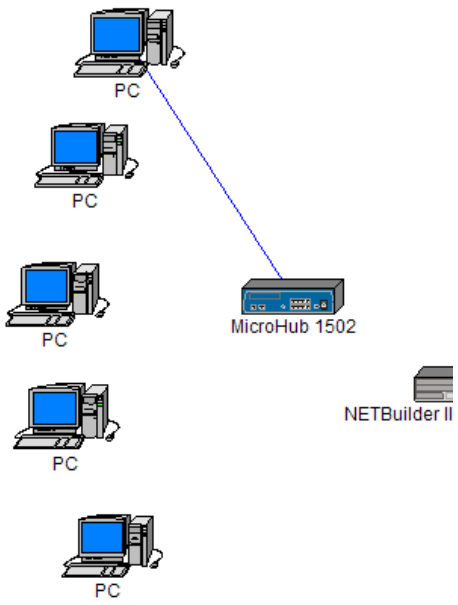


Рисунок 27 - Соединение устройств

– в раскрывшемся диалоге **Link Assistant** (рисунок 28) нажмите кнопку **[Link]** и установите длину сетевого кабеля между устройствами;

– остальные параметры устанавливаются автоматически, убедитесь, что они соответствуют заданию;

– закройте диалог кнопкой **[Close]**;

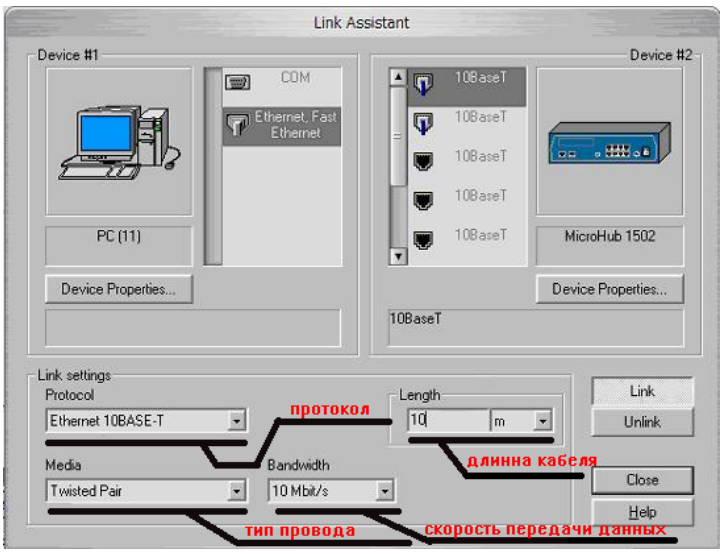


Рисунок 28 - Диалог **Link Assistant**

– аналогичным образом соедините с концентратором остальные 4 рабочие станции;

– на следующем этапе необходимо объединить оставшиеся пять компьютеров и сервер кабелем типа «витая пара», используя технологию **100Base-T4** в соответствии с описанной выше методикой;

– в диалоге **Link Assistant** установите длину между устройствами и протокол **100Base-T4**;

– остальные параметры должны соответствовать значениям по умолчанию.

Примечание. Для соединения двух устройств с параметрами по умолчанию (**Protocol, Media, Cable Length, Bandwidth**), задаваемыми в диалоге **Link Assistant**, необходимо, удерживая клавишу **SHIFT**, последовательно выбрать указателем мыши соединяемые устройства. В этом случае диалог **Link Assistant** не открывается.

17. В соответствии с примечанием соедините построенные сегменты сети:

– концентратор **MicroHub 1502** и маршрутизато **NETBuilder II Chassis, 4-Slot**;

– коммутатор **Cisco 1548M Micro Switch 10/100** и маршрутизатор **NETBuilder II Chassis, 4-Slot**.

18. Установите программное обеспечение (ПО) на сервер. Для выполнения

указанной операции в среде **NetCracker** необходимо в браузере раскрыть раздел **Network and Enterprise software** (рисунок 29), а затем выделить раздел **Server software**, в области окна изображения выбрать ПО **FTP server (**рисунок 29**)** и установить его на сервер в рабочей области проекта.

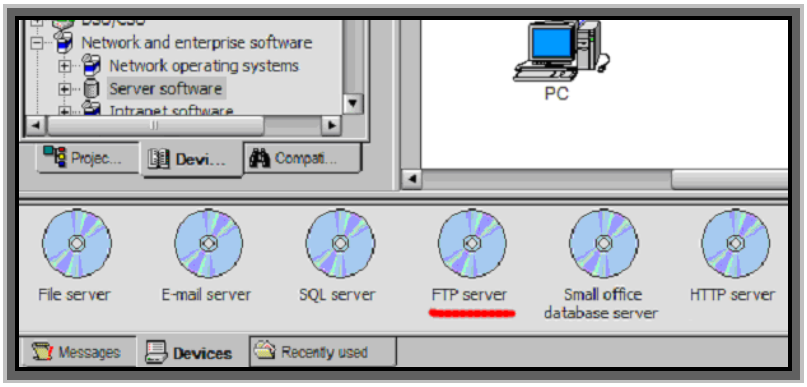


Рисунок 29 - Раздел **Server software**

19. Далее в соответствии с заданием установите на сервер ПО **SQL server** и **HTTP server**. В конечном итоге это дает возможность серверу обслуживать **SQL**-клиентов базы данных, клиентов **HTTP**-приложений, а так же предоставляет **FTP** доступ к файлам, поскольку соответствующие протоколы прикладного уровня **SQL**, **HTTP**, **FTP** (в соответствии с ними будут обмениваться пакетами данных клиентские рабочие станции с сервером) стека протоколов **TCP/IP** окажутся установленными по умолчанию. Убедитесь в этом следующим образом:

– двойным щелчком на сервере откройте диалог настройки конфигурации сервера (рисунок 30);

– в конфигурационной панели выбора выделите элемент **FTP server** (рисунок 30), щелкните по кнопке **[Plug-in Setup]** и перейдите на вкладку **Traffic**, при этом флажок протокола **FTP** будет уже поднят **(**рисунок 30**).**

Нажатие на кнопку **[OK]** приводит к диалогу настройки конфигурации сервера (рисунок 31). По аналогии убедитесь в том, что установлены протоколы **SQL** и **HTTP**.

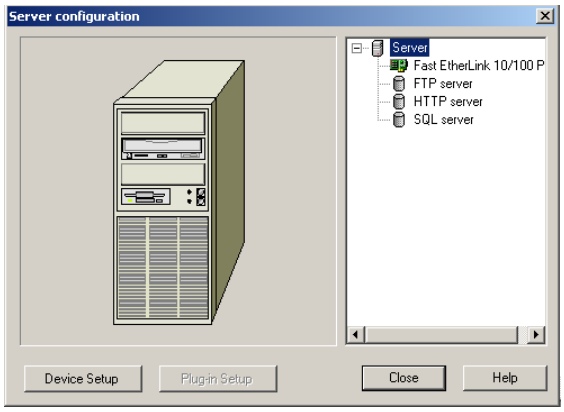


Рисунок 30 - Диалог настройки конфигурации сервера

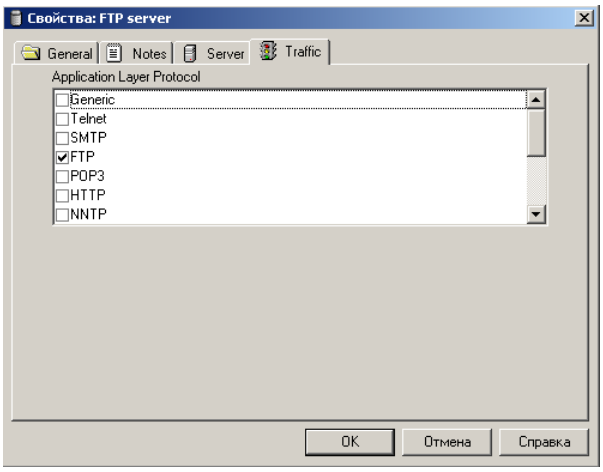


Рисунок 31 - Вкладка настройки протоколов прикладного уровня

20. В соответствии с заданием размер ответа сервера на запрос (**Reply Size**) рассчитывается по нормальному закону: математическое ожидание – 1000, дисперсия – 800 (размеры в байтах). Задержка ответа на запрос (**Reply Delay**) рассчитывается по экспоненциальному закону: математическое ожидание – 5 (время в секундах). Задание указанных параметров для каждого из установленных серверов (**FTP server**, **HTTP server**, **SQL server**) реализуется действиями, описанными в пункте 19, с той лишь разницей, что перейти требуется на вкладку **Server** (рисунок 32).

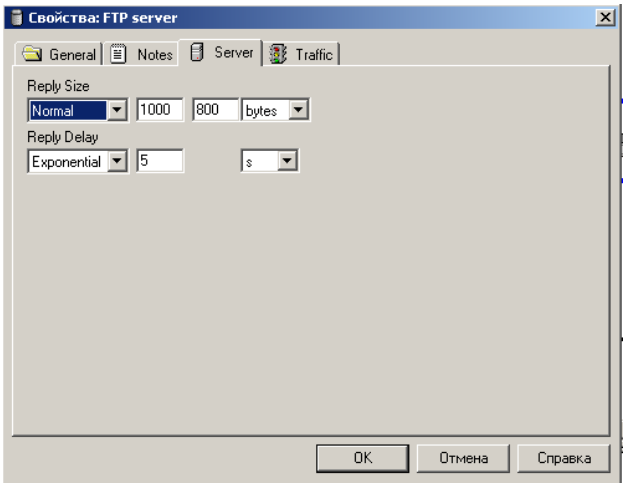


Рисунок 32 - Вкладка настройки параметров ответов сервера

21. Далее требуется настроить трафик, графика которого в инструментальной среде **NetCracker** отобразит обмен данными между устройствами. Для этого на панели инструментов **Modes Bar** нажмите на кнопку **[Set traffic]** (рисунок 33), а затем выполните следующие действия: левой кнопкой мыши щелкните по рабочей станции (источник запроса), затем по серверу (обработчик запроса) и в конце, используя диалог **Profiles** (рисунок 34), необходимо задать вид трафика, который будет отображаться при обмене данными и его цвет. Для завершения настройки трафика нажмите на кнопку **[Assign]**.

В соответствии с заданием настройте трафик остальных рабочих станций.

Рисунок 33 - Панель инструментов **Modes Bar**

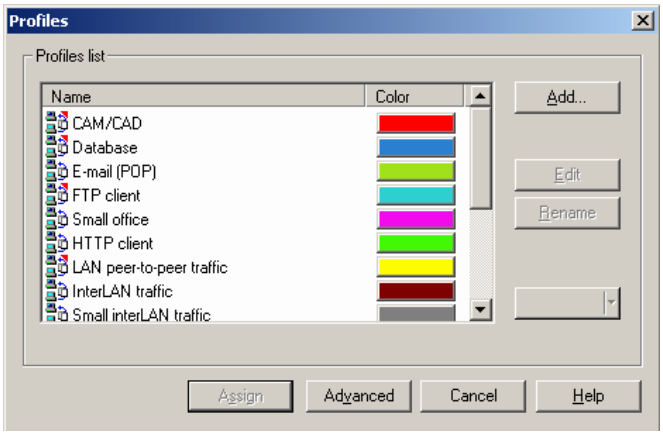


Рисунок 34 - Окно диалога **Profiles**

22. Проверьте правильность соединения, используя команды основного меню **Global → Data Flow…** . Это приводит к открытию диалога **Data Flow** (рисунок 35). Щелчок по кнопке **[Check All]** приводит к поднятию флажков для всего настроенного трафика сетевого проекта, что означает его графическое отображение при последующем запуске анимации. Если в поле **Name** выделить какой-либо трафик, то становятся доступными кнопки **[Set Visible]** и **[Set Invisible]**, позволяющие сделать соответственно видимым или не видимым только этот трафик. Для выделенного трафика можно поменять его параметры (кнопка **[Edit]**) или совсем удалить из проекта (кнопка **[Delete]**).

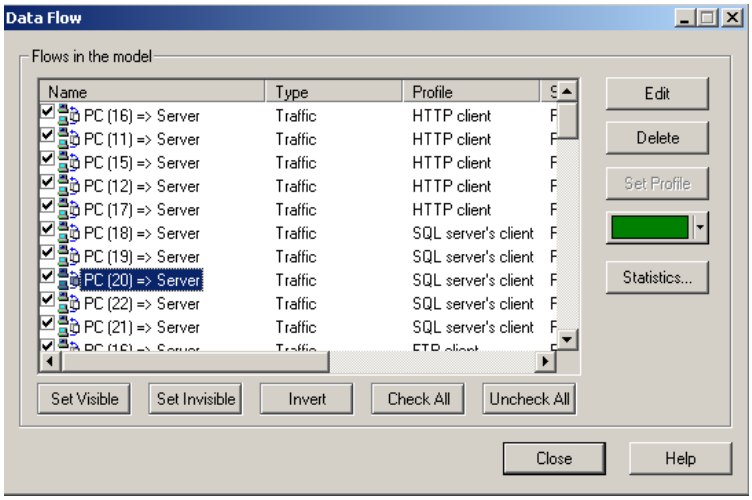


Рисунок 35 - Диалог **Data Flow**

23. Убедитесь в том, что построенная модель ЛВС имеет вид, представленный на рисунке 36.

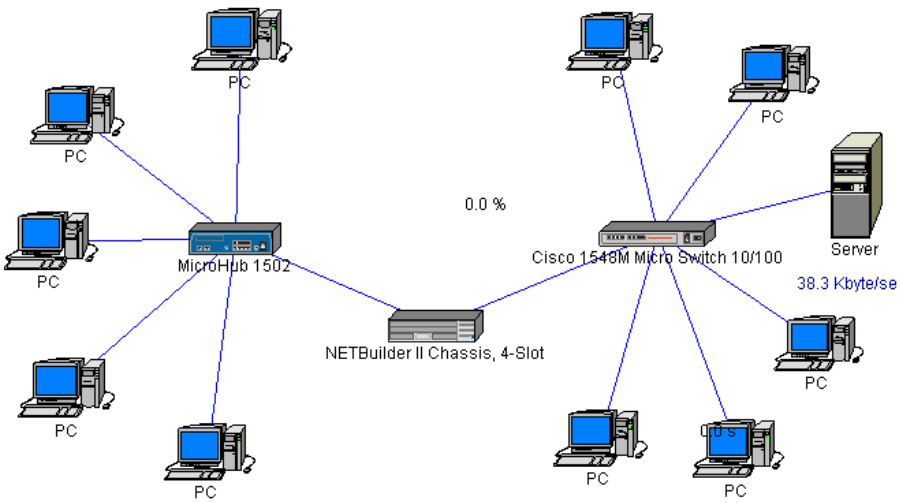


Рисунок 36 - Окончательный вид сетевого проекта

24. Запустив анимацию, проверьте работоспособность сетевого проекта (рисунок 37), используя команды основного меню **Control** → **Start** или нажав на кнопку **Start -**  инструментальной панели **Control**.

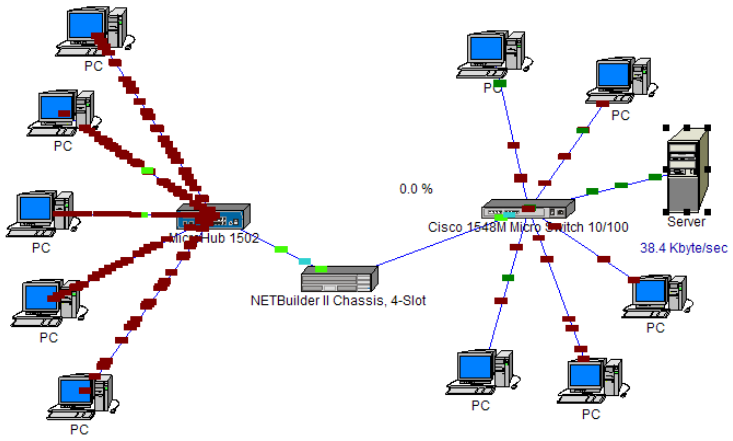


Рисунок 37 - Проект в рабочем состоянии

Содержание отчета по лабораторной работе.

1. Название и цель работы.
2. Структурная схема созданной сети (копия рабочей области сетевого проекта с анимацией).
3. Отчёт полученной статистики в соответствии с заданием.
4. Выводы по выполненной работе.
5. Список использованных источников.

Контрольные вопросы.

1. Укажите особенности **Ethernet** технологий **10BASE-T** и **100BASE-T4**.
2. Что такое технология **SQL**?
3. Что такое **HTTP**-приложение?
4. Как осуществляется **FTP** доступ к файлам?
5. В чём состоят функциональные различия коммутатора и концентратора?
6. В чем состоит функциональное назначението маршрутизатора?

16 Лабораторная работа № 16 Технологии беспроводных сетей. Физический уровень протоколов IEEE 802.11

Цель работы. Исследовать протоколы и технологии передачи данных в беспроводных сетях на физическом уровне и получить навыки выбора оборудования для построения беспроводной локальной вычислительной сети с использованием ПО «NetCracker».

Теоретическая справка.

Для передачи данных стандарты 802.11 используют безлицензионные частотные диапазоны 2,4 ГГц и 5 ГГц. Связь обеспечивается в радиусе 100 - 300 метров от стандартной точки доступа на открытой местности. На сегодняшний день основными стандартами являются 802.11a, 802.11b и 802.11g [2, 3].

В основу стандарта IEEE 802.11 положена сотовая архитектура, причем сеть может состоять как из одной, так и нескольких ячеек. Каждая сота управляется базовой станцией, называемой точкой доступа (Access Point, AP), которая вместе с находящимися в пределах радиуса ее действия рабочими станциями пользователей образует базовую зону обслуживания (Basic Service Set, BSS) Точки доступа многосотовой сети взаимодействуют между собой через распределительную систему (Distribution System, DS), представляющую собой эквивалент магистрального сегмента кабельных локальных сетей. Вся инфраструктура, включающая точки доступа и распределительную систему образует расширенную зону обслуживания (Extended Service Set).

Стандартом предусмотрен также односотовый вариант беспроводной сети, который может быть реализован и без точки доступа, при этом часть ее функций выполняются непосредственно рабочими станциями.

Обеспечение безопасности.

Для защиты WLAN стандартом IEEE 802.11 предусмотрен целый комплекс мер безопасности передачи данных под общим названием Wired Equivalent Privacy, WEP. Он включает средства противодействия несанкционированному доступу к

сети (механизмы и процедуры аутентификации), а также предотвращение перехвата информации (шифрование).

Стандарт IEEE 802.11a.

В отличие от базового стандарта, ориентированного на область частот 2,4 ГГц, спецификациями IEEE 802.11a предусмотрена работа в диапазоне 5 ГГц. В качестве метода модуляции сигнала выбрано ортогональное частотное мультиплексирование (OFDM). Наиболее существенное различие между этим методом и радиотехнологиями DSSS и FHSS заключается в том, что OFDM предполагает параллельную передачу полезного сигнала одновременно по нескольким частотам диапазона, в то время как технологии расширения спектра передают сигналы последовательно. В результате повышается пропускная способность канала и качество сигнала [2, 3].

В таблице 35 приведены основные технические параметры 802.11a.

Таблица 35 - Основные технические параметры 802.11a

Особенность/Функция	Характеристика
Тип связи	Расширение спектра (скачкообразная перестройка частоты - FHSS)
Диапазон частот	Две полосы частот: 5,15-5,35 ГГц и 5,725-5,825 ГГц
Мощность передачи	50 мВт, 250 мВт, 1000 мВт
Скорость передачи данных	Три обязательные (6, 12 и 24 Мбит/с) и пять дополнительных (9, 18, 24, 48 и 54 Мбит/с)
Дальность	До 300 метров на открытом пространстве
Количество устройств в сети	Теоретически до 255 устройств на одну точку доступа; несколько точек доступа в сети
Голосовые каналы	Передача голоса по Интернет-протоколу
Защита данных	Аутентификация: вызов-ответ между точкой доступа и клиентом по стандарту WEP (Wired Equivalent Privacy. 128-битное кодирование.
Адресация	48-битный MAC адрес

К недостаткам IEEE 802.11a относятся высокая потребляемая мощность радиопередатчиков для частот 5 ГГц, а так же меньший радиус действия (оборудование для 2,4 ГГц может работать на расстоянии до 300 м, а для 5 ГГц -

около 100 м).

Стандарт IEEE 802.11b.

Благодаря высокой скорости передачи данных (до 11 Мбит/с), практически эквивалентной пропускной способности обычных проводных локальных сетей Ethernet, а также ориентации на "освоенный" диапазон 2,4 ГГц, этот стандарт завоевал наибольшую популярность у производителей оборудования для беспроводных сетей.

В окончательной редакции стандарт IEEE 802.11b, известный также как Wi-Fi (Wireless Fidelity), был принят в 1999 г. В качестве базовой радиотехнологии в нем используется метод DSSS с 8-разрядными последовательностями Уолша.

Оборудование, работающее на максимальной скорости 11 Мбит/с имеет меньший радиус действия, чем на более низких скоростях, то стандартом 802.11b предусмотрено автоматическое понижение скорости при ухудшении качества сигнала. В таблице 36 приведены основные технические параметры 802.1lb [2, 3].

Таблица 36 - Основные технические параметры IEEE 802.1lb

Особенность/Функция	Характеристика
Тип связи	Расширение спектра (прямая последовательность DSSS)
Диапазон частот	От 2,4 до 2,4835 ГГц
Мощность передачи	100 мВт, 500 мВт
Скорость передачи	До 11 Мбит/сек
Дальность	До 100 метров
Количество устройств в сети	Теоретически до 255 устройств на одну точку доступа, несколько точек доступа в сети
Голосовые каналы	Передача голоса по Интернет-протоколу
Защита данных	Аутентификация - вызов-ответ между точкой доступа и клиентом по стандарту WEP (Wired Equivalent Privacy), 128 - битное кодирование
Адресация	48 - битный MAC адрес

Спецификация IEEE 802.11g.

Этот стандарт принят в середине 2003 года, как развитие стандарта 802.11b. В нем используется тот же частотный диапазон 2,4 ГГц, но вместе с технологией мультиплексирования (OFDM) и алгоритмом

псевдослучайной скачкообразной перестройки рабочей частоты (Frequency Hopping Spread Spectrum -FHSS), что обеспечивает достижение скорости передачи данных до 54 Мбит/с. При этом оборудование стандарта 802.11g совместимо с оборудованием 802.11b, что обеспечивает одновременное подключение к сети устройств стандартов IEEE 802.11g и IEEE 802.11b. Мощность устройств составляет 10 - 100 мВт. В таблице 37 приведены основные технические параметры IEEE 802.11g.

Таблица 37 - Основные технические параметры IEEE 802.11g

Особенность/Функция	Характеристика
Тип связи	Расширение спектра (перестройка частоты - FHSS)
Диапазон частот	От 2,4 до 2,4835 ГГц
Мощность передачи	10 - 100 мВт
Скорость передачи	До 54 Мбит/сек
Дальность	100 – 300 метров
Количество устройств в сети	Теоретически до 255 устройств на одну точку доступа и несколько точек доступа в сети
Голосовые каналы	Передача голоса по Интернет-протоколу
Защита данных	Аутентификация - вызов-ответ между точкой доступа и клиентом по стандарту WEP (Wired Equivalent Privacy), 128-битное кодирование
Адресация	48 - битный MAC адрес

Технология уширения спектра.

В основе всех беспроводных протоколов семейства 802.11 лежит технология уширения спектра (Spread Spectrum, SS). Данная технология подразумевает, что первоначально узкополосный (в смысле ширины спектра) полезный информационный сигнал при передаче преобразуется таким образом, что его спектр оказывается значительно шире спектра первоначального сигнала. То есть спектр сигнала как бы «размазывается» по частотному диапазону. Одновременно с уширением спектра сигнала происходит и перераспределение спектральной энергетической плотности сигнала - энергия сигнала также «размазывается» по спектру. В результате максимальная мощность преобразованного сигнала оказывается значительно ниже мощности исходного сигнала. При этом уровень полезного информационного сигнала может сравниваться с уровнем естественного

шума. В результате сигнал становится «невидимым» - он теряется на уровне естественного шума [2, 3].

Технология DSSS.

При потенциальном кодировании информационные биты - логические нули и единицы - передаются прямоугольными импульсами напряжений. Прямоугольный импульс длительности T имеет спектр, ширина которого обратно пропорциональна длительности импульса. Поэтому чем меньше длительность информационного бита, тем больший спектр занимает такой сигнал.

Для преднамеренного уширения спектра первоначально узкополосного сигнала в технологии DSSS в каждый передаваемый информационный бит (логический 0 или 1) в буквальном смысле встраивается последовательность так называемых чипов. Если информационные биты -логические нули или единицы - при потенциальном кодировании информации можно представить в виде последовательности прямоугольных импульсов, то каждый отдельный чип - это тоже прямоугольный импульс, но его длительность в несколько раз меньше длительности информационного бита. Последовательность чипов представляет собой последовательность прямоугольных импульсов, то есть нулей и единиц, однако эти нули и единицы не являются информационными. Поскольку длительность одного чипа в n раз меньше длительности информационного бита, то и ширина спектра прео бразованного сигнала будет в n-раз больш е ширины спектра первоначального сигнала. При этом и амплитуда передаваемого сигнала уменьшится в n раз.

Чиповые последовательности, встраиваемые в информационные биты, называют шумоподобными кодами (PN-последовательности), что подчеркивает то обстоятельство, что результирующий сигнал становится шумоподобным и его трудно отличить от естественного шума.

Используемые для уширения спектра сигнала чиповые последовательности должны удовлетворять определенным требованиям автокорреляции. Под термином автокорреляции в математике подразумевают степень подобия функции самой себе в различные моменты времени. Если подобрать такую чиповую последовательность,

для которой функция автокорреляции будет иметь резко выраженный пик лишь для одного момента времени, то такой информационный сигнал возможно будет выделить на уровне шума. Для этого в приемнике полученный сигнал умножается на ту же чиповую последовательность, то есть вычисляется автокорреляционная функция сигнала. В результате сигнал становится опять узкополосным, поэтому его фильтруют в узкой полосе частот и любая помеха, попадающая в полосу исходного широкополосного сигнала, после умножения на чиповую последовательность, наоборот, становится широкополосной и обрезается фильтрами, а в узкую информационную полосу попадает лишь часть помехи, по мощности значительно меньшая, чем помеха, действующая на входе приемника (рисунок 38) [2, 3].

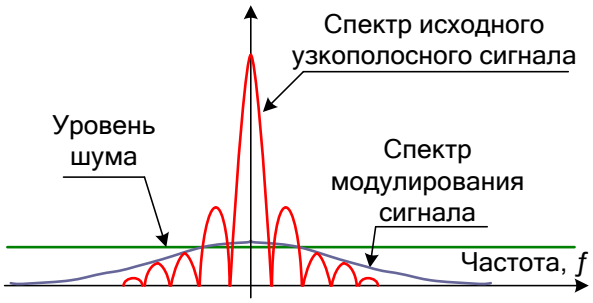


Рисунок 38 - Использование технологии уширения спектра позволяет предавать данные на уровне естественного шума

Двоичное пакетное сверточное кодирование PBCC.

Для дальнейшего рассмотрения протокола 802.11b/b+ необходимо познакомиться с еще одним типом кодирования - так называемым двоичным пакетным сверточным кодированием (Packet Binary Convolutional Coding, PBCC).

Идея сверточного кодирования заключается в следующем. Входящая по следовательность информационных бит преобразуется в специальном сверточном кодере таким образом, чтобы каждому входному биту соответствовало более одного выходного. То есть сверточный кодер добавляет определенную избыточную информацию к исходной последовательности. Если, к примеру, каждому входному биту соответствует два выходных, то говорят о сверточном кодировании со скоростью r = 1/2. Если же каждым двум входным битам соответствует три

выходных, то скорость сверточного кодирования будет составлять уже 2/3.

Любой сверточный кодер строится на основе нескольких последовательно связанных запоминающих ячеек и логических элементов, связывающих эти ячейки между собой. Количество запоминающих ячеек определяет количество возможных состояний кодера. Если, к примеру, в сверточном кодере используется шесть запоминающих ячеек, то в кодере хранится информация о шести предыдущих состояниях сигнала, а с учетом значения входящего бита получим, что в таком кодере используется семь бит входной последовательности. Такой сверточный кодер называется кодером на семь состояний (K = 7).

Выходные биты, формируемые в сверточном кодере, определяются значениями входного бита и битами, хранимыми в запоминающих ячейках, то есть значение каждого формируемого выходного бита зависит не только от входящего информационного бита, но и от нескольких предыдущих битов.

В технологии PBCC используются сверточные кодеры на семь состояний (K = 7) со скоростью r=1/2. Главным достоинством сверточных кодеров является помехоустойчивость формируемой ими последовательности. Дело в том, что при избыточности кодирования даже в случае возникновения ошибок приема исходная последовательность бит может быть безошибочно восстановлена. Для восстановления исходной последовательности битов на стороне приемника применяется декодер Витерби.

Дибит, формируемый в сверточном кодере, используется в дальнейшем в качестве передаваемого символа, но предварительно этот дибит подвергается фазовой модуляции. Причем в зависимости от скорости передачи возможна двоичная, квадратурная или даже восьмипозиционная фазовая модуляция.

Метод пакетного сверточного кодирования опционально предусмотрен как альтернативный метод кодирования в протоколе IEEE 802.11b на скоростях передачи 5,5 и 11 Мбит/с. Кроме того, именно данный режим кодирования лег в основу протокола IEEE 802.11b+ - расширения протокола IEEE 802.11b. Собственно, протокола IEEE 802.11b+ как такового официально не существует, однако данное расширение поддержано многими производителями беспроводных устройств. В

протоколе IEEE 802.11b+ предусматривается еще одна скорость передачи данных -22 Мбит/с с использованием технологии PBCC.

При скорости передачи 5,5 Мбит/с для модуляции дибита, формируемого сверточным кодером, используется двоичная фазовая модуляция, а при скорости 11 Мбит/с - квадратурная фазовая модуляция. При этом для скорости 11 Мбит/с в каждом символе кодируется по одному входному биту и скорость передачи бит соответствует скорости передачи символов, а при скорости 5,5 Мбит/с скорость передачи битов равна половине скорости передачи символов (поскольку каждому входному биту в данном случае соответствует два выходных символа) [2, 3].

Поэтому и для скорости 5,5 Мбит/с, и для скорости 11 Мбит/с символьная скорость составляет 11×10^6 символов в секунду.

Для скорости 22 Мбит/с по сравнению с уже рассмотренной нами схемой PBCC передача данных имеет две особенности. Прежде всего, используется фазовая 8-позиционная фазовая модуляция (8-PSK), то есть фаза сигнала может принимать восемь различных значений, что позволяет в одном символе кодировать уже 3 бита. Кроме того, в схему кроме сверточного кодера добавлен пунктурный кодер (Puncture). Смысл такого решения довольно прост: избыточность сверто чного кодера, равная 2 (на каждый входной бит приходится два выходных), достаточно высока и при определенных условиях помеховой обстановки является излишней, поэтому можно уменьшить избыточность, чтобы, к примеру, каждым двум входным битам соответствовало три выходных.

Для этого можно, конечно, разработать соответствующий сверточный кодер, но лучше добавить в схему специальный пунктурный кодер, который будет просто уничтожать лишние биты.

В сверточный кодер (K = 7, R = 1/2) данные по ступают со скоростью 22 Мбит/с. После добавления избыточности в сверточном кодере биты со скоростью потока 44 Мбит/с поступают в пунктурный кодер 4:3, в котором избыточность уменьшается так, чтобы на каждые четыре входных бита приходилось три выходных. Следовательно, после пунктурного кодера скорость потока составит уже 33 Мбит/с (не информационная, а общая скорость с учетом добавленных

избыточных битов). Полученная в результате последовательность направляется в фазовый модулятор 8-PSK, где каждые три бита упаковываются в один символ. При этом скорость передачи составит 11×10^6 символов в секунду, а информационная скорость - 22 Мбит/с (рисунок 39).

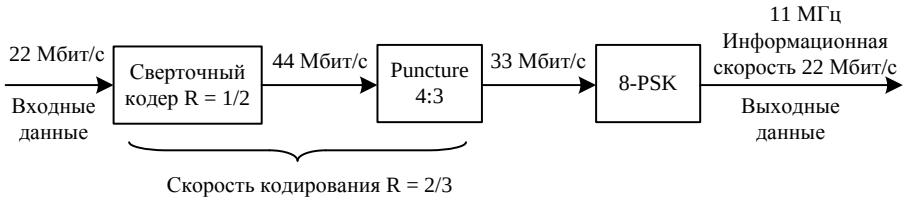


Рисунок 39 - Реализация скорости 22 Мбит/с в протоколе 802.11g

В таблице 38 приводятся соответствие между скоростями передачи и типом кодирования.

Таблица 38 - Соответствие между скоростями передачи и типом кодирования

Скорость передачи, Мбит/с		Метод кодирования	Модуляция	Скорость сверточн. кодиров.	Символьная скорость, 10^6 символ/с	Количество бит в одном символе
1	(обязательно)	Код Баркера	DBPSK	-	1	1
2	(обязательно)	Код Баркера	DQPSK	-	1	2
5,5	(обязательно)	CCk	DQPSK	-	1,375	2
11	(опционально)	PBCC	DBPSK	1/2	11	0,5
22	(обязательно)	CCk	DQPSK	-	1,375	8
	(опционально)	PBCC	DQPSK	1/2	11	1
	(обязательно)	PBCC	DQPSK	3/4	11	2

Ортогональное частотное разделение каналов с мультиплексированием.

Распространение сигналов в открытой среде, которой является радиоэфир, сопровождается возникновением различных помех, источником которых служат сами распространяемые сигналы. Классический пример такого рода помех - эффект многолучевой интерференции сигналов, заключающийся в том, что в результате многократных отражений сигала от естественных преград один и тот же сигнал может попадать в приемник различными путями. Но подобные пути распространения имеют и разные длины, а потому для различных путей распространения ослабление сигнала будет неодинаковым. Следовательно, в точке

приема результирующий сигнал представляет собой суперпозицию (интерференцию) многих сигналов, имеющих различные амплитуды и смещенных друг относительно друга по времени, что эквивалентно сложению сигналов с разными фазами [2, 3].

Следствием многолучевой интерференции является искажение принимаемого сигнала. Многолучевая интерференция присуща любому типу сигналов, но особенно негативно она сказывается на широкополосных сигналах. Дело в том, что при использовании широкополосного сигнала в результате интерференции определенные частоты складываются синфазно, что приводит к увеличению сигнала, а некоторые, наоборот, - противофазно, вызывая ослабление сигнала на данной частоте.

Говоря о многолучевой интерференции, возникающей при передаче сигналов, различают два крайних случая. В первом случае максимальная задержка между различными сигналами не превосходит времени длительности одного символа и интерференция возникает в пределах одного передаваемого символа. Во втором случае максимальная задержка между различными сигналами больше длительности одного символа, а в результате интерференции складываются сигналы, представляющие разные символы, и возникает так называемая межсимвольная интерференция (Inter Symbol Interference, ISI).

Чтобы частично компенсировать эффект многолучевого распространения, используются частотные эквалайзеры, однако по мере роста скорости передачи данных либо за счет увеличения символьной скорости, либо из-за усложнения схемы кодирования, эффективность использования эквалайзеров падает.

В стандарте IEEE 802.11b с максимальной скоростью передачи 11 Мбит/с при использовании CCK - кодов схемы компенсации межсимвольной интерференции вполне успешно справляются с возложенной на них задачей, но при более высоких скоростях такой подход становится неприемлемым.

Поэтому при более высоких скоростях передачи применяется принципиально иной метод кодирования данных - ортогональное частотное разделение каналов с мультиплексированием (Orthogonal Frequency Division Multiplexing, OFDM). Идея

данного метода заключается в том, что поток передаваемых данных распределяется по множеству частотных подканалов и передача ведется параллельно на всех этих подканалах. При этом высокая скорость передачи достигается именно за счет одновременной передачи данных по всем каналам, а скорость передачи в отдельном подканале может быть и невысокой. Поскольку в каждом из частотных подканалов скорость передачи данных можно сделать не слишком высокой, это создает предпосылки для эффективного подавления межсимвольной интерференции [2, 3].

При частотном разделении каналов необходимо, чтобы ширина отдельного канала была, с одной стороны, достаточно узкой для минимизации искажения сигнала в пределах отдельного канала, а с другой - достаточно широкой для обеспечения требуемой скорости передачи. Кроме того, для экономного использования всей полосы канала, разделяемого на подканалы, желательно как можно более плотно расположить частотные подканалы, но при этом избежать межканальной интерференции, чтобы обеспечить полную независимость каналов друг от друга. Частотные каналы, удовлетворяющие перечисленным требованиям, называются ортогональными. Несущие сигналы всех частотных подканалов (а точнее, функции, описывающие эти сигналы) ортогональны друг другу.

Важно, что хотя сами частотные подканалы могут частично перекрывать друг друга, ортогональность несущих сигналов гарантирует частотную независимость каналов друг от друга, а, следовательно, и отсутствие межканальной интерференции (рисунок 40).

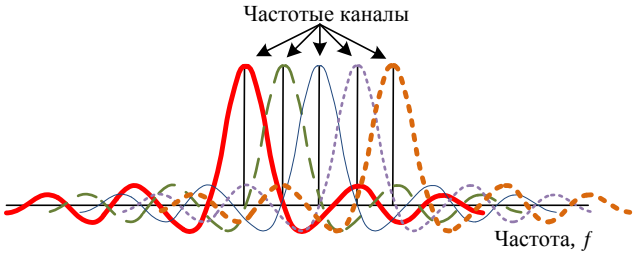


Рисунок 40 - Пример перекрывающихся частотных каналов с ортогональными несущими

Рассмотренный способ деления широкополосного канала на ортогональные частотные подканалы называется ортогональным частотным разделением с

мультиплексированием (OFDM). Одним из ключевых преимуществ метода OFDM является сочетание высокой скорости передачи с эффективным противостоянием многолучевому распространению. Охранный интервал является избыточной информацией и в этом смысле снижает полезную (информационную) скорость передачи, но именно он служит защитой от возникновения межсимвольной интерференции. Эта избыточная информация добавляется к передаваемому символу в передатчике и отбрасывается при приеме символа в приемнике [2, 3].

Скоростные режимы и методы кодирования в протоколе IEEE 802.11g.

В протоколе IEEE 802.11g предусмотрена передача на скоростях 1, 2, 5,5, 6, 9, 11, 12, 18, 22, 24, 33, 36, 48 и 54 Мбит/с. Некоторые из данных скоростей являются обязательными, а некоторые – опциональными. Кроме того, одна и та же скорость может реализовываться при различной технологии кодирования. Протокол IEEE 802.11g включает в себя как подмножество протоколы IEEE 802.11b/b+.

Технология кодирования PBCC опционально может использоваться на скоростях 5,5, 11, 22 и 33 Мбит/с. Вообще же в самом стандарте обязательными являются скорости передачи 1, 2, 5,5, 6, 11, 12 и 24 Мбит/с, а более высокие скорости передачи (33, 36, 48 и 54 Мбит/с) - опциональными.

Для обязательных скоростей в стандарте IEEE 802.11g используется только кодирование CCK и OFDM, а гибридное кодирование и кодирование PBCC является опциональным.

В протоколе IEEE 802.11b для модуляции использовалась либо двоичная (BDPSK), либо квадратурная (QDPSK) относительная фазовая модуляция. В протоколе IEEE 802.11g на низких скоростях передачи также используется фазовая модуляция (только не относительная), то есть двоичная и квадратурная фазовые модуляции BPSK и QPSK. При использовании BPSK-модуляции в одном символе кодируется только один информационный бит, а при использовании QPSK-модуляции - два информационных бита. Модуляция BPSK используется для передачи данных на скоростях 6 и 9 Мбит/с, а модуляция QPSK - на скоростях 12 и 18 Мбит/с. Для передачи на более высоких скоростях используется квадратурная амплитудная модуляция QAM (Quadrature Amplitude Modulation), при которой

информация кодируется за счет изменения фазы и амплитуды сигнала. В протоколе IEEE 802.11g используется модуляция 16-QAM и 64-QAM. В первом случае имеется 16 различных состояний сигнала, что позволяет закодировать 4 бита в одном символе. Во втором случае имеется уже 64 возможных состояний сигнала, что позволяет закодировать последовательность 6 бит в одном символе. Модуляция 16-QAM применяется на скоростях 24 и 36 Мбит/с, а модуляция 64-QAM - на скоростях 48 и 54 Мбит/с [2, 3].

Таблица 39 - Соотношение между различными скоростями передачи и используемыми методами кодирования

Скорость передачи, Мбит/с		Метод кодирования	Модуляция
1	(обязательно)	Код Баркера	DBPSK
2	(обязательно)	Код Баркера	DQPSK
5,5	(обязательно)	сек	DQPSK
	(опционально)	PBCC	DBPSK
6	(обязательно)	OFDM	BPSK
	(опционально)	CCK-OFDM	BPSK
9	(опционально)	OFDM, CCK-OFDM	BPSK
11	(обязательно)	сек	DQPSK
	(опционально)	PBCC	DQPSK
12	(обязательно)	OFDM	QPSK
	(опционально)	CCK-OFDM	QPSK
18	(опционально)	OFDM, CCK-OFDM	QPSK
22	(опционально)	PBCC	DQPSK
24	(обязательно)	OFDM	16-QAM
	(опционально)	CCK-OFDM	
33	(опционально)	PBCC	
36	(опционально)	OFDM, CCK-OFDM	16-QAM
48	(опционально)	OFDM, CCK-OFDM	64-QAM
54	(опционально)	OFDM, CCK-OFDM	64-QAM

Задание к проведению лабораторной работы.

1. Используя пакет NetCracker, изучить состав и функциональные характеристики типового оборудования беспроводных локальных сетей.

2. В соответствии с вариантом задания построить беспроводную сеть с использованием стандартов IEEE 802.11.

3. Для полученной модели сети задать необходимые типы потоков данных между рабочими станциями и серверами и произвести имитационное моделирование работы сети.

4. Провести анализ средней загрузки сетевого оборудования, а также количество теряемых пакетов.

5. Представить выводы по проведенной лабораторной работы.

Таблица 40 - Исходные данные для лабораторной работы

№ варианта	Технология магистрали	Количество HTTP серверов	Количество FTP серверов	Количество беспроводных станций
1	Ethernet	1	2	6
2	Token Ring	2	3	7
3	Ethernet	3	2	5
4	Token Ring	4	1	4
5	Ethernet	1	3	5
6	Token Ring	2	4	4
7	Ethernet	3	3	5
8	Token Ring	4	2	6
9	Ethernet	1	4	3
10	Token Ring	2	1	7
11	Ethernet	3	4	5
12	Token Ring	4	2	3
13	Ethernet	1	1	7
14	Token Ring	2	2	4
15	Ethernet	3	3	2

Содержание отчета по лабораторной работе.

1. Название и цель работы.

2. Привести состав и функциональные характеристики типового оборудования беспроводных локальных сетей.

3. Представит схемные решения беспроводной сети с использованием стандартов IEEE 802.11.

4. Представить результаты имитационного моделирования работы сети.

5. Представить результаты анализа средней загрузки сетевого оборудования, а также количество теряемых пакетов.

6. Выводы по выполненной работе.

7. Список использованных источников.

Контрольные вопросы.

1. Характеристика семейства стандартов IEEE 802.11.
2. С какой целью используется технология уширения спектра?
3. Понятие технологии DSSS.
4. Двоичное пакетное сверточное кодирование PBCC.
5. Ортогональное частотное разделение каналов с мультиплексированием.
6. Какие виды модуляции используются в стандартах IEEE 802.11.

17 Лабораторная работа № 17. Технологии беспроводных сетей. Канальный уровень протоколов IEEE 802.11

Цель работы. Исследовать протоколы и технологии передачи данных в беспроводных сетях на канальном уровне, получить навыки выбора оборудования для построения беспроводной локальной вычислительной сети с использованием ПО «NetCracker».

Теоретическая справка.

Технология коллективного доступа в беспроводных сетях семейства IEEE 802.11 b/g.

Совместное использование среды передачи данных определяются на уровне доступа к среде передачи данных. Этот уровень называют MAC-уровнем (Media Access Control). На MAC-уровне устанавливаются правила совместного использования среды передачи данных одновременно несколькими узлами беспроводной сети. На MAC-уровне определяются два основных типа архитектуры беспроводных сетей - Ad Hoc и Infrastructure Mode [2, 3].

Режим Ad Hoc.

В режиме Ad Hoc (рисунок 41), который называют также Independent Basic Service Set (IBSS) или режимом Peer to Peer (точка-точка), станции непосредственно взаимодействуют друг с другом. Для этого режима нужен минимум оборудования: каждая станция должна быть оснащена беспроводным адаптером. При такой конфигурации не требуется создания сетевой инфраструктуры. Основным недостатком режима Ad Hoc являются ограниченный диапазон действия возможной сети.

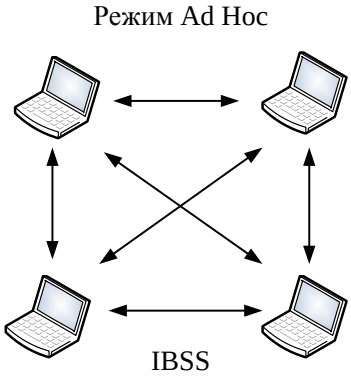


Рисунок 41 - Режим функционирования Ad Hoc

Режим Infrastructure Mode.

В режиме Infrastructure Mode (рисунок 42) станции взаимодействуют друг с другом не напрямую, а через точку доступа (Access Point), которая выполняет в беспроводной сети роль своеобразного концентратора (аналогично тому, как это происходит в традиционных кабельных сетях). Рассматривают два режима взаимодействия с точками доступа - BSS (Basic Service Set) и ESS (Extended Service Set). В режиме BSS все станции связываются между собой только через точку доступа, которая может выполнять также роль моста к внешней сети. В расширенном режиме ESS существует инфраструктура нескольких сетей BSS, причем сами точки доступа взаимодействуют друг с другом, что позволяет передавать трафик от одной BSS к другой. Между собой точки доступа соединяются с помощью либо сегментов кабельной сети, либо радиомостов.

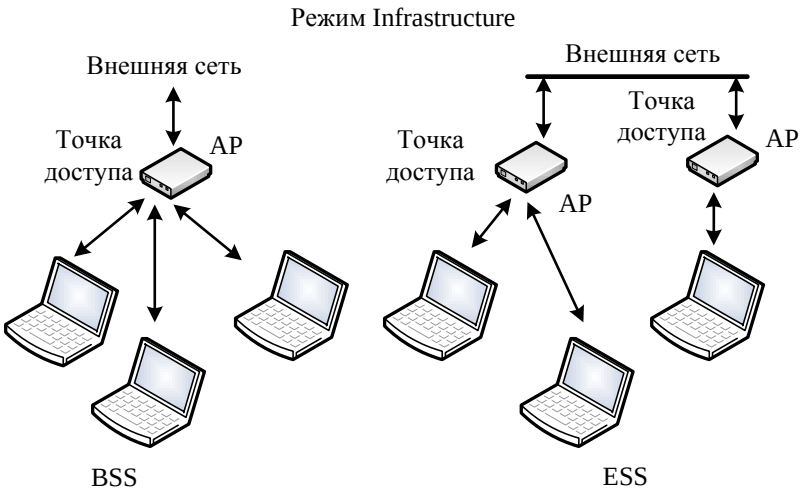


Рисунок 42 - Режим функционирования Infrastructure Mode

Кроме двух различных режимов функционирования беспроводных сетей на MAC-уровне определяются правила коллективного доступа к среде передачи данных. Необходимость существования таких регламентирующих правил вполне очевидна. Если каждый узел беспроводной сети, не соблюдая никаких правил, стал бы передавать данные в эфир, то в результате интерференции нескольких таких сигналов узлы, которым предназначалась отправленная информация, не смогли бы не только ее получить, но и понять, что данная информация адресована им. Именно поэтому, необходимо существование жестких регламентирующих правил, которые определяли бы коллективный доступ к среде передачи данных [2, 3].

На MAC-уровне протокола IEEE 802.11 определяются два типа коллективного доступа к среде передачи данных - функция распределенной координации (Distributed Coordination Function, DCF) и функция централизованной координации (Point Coordination function, PCF).

Функция распределенной координации DCF.

Если организовать совместный доступ к среде передачи данных, когда все узлы передавали данные только тогда и когда среда является свободной, то такой механизм приведет к коллизиям, поскольку велика вероятность того, что два или более узлов одновременно, пытаясь получить доступ к среде передачи данных, решат, что среда свободна и начнут одновременную передачу. Поэтому необходимо

разработать алгоритм, способный снизить вероятность возникновения коллизий и в то же время гарантировать всем узлам сети равноправный доступ к среде передачи данных.

Одним из вариантов организации такого равноправного доступа к среде передачи данных является функция распределенной координации (DCF). Эта функция основана на методе коллективного доступа с обнаружением несущей и механизмом избежания коллизий (Carrier Sense Multiple Access/Collision Avoidance, CSMA/CA). При такой организации каждый узел, прежде чем начать передачу, «прослушивает» среду, пытаясь обнаружить несущий сигнал, и только при условии, что среда свободна, может начать передачу данных.

Однако, в этом случае велика вероятность возникновения коллизий - когда два или более узлов сети одновременно (или почти одновременно) решат, что среда свободна, и начнут предавать данные. Для того чтобы снизить вероятность возникновения подобных ситуаций, используется механизм избежания коллизий (Collision Avoidance, CA). Суть данного механизма заключается в следующем. Каждый узел сети, убедившись, что среда свободна, прежде чем начать передачу, выжидает в течение определенного промежутка времени. Этот промежуток является случайным и складывается из двух составляющих: обязательного промежутка DIFS (DCF Interframe Space) и выбираемого случайным образом промежутка обратного отсчета (backoff time). В результате каждый узел сети перед началом передачи выжидает в течение случайного промежутка времени, что, естественно, значительно снижает вероятность возникновения коллизий, поскольку вероятность того, что два узла сети будут выжидать в течение одного и того же промежутка времени, чрезвычайно мала.

Для того чтобы гарантировать всем узлам сети равноправный доступ к среде передачи данных, необходимо соответствующим образом определить алгоритм выбора длительности промежутка обратного отсчета (backoff time). Промежуток обратного отсчета хотя и является случайным, но в то же время определяется на основании множества некоторых дискретных промежутков времени, то есть, равен целому числу элементарных временных промежутков, называемых тайм-слотами

(SlotTime). Для выбора промежутка обратного отсчета каждый узел сети формирует так называемое окно конкурентного доступа (Contention Window, CW), использующееся для определения количества тайм-слотов, в течение которых станция выжидала перед передачей. Фактически окно CW - это диапазон для выбора количества тайм-слотов, причем минимальной размер окна определяется в 31 тайм-слот, а максимальный размер - в 1023 тайм-слота.

Когда узел сети пытается получить доступ к среде передачи данных, то после обязательного промежутка ожидания DIFS запускается процедура обратного отсчета, то есть включается обратный отсчет счетчика тайм-слотов начиная от выбранного значения окна CW. Если в течение всего промежутка ожидания среда оставалась свободной (счетчик обратного отсчета равен нулю), то узел начинает передачу.

После успешной передачи окно CW формируется вновь. Если же за время ожидания передачу начал другой узел сети, то значение счетчика обратного отсчет останавливается и передача данных откладывается. После того как среда станет свободной, данный узел снова начинает процедуру обратного отсчета, но уже с меньшим размером окна CW, определяемого предыдущим значением счетчика обратного отсчета и соответственно с меньшим значением времени ожидания. При этом очевидно, что чем большее число раз узел откладывает передачу по причине занятости среды, тем выше вероятность того, что в следующий раз он получит доступ к среде передачи данных (рисунок 43) [2, 3].

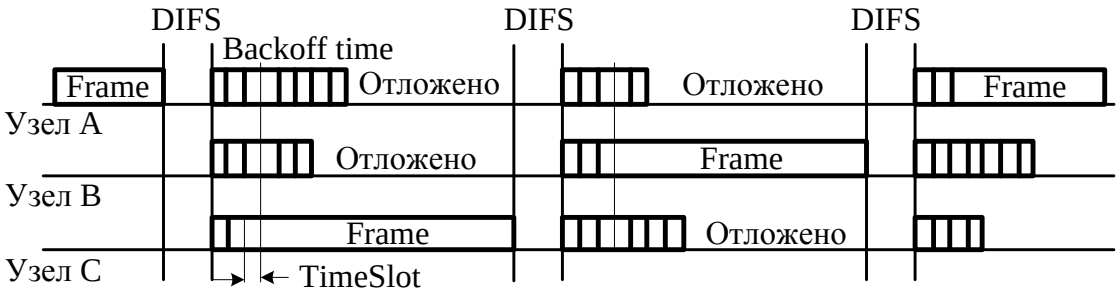


Рисунок 43 - Реализация равноправного доступа к среде передачи данных в методе DCF

Рассмотренный алгоритм реализации коллективного доступа к среде передачи данных гарантирует равноправный доступ всех узлов сети к среде.

Для минимизации коллизий используется следующий алгоритм. После каждого успешного приема кадра принимающая сторона через короткий промежуток SIFS (Short Interframe Space) подтверждает успешный прием, посылая ответную квитанцию - кадр ACK (ACKnowledgement) (рисунок 44). Если в процессе передачи данных возникла коллизия, то передающая сторона не получает кадр ACK об успешном приеме [2, 3].

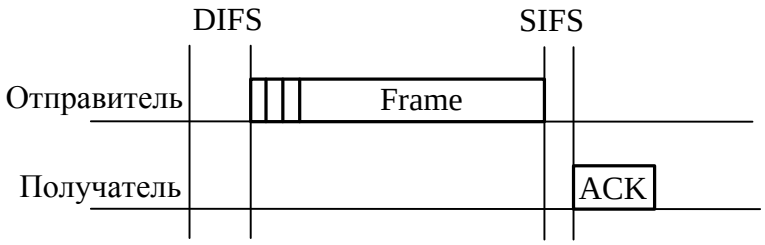


Рисунок 44 - Кадры квитанции, отсылаемые в случае успешной передачи данных

В этом случае размер CW-окна для передающего узла увеличивается почти вдвое. Так, если для первой передачи размер окна равен 31 слоту, то для второй попытки передачи он уже составляет 63 слота, для третьей - 127 слотов, для четвертой - 255, для пятой - 511, а для всех последующих - 1023 слота. То есть для каждой i-й передачи (если все предыдущие оказались безуспешными) размер CW-окна увеличивается по следующему правилу:

$$CW_i = 2CW_{i-1} + 1. \quad (15)$$

Таким образом, увеличение размера окна происходит динамически по мере роста числа коллизий, что позволяет, с одной стороны, уменьшить временные задержки и, с другой стороны, снизить вероятность возникновения коллизий.

Использование кадров данных небольшого размера, хотя и позволяет гарантировать равноправный доступ всех узлов к среде передачи данных и минимизирует издержки при возникновении коллизий, не может не отразиться негативно на полезном сетевом трафике. Так как каждый кадр наряду с полезной

информацией содержит информацию служебную (заголовок кадра). При уменьшении размера кадра сокращается величина именно полезной информации (пользовательских данных), что обусловливает передачу по сети избыточного количества служебной информации.

Рассмотренный механизм регламентирования коллективного доступа к среде передачи данных имеет одно узкое место - так называемую проблему скрытых узлов. Из-за наличия естественных препятствий возможна ситуация, когда два узла сети не могут «слышать» друг друга напрямую. Такие узлы называют «скрытыми».

Для того чтобы разрешить проблему скрытых узлов, функция DCF опционально предусматривает возможность использования алгоритма RTS/CTS.

Алгоритм RTS/CTS.

В соответствии с алгоритмом RTS/CTS каждый узел сети, перед тем как послать данные в «эфир», сначала отправляет специальное короткое сообщение, которое называется RTS (Ready To Send) и означает готовность данного узла к отправке данных. Такое RTS-сообщение содержит информацию о продолжительности предстоящей передачи и об адресате и доступно всем узлам в сети (если только они не скрыты от отправителя). Это позволяет другим узлам задержать передачу на время, равное объявленной длительности сообщения. Приемная станция, получив сигнал RTS, отвечает посылкой сигнала CTS (Clear To Send), свидетельствующего о готовности станции к приему информации. После этого передающая станция посылает пакет данных, а приемная станция должна передать кадр ACK, подтверждающий безошибочный прием. Последовательность отправки кадров между двумя узлами сети показана на рисунке 45 [2, 3].

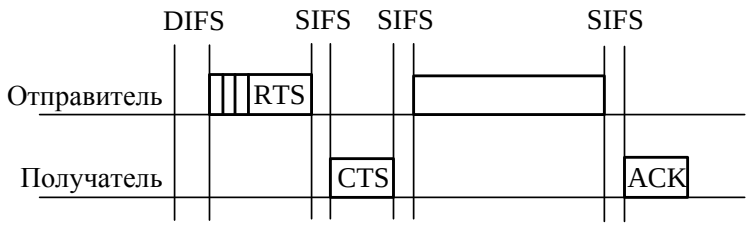


Рисунок 45 - Взаимодействие между двумя узлами сети в соответствии с алгоритмом RTS/CTS.

Теперь рассмотрим ситуацию, когда сеть состоит из четырех узлов - A, B, C и D (рисунок 46). Предположим, что узел C находится в зоне досягаемости только узла A, узел A находится в зоне досягаемости узлов C и B, узел B находится в зоне досягаемости узлов A и D, а узел D находится в зоне досягаемости только узла B. То есть в такой сети имеются скрытые узлы: узел C скрыт от узлов B и D, узел A скрыт от узла D.

В подобной сети алгоритм RTS/CTS позволяет справиться с проблемой возникновения коллизий, которая не решается посредством рассмотренного базового способа организации коллективного доступа в DCF. Действительно, пусть узел A пытается передать данные узлу B. Для этого он посылает сигнал RTS, который, помимо узла B, получает также узел C, но не получает узел D. Узел C, получив данный сигнал, блокируется, то есть приостанавливает попытки передавать сигнал до момента окончания передачи между узлами A и B. Узел B, в ответ на полученный сигнал RTS, посылает кадр CTS, который получают узлы A и D. Узел D, получив данный сигнал, также блокируется на время передачи между узлами A и B.

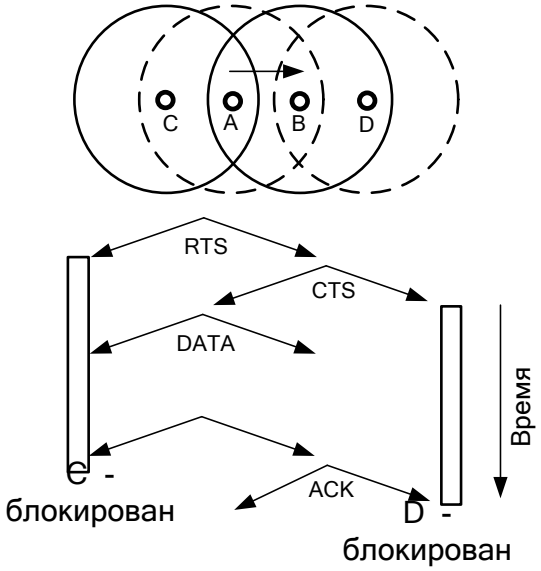


Рисунок 46 - Решение проблемы скрытых узлов в алгоритме RTS/CTS

У алгоритма RTS/CTS имеются свои подводные камни, которые в определенных ситуациях могут приводить к снижению эффективности использования среды передачи данных. К примеру, в некоторых ситуациях

возможно такое явление, как распространение эффекта ложных блокировок узлов, что в конечном счете может привести к ступору в сети [2, 3].

Пример. Пусть узел B (рисунок 47) пытается передать данные узлу A, посылая ему кадр RTS. Поскольку этот кадр получает также и узел C, то он блокируется на время передачи между узлами A и B. Узел D, пытаясь передать данные узлу C, посылает кадр RTS, но поскольку уз ел C заблокирован, то он не получает ответа и начинает процедуру обратного отсчета с увеличенным размером окна. В то же время кадр RTS, посланный узлом D, получает и узел E, который, ложно предполагая, что за этим последует сеанс передачи данных от узла D к узлу C, блокируется. Однако это ложная блокировка, поскольку реально между узлами D и C передачи нет. Более того, если узел F попытается передать данные ложно заблокированному узлу E и пошлет свой кадр RTS, то он ложно заблокирует узел G.

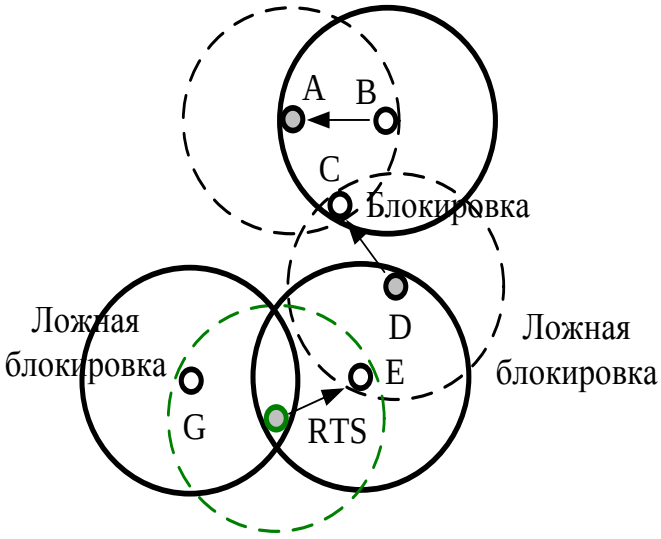


Рисунок 47 - Возникновение ложных блокировок узлов сети

Описанное явление ложной блокировки узлов может приводить к кратковременному ступору всей сети [2, 3].

Функция централизованной координации PCF.

Рассмотренный механизм распределенной координации DCF является базовым для протоколов IEEE 802.11 и может использоваться как в беспроводных сетях, функционирующих в режиме Ad Hoc, так и в сетях, функционирующих в

режиме Infrastructure, то есть в сетях, инфраструктура которых включает точку доступа.

Функция централизованной координации (Point Coordination Function) PCF.

Механизм PCF является опциональным и применяется только в сетях с точкой доступа. В случае задействования механизма PCF один из узлов сети (точка доступа) является центральным и называется центром координации (Point Coordinator, PC). На центр координации возлагается задача управления коллективным доступом всех остальных узлов сети к среде передачи данных на основе определенного алгоритма опроса или исходя из приоритетов узлов сети. Центр координации опрашивает все узлы сети, внесенные в его список, и на основании этого опроса организует передачу данных между всеми узлами сети. Важно, что такой подход полностью исключает конкурирующий доступ к среде, как в случае механизма DCF, и делает невозможным возникновение коллизий, а для времезависимых приложений гарантирует приоритетный доступ к среде. Таким образом, PCF может использоваться для организации приоритетного доступа к среде передачи данных. Функция централизованной координации не отрицает функцию распределенной координации, а скорее, дополняет ее, накладываясь поверх. Фактически в сетях с механизмом PCF реализуется как механизм PCF, так и традицио нный механизм DCF. В течение определенного промежутка времени реализуется механизм PCF, затем - DCF, а потом все повторяется заново.

Для того чтобы иметь возможность чередовать режимы PCF и DCF, необходимо, чтобы точка доступа, выполняющая функции центра координации и реализующая режим PCF, имела бы приоритетный доступ к среде передачи данных. Это можно сделать, если использовать конкурентный доступ к среде передачи данных (как и в методе DCF), но для центра координации разрешить использовать промежуток ожидания, меньший DIFS. В этом случае если центр координации пытается получить доступ к среде, то он ожидает (как и все остальные узлы сети) окончания текущей передачи и, поскольку для него определяется минимальный режим ожидания после обнаружения «тишины» в эфире, первым получает доступ к

среде. Промежуток ожидания, определяемый для центра координации, называется PIFS (PCF Interframe Space), причем SIFS<PIFS<DIFS.

Максимальная скорость передачи данных в протоколах IEEE 802.11b/g.

Максимальная скорость, определяемая протоколом IEEE 802.11b, составляет 11 Мбит/с, а для протокола IEEE 802.11g - 54 Мбит/с.

Следует различать полную скорость передачи и полезную скорость передачи. Технология доступа к среде передачи данных, структура передаваемых кадров, заголовки, прибавляемые к передаваемым кадрам на различных уровнях модели OSI, - все это предполагает наличие достаточно большого объема служебной информации. В результате полезная или реальная скорость передачи, то есть скорость передачи пользовательских данных, всегда оказывается ниже полной скорости передачи [2, 3].

Реальная скорость передачи зависит от структуры беспроводной сети. Так, если все клиенты сети используют один и тот же протокол, например IEEE 802.11g, то сеть является гомогенной и скорость передачи данных в такой сети выше, чем в смешанной сети, где имеются клиенты как IEEE 802.11g, так и IEEE 802.11b. Дело в том, что клиенты IEEE 802.11b «не слышат» клиентов IEEE 802.11g, которые используют OFDM-кодирование. Поэтому с целью обеспечения совместного доступа к среде передачи данных клиентов, использующих различные типы модуляции, в подобных смешанных сетях точки доступа должны отрабатывать определенный механизм защиты. В результате использования механизмов защиты в смешанных сетях реальная скорость передачи становится еще меньше.

Кроме того, реальная скорость передачи данных зависит и от используемого протокола (TCP или UDP) и от размера длины пакета. Естественно, что протокол UDP предусматривает более высокие скорости передачи. Теоретические максимальные скорости передачи данных для различных типов сетей и протоколов представлены в таблице 41.

Таблица 41 - Скорости передачи данных для различных типов сетей и протоколов

Тип сети	Модуляция	Максимальная скорость соединения, Мбит/с	Теоретическая максимальная скорость передачи по протоколу TCP, Мбит/с	Теоретическая максимальная скорость передачи по протоколу UDP, Мбит/с
802.11b	сек	11	5,9	7,1
802.11g (совместно с 802.11b)	OFDM/CCK	54	14,4	19,5
802.11g	OFDM/CCK	54	24,4	30,5

Расширения протокола IEEE 802.11g.

Фактически, речь идет о неком нестандартизированном расширении протокола IEEE 802.11g, позволяющем добиться более высоких скоростей передачи. В решениях под маркой 802.11g+ на физическом уровне используются те же самые режимы передачи, что и в протоколе IEEE 802.11g. Собственно, речь идет не об изменении физического уровня, а о некоторых изменениях MAC-уровня, то есть уровня доступа к среде передачи данных.

В основе всех технологий расширения протокола IEEE 802.11g лежат такие принципы, как пакетная передача (packet bursting), позаимствованная из протокола IEEE 802.11e, а также сжатие данных, быстрые кадры и связывание каналов. В режиме блочной передачи все пакеты, передаваемые в одном блоке, используют сокращенные заголовки, что позволяет уменьшить объем передаваемой служебной информации и тем самым увеличить полезный трафик.

Технология Super-G использует пакетную передачу, "быстрые кадры" и сжатие данных "на лету", а также связывание двух каналов. Основная идея, лежащая в основе технологии Super-G заключается в связывании двух каналов (channel bonding) для увеличения общей пропускной способности. Поскольку теоретическая пропускная способность одного канала в протоколе IEEE 802.11g составляет 54 Мбит/с, то при связывании двух каналов можно достигнуть пропускной способности в 108 Мбит/с. Именно поэтому, продукты, поддерживающие технологию Super-G часто сопровождают надписями типа 108 Мбит/с [2, 3].

Стандарт IEEE 802.11g используют одиннадцать каналов в частотной полосе

2,4 ГГц, которые разделены промежутками по 5 МГц. Поскольку общепринятая ширина каждого канала составляет 22 МГц имеется три канала без частичного наложения (1, 6 и 11), центральные частоты которых отстоят друг от друга на 25 МГц. Реализация режима Super-G возможна только на центральном канале 6.

Технология Super-G предусматривает два режима функционирования: динамический и статический. Статический режим предполагается использовать в WLAN на базе только оборудования Super-G, при этом включаются все функции Super-G, включая объединение двух каналов.

Динамический режим предполагается использовать в смешанных сетях WLAN, то есть когда имеются как клиенты Super-G, так и клиенты IEEE 802.11b/g. Поскольку клиенты IEEE 802.11b/g не поддерживают режима Super-G, то при обнаружении таких клиентов в сети при использовании динамического режима происходит автоматический переход работы всей сети на обычный режим IEEE 802.11b/g.

Кроме того, многие производители реализуют также и гибридный режим работы, когда технология Super-G используется без связывания каналов [2, 3].

Задание к проведению лабораторной работы.

1. Используя пакет NetCracker, изучить состав и функциональные характеристики типового оборудования беспроводных локальных сетей.

2. В соответствии с вариантом задания построить беспроводную сеть с использованием стандартов IEEE 802.11.

3. Для полученной модели сети задать необходимые типы потоков данных между рабочими станциями и серверами и произвести имитационное моделирование работы сети.

4. Проанализировать среднюю загрузку сетевого оборудования, количество теряемых пакетов.

5. Сделать выводы.

Таблица 42 - Исходные данные для лабораторной работы

№ Варианта	Тип архитектуры	Количество HTTP серверов	Количество FTP серверов	Количество беспроводных станций
1	Ad Hoc	2	1	4
2	Infrastructure Mode	3	2	5
3	Ad Hoc	2	3	3
4	Infrastructure Mode	1	4	4
5	Ad Hoc	3	1	4
6	Infrastructure Mode	4	2	4
7	Ad Hoc	3	3	5
8	Infrastructure Mode	2	4	4
9	Ad Hoc	4	1	2
10	Infrastructure Mode	1	2	5
11	Ad Hoc	4	3	5
12	Infrastructure Mode	2	4	3
13	Ad Hoc	1	1	6
14	Infrastructure Mode	2	2	4
15	Ad Hoc	2	3	3

Содержание отчета по лабораторной работе.

1. Название и цель работы.

2. Привести состав и функциональные характеристики типового оборудования беспроводных локальных сетей.

3. Представит схемные решения беспроводной сети с использованием стандартов IEEE 802.11.

4. Представить полученную модель сети.

5. Представить типы потоков данных между рабочими станциями и серверами.

6. Представить результаты имитационного моделирования работы сети.

7. Представить расчеты (результаты) средней загрузки сетевого оборудования и количества теряемых пакетов.

8. Выводы по выполненной работе.

9. Список использованных источников.

Контрольные вопросы.

1. Характеристика основных режимов работы беспроводных сетей.

2. Функции Mac-уровня в протоколах IEEE 802.11.

3. Способы доступа к передающей среде DCF и PCF.

4. Каким образом обеспечивается равноправный доступ абонентов к передающей среде в методе DCF.

5. В чем заключается проблема ложной блокировки узлов сети.

6. Как решается проблема скрытых узлов.

7. Какие способы повышения скорости передачи данных используются в беспроводных сетях.

18 Лабораторная работа № 18. Пересылка - прием сообщений через сокеты

Цель работы. Изучение особенностей использования сокета, для передачи сообщений в ЛВС.

Теоретическая справка.

Socket (гнездо, разъем) - абстрактное программное понятие, используемое для обозначения в прикладной программе конечной точки канала связи с коммуникационной средой, образованной вычислительной сетью. При использовании протоколов TCP/IP можно говорить, что socket является средством подключения прикладной программы к порту локального узла сети.

Рассмотрим механизм реализации сокетов в Borland Delphi. Для работы с сокетами в Delphi используются компоненты TClientSocket и TServerSocket. Они являются потомками абстрактного класса TabstractSocket, который включает методы и свойства, позволяющие прикладному приложению использовать Windows socket.

Windows socket объединяет в себе набор коммуникационных протоколов, предоставляющие возможность приложению подключаться к другим компьютерам для обмена информацией. Windows sockets поддерживает следующие семейства протоколов:

- TCP/IP;

- Xerox Network System (XNS);

- IPX/SPX;

- DECnet.

Сокеты позволяют приложению создавать соединение с другими машинами без знания конкретного типа протокола.

Для создания сокета, инициирующего соединение с другими машинами используют TclientSocket, а для создания сокета, отвечающего на запросы с других машин, - TserverSocket.

Примерная схема работы с сокетом клиента включает в себя следующие шаги:

1. Определение свойств сокета Host и Port. Host – это имя хост-имя или IP-адрес компьютера, с которым необходимо установить соединение. Port –имя порта.
2. Открытие сокета. В данном шаге сокет клиента определяет сервер и подключается к нему.
3. Пересылка данных.
4. Закрытие сокета.

Алгоритм работы сокета сервера немного отличается от рассмотренного выше алгоритма для сокета клиента:

1. Определение свойств Port и ServerType. Свойство Port аналогично свойству сокета клиента. ServerType – определяет тип подключения.
2. Открытие сокета. Сокет на данном шаге переходит в режим ожидания подключений клиентов.
3. Подключение клиентов и пересылка данных.
4. Отключение клиентов.
5. Закрытие сокета.

Порядок выполнения работы:

– изучить возможности сокетов для передачи данных в ЛВС;

– реализовать прикладное приложение на основе сокетов, обеспечивающее передачу сообщений по ЛВС;

– осуществите передачу сообщений между компьютерами, используя созданное прикладное приложение.

Пример работы программы.

1. Подключение клиента.

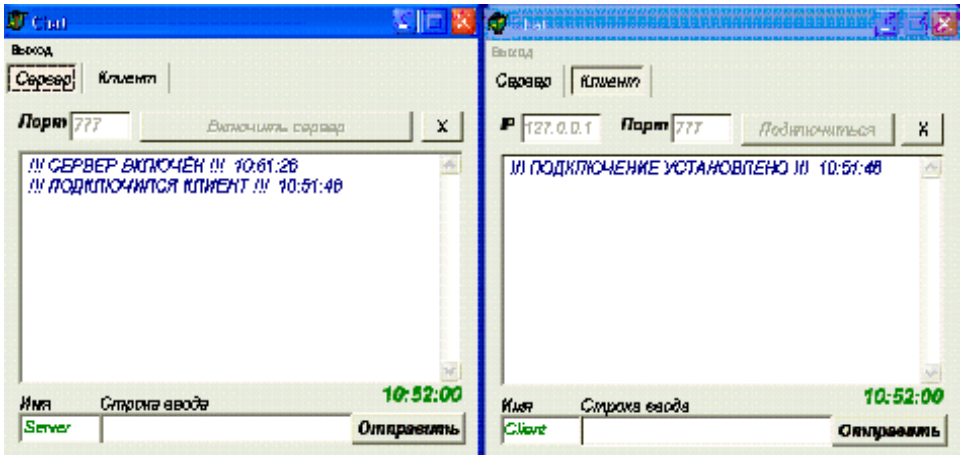


Рисунок 48 – Пример выполнения работы

2. Пересылка сообщения серверу и с сервера клиенту.

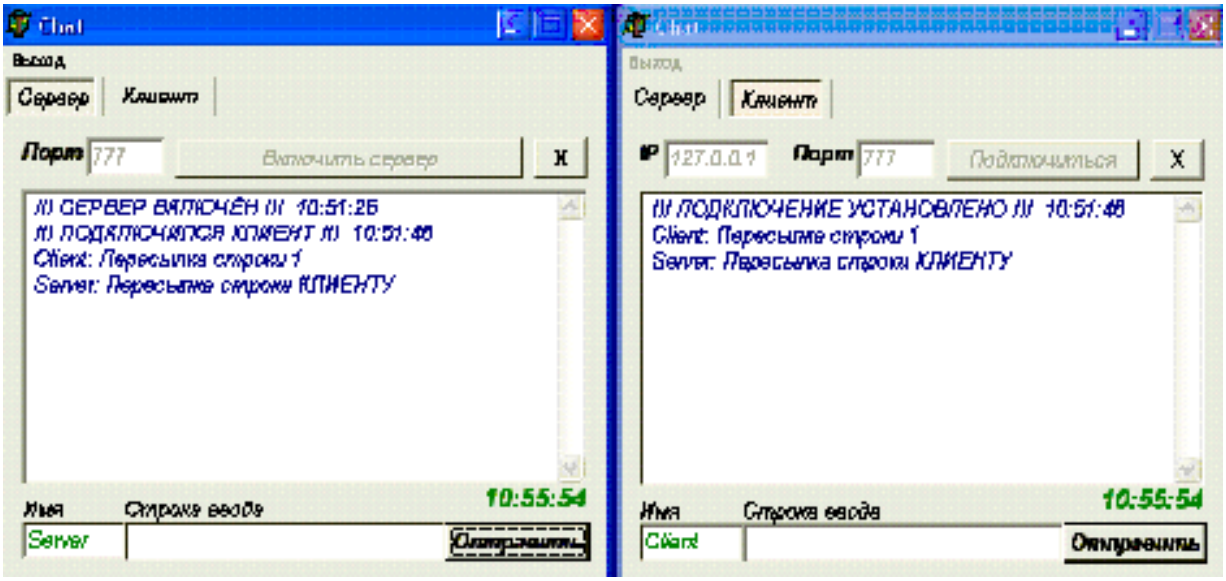


Рисунок 49 – Пример выполнения работы

3. Экранная форма программы по пересылке сообщений.

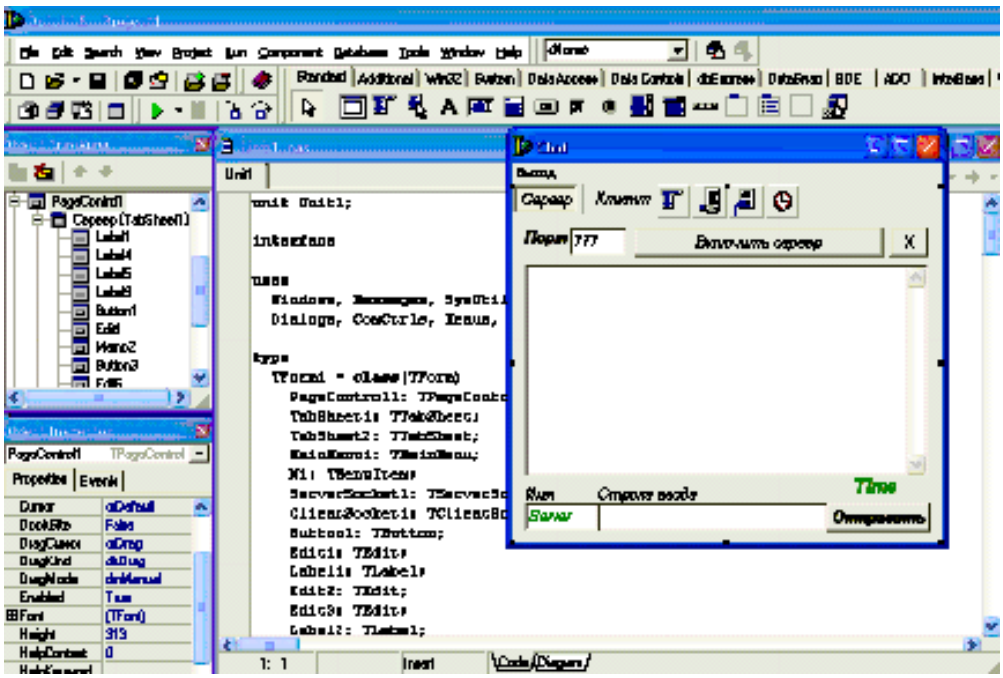


Рисунок 50 – Пример выполнения работы

Содержание отчета по лабораторной работе.

1. Название и цель работы.
2. Алгоритм и листинг программы.
3. Экранные формы работы программы.
4. Выводы по выполненной работе.
5. Список использованных источников.

Контрольные вопросы.

1. Что такое сокет?
2. Семейство каких протоколов поддерживает Windows socket?
3. Объясните и представьте алгоритм работы сокета для клиентского приложения?
4. Объясните и представьте алгоритм работы сокета для приложения сервера?
5. Объясните свойства, которые должны быть определены для создания соединения с помощью сокетов?

19 Лабораторная работа № 19. Пересылка - прием сложных данных через сокеты

Цель работы. Изучение особенностей использования сокета, для передачи сложных данных по ЛВС.

Теоретическая справка.

Сокеты позволяют передавать информацию различного вида. Данные передаются в виде последовательности символов, в результате можно передавать как текстовые сообщения, так и целые файлы.

Методов организации работы с сокетами в Delphi существует большое количество. Для передачи сложных данных между компьютерами нужно воспользоваться специальными операторами. Рассмотрим некоторые из них.

1. *SendBuf*(var Buf; Count: Integer) – метод передачи буфера через сокет. Вторым параметром Count метода указывается размер буфера в байтах.
2. *SendText*(const S: string) – текстовой строки.
3. *SendStream*(AStream: TStream) –передача потока через сокет. Поток в Delphi – это обобщенная модель двоичных данных, размещенных на устройствах-накопителях, таких как диски, ленточные накопители, оперативная память и т. п. Любой поток обладает ключевыми свойствами – размером в байтах (свойство Size) и текущей позицией (Position).

Всем данным методам передачи данных существуют соответствующие методы приема данных.

Выполнение работы.

– рассмотреть различные методы передачи сложных видов данных через сокеты;

– реализовать прикладное приложение на основе сокетов, обеспечивающее передачу файлов по ЛВС;

– осуществить передачу файлов различных размеров по ЛВС и проверить результат выполнения передачи данных.

Содержание отчета по лабораторной работе.

1. Название и цель работы.
2. Руководство для пользователя программы.
3. Руководство для программиста.
4. Алгоритм программы.
5. Листинг программы.
6. Экранные формы работы программы.
7. Вывод по выполненной работе.
8. Список использованных источников.

Контрольные вопросы.

1. Перечислите методы и опишите их характерные особенности (в зависимости от выбранного языка программирования), которые используются сокетами для передачи данных между компьютерами?
2. В каком виде передаются данные через сокеты?
3. Для чего была создана модель OSI?
4. Какие уровни включает модель OSI и охарактеризуйте каждый из них?
5. Алгоритм формирования сообщения для передачи данных по сети согласно модели OSI?
6. Как происходит обработка сообщения согласно модели OSI?

20 Лабораторная работа № 20. Компьютерные игры. Крестики - нолики

Цель работы. Получение навыков работы с протоколом TCP/IP, программирования сокетов, написание собственной игровой программы.

Теоретическая справка.

Компонента ServerSocket. Сервер, основанный на сокетном протоколе, позволяет обслуживать сразу множество клиентов. Для каждого подключенного

клиента сервер открывает отдельный сокет, по которому можно обмениваться данными с клиентом. Также возможно создание для каждого подключения отдельного процесса.

Определение свойств Port и ServerType необходимо, чтобы к серверу могли подключаться клиенты, нужно, чтобы порт, используемый сервером точно совпадал с портом, используемым клиентом (и наоборот). Свойство ServerType определяет тип подключения.

На этапе открытия сокета и указанного порта может выполняться автоматическое начало ожидания подсоединения клиентов (Listen).

При отключении клиента закрывается его сокетное соединение с сервером.

По команде приложения сервер завершает свою работу, закрывая все открытые сокетные каналы и прекращая ожидание подключений клиентов.

Свойство ServerType:TServerType указывает тип сервера. Оно может принимать одно из двух значений: stNonBlocking - синхронная работа с клиентскими сокетами. При таком типе сервера можно работать с клиентами через события OnClientRead и OnClientWrite. StThreadBlocking - асинхронный тип. Для каждого клиентского сокетного канала создается отдельный процесс (Thread).

Свойство Active: Boolean - показатель того, активен в данных момент сервер, или нет. Значение True указывает на то, что сервер работает и готов к приему клиентов, а False - сервер выключен. Чтобы запустить сервер, нужно просто присвоить этому свойству значение True.

Port: Integer это номер порта для установления соединений с клиентами. Значение порта у сервера и у клиентов должны быть одинаковыми. Рекомендуются значения от 1025 до 65535, т.к. от 1 до 1024 - могут быть заняты системой.

Метод Open запускает сервер. Эта команда идентична присвоению значения True свойству Active.

Метод Close останавливает сервер.

Событие OnClientConnect возникает, когда клиент установил сокетное соединение и ждет ответа сервера (OnAccept).

Событие OnClientDisconnect возникает, когда клиент отсоединился от сокетного канала.

Событие OnClientError возникает, когда текущая операция завершилась неудачно, т.е. произошла ошибка.

Событие OnClientRead возникает, когда клиент передал серверу какие-либо данные. Доступ к этим данным можно получить через передаваемый параметр Socket: TCustomWinSocket.

Событие OnClientWrite возникает, когда сервер может отправлять данные клиенту по сокету.

Событие OnAccept возникает, когда сервер принимает клиента или отказывает ему в соединении;

Событие OnListen возникает, когда сервер переходит в режим ожидания подсоединения клиентов.

Компонента ClientSocket. После назначения свойствам Host и Port соответствующих значений, можно приступить непосредственно к открытию сокета (сокет здесь рассматривается как очередь, в которой содержатся символы, передающиеся от одного компьютера к другому). Для этого можно вызвать метод Open компонента TClientSocket, либо присвоить свойству Active значение True.

Этап авторизации необходим, если сервер требует ввода логина и/или пароля. На этом этапе посылается серверу логин (имя пользователя) и пароль. Механизм авторизации зависит уже от конкретного сервера.

Свойство Host: string это строка, указывающая на хост-имя компьютера, к которому следует подключиться.

Address: string - строка, указывающая на IP-адрес компьютера, к которому следует подключиться. В отличие от Host, здесь может содержаться лишь IP. Отличие в том, что при указании в Host символьного имя компьютера, IP адрес, соответствующий этому имени, будет запрошен у DNS.

Строка Service : string, определяет службу (ftp, http, pop, и т.д.), к порту которой произойдет подключение. Это своеобразный справочник соответствия номеров портов различным стандартным протоколам.

Свойство ClientType это тип соединения. CtNonBlocking - асинхронная передача данных, т.е. посылать и принимать данные по сокету можно с помощью OnRead и OnWrite. CtBlocking - синхронная (одновременная) передача данных. События OnRead и OnWrite не работают. Этот тип соединения полезен для организации обмена данными с помощью потоков.

Событие OnConnect возникает при установлении соединения. Т.е. в обработчике этого события уже можно начинать авторизацию или прием/передачу данных.

Событие OnConnecting возникает при установлении соединения. Отличие от OnConnect в том, что соединение еще не установлено. Обычно такие промежуточные события используются для обновления статуса.

Событие OnDisconnect возникает при закрытии сокета.

Событие OnError возникает при ошибке в работе сокета. Следует отметить, что это событие не поможет отловить ошибку в момент открытия сокета (Open). Для того, чтобы избежать выдачи сообщения об ошибке, необходимо заключить операторы открытия сокета в блок try..except (обработка исключительных ситуаций).

Событие OnLookup возникает при попытке получения от DNS IP-адреса указанного хоста.

Событие OnRead возникает, когда удаленный компьютер послал какие-либо данные. При возникновении этого события возможна обработка данных.

Событие OnWrite возникает, когда разрешена запись данных в сокет.

Метод SendBuf(var Buf; Count: Integer) используется при посылке буфера через сокет. Буфером может являться любой тип, будь то структура (record), либо простая переменная типа Integer. Буфер указывается параметром Buf, вторым параметром необходимо указать размер пересылаемых данных в байтах (Count).

Метод SendText(const S: string) используется при посылке текстовой строки через сокет.

SendStream(AStream: TStream) это посылка содержимого указанного потока через сокет. Пересылаемый поток должен быть открыт. Поток может быть любого типа – файловый или из оперативной памяти.

Ход работы.

1. Установить сетевое соединение двух компьютеров.

2. После установки соединения у каждого из игроков отображается игровое поле (рисунок 51).

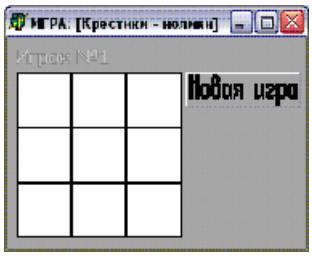


Рисунок 51 – Игровое поле

3. Игра. На основе договоренности первого хода (т.е. заранее решено, кто будет делать первый ход: «сервер» или «клиент») игрок №1 делает первый ход – посылает информацию о нажатой игровой клетке. Далее игрок №2 выбирает одно из оставшихся игровых полей. Игра продолжается до тех пор пока на произойдет одно из двух возможных действий:

– на одной линии (по горизонтали, вертикали или диагонали) будут «выстроены» крестики или нолики (рисунок 52), в этом случае должно выводиться сообщение (рисунок 53);

Рисунок 52 – Выигрыш одного из игроков

– не останется ни одной свободной клетки.

Рисунок 53 – Сообщение о выигрыше

4. Выход из программы или новая игра (возврат к пункту 2).

Содержание отчета по лабораторной работе.

1. Название и цель работы.
2. Руководство для пользователя программы.
3. Руководство для программиста.
4. Алгоритм программы.
5. Листинг программы.
6. Экранные формы работы программы.
7. Вывод по выполненной работе.
8. Список использованных источников.

Контрольные вопросы.

1. Перечислите методы и опишите их характерные особенности, которые используются сокетами для передачи данных между компьютерами.
2. Какие компоненты использовались при построении программы?
3. В каком виде передаются данные через сокеты?
4. Стек TCP/IP.
5. Модель OSI.

21 Лабораторная работа № 21. Компьютерные игры. Морской бой

Цель работы. Получение навыков работы с протоколом TCP/IP, программирования сокетов, написание собственного протокола прикладного уровня OSI.

Теоретическая справка.

Компонента ServerSocket Сервер, основанный на сокетном протоколе, позволяет обслуживать сразу множество клиентов. Для каждого подключенного клиента сервер открывает отдельный сокет, по которому можно обмениваться данными с клиентом. Также возможно создание для каждого подключения отдельного процесса.

Определение свойств Port и ServerType необходимо, чтобы к серверу могли подключаться клиенты, нужно, чтобы порт, используемый сервером точно совпадал с портом, используемым клиентом (и наоборот). Свойство ServerType определяет тип подключения.

На этапе открытия сокета и указанного порта может выполняться автоматическое начало ожидания подсоединения клиентов (Listen).

При отключении клиента закрывается его сокетное соединение с сервером.

По команде приложения сервер завершает свою работу, закрывая все открытые сокетные каналы и прекращая ожидание подключений клиентов.

Свойство ServerType:TServerType указывает тип сервера. Оно может принимать одно из двух значений: stNonBlocking - синхронная работа с клиентскими сокетами. При таком типе сервера можно работать с клиентами через события OnClientRead и OnClientWrite. StThreadBlocking - асинхронный тип. Для каждого клиентского сокетного канала создается отдельный процесс (Thread).

Свойство Active: Boolean - показатель того, активен в данных момент сервер, или нет. Значение True указывает на то, что сервер работает и готов к приему клиентов, а False - сервер выключен. Чтобы запустить сервер, нужно просто присвоить этому свойству значение True.

Port: Integer это номер порта для установления соединений с клиентами. Значение порта у сервера и у клиентов должны быть одинаковыми. Рекомендуются значения от 1025 до 65535, т.к. от 1 до 1024 - могут быть заняты системой.

Метод Open запускает сервер. Эта команда идентична присвоению значения True свойству Active.

Метод Close останавливает сервер.

Событие OnClientConnect возникает, когда клиент установил сокетное соединение и ждет ответа сервера (OnAccept).

Событие OnClientDisconnect возникает, когда клиент отсоединился от сокетного канала.

Событие OnClientError возникает, когда текущая операция завершилась неудачно, т.е. произошла ошибка.

Событие OnClientRead возникает, когда клиент передал серверу какие-либо данные. Доступ к этим данным можно получить через передаваемый параметр Socket: TCustomWinSocket.

Событие OnClientWrite возникает, когда сервер может отправлять данные клиенту по сокету.

Событие OnAccept возникает, когда сервер принимает клиента или отказывает ему в соединении;

Событие OnListen возникает, когда сервер переходит в режим ожидания подсоединения клиентов.

Компонента ClientSocket.

После назначения свойствам Host и Port соответствующих значений, можно приступить непосредственно к открытию сокета (сокет здесь рассматривается как очередь, в которой содержатся символы, передающиеся от одного компьютера к другому). Для этого можно вызвать метод Open компонента TClientSocket, либо присвоить свойству Active значение True.

Этап авторизации необходим, если сервер требует ввода логина и/или пароля. На этом этапе посылается серверу логин (имя пользователя) и пароль. Механизм авторизации зависит уже от конкретного сервера.

Свойство Host: string это строка, указывающая на хост-имя компьютера, к которому следует подключиться.

Address: string - строка, указывающая на IP-адрес компьютера, к которому следует подключиться. В отличие от Host, здесь может содержаться лишь IP. Отличие в том, что при указании в Host символьного имя компьютера, IP адрес, соответствующий этому имени, будет запрошен у DNS.

Строка Service : string, определяет службу (ftp, http, pop, и т.д.), к порту которой произойдет подключение. Это своеобразный справочник соответствия номеров портов различным стандартным протоколам.

Свойство ClientType это тип соединения. CtNonBlocking - асинхронная передача данных, т.е. посылать и принимать данные по сокету можно с помощью OnRead и OnWrite. CtBlocking - синхронная (одновременная) передача данных. События OnRead и OnWrite не работают. Этот тип соединения полезен для организации обмена данными с помощью потоков.

Событие OnConnect возникает при установлении соединения. Т.е. в обработчике этого события уже можно начинать авторизацию или прием/передачу данных.

Событие OnConnecting возникает при установлении соединения. Отличие от OnConnect в том, что соединение еще не установлено. Обычно такие промежуточные события используются для обновления статуса.

Событие OnDisconnect возникает при закрытии сокета.

Событие OnError возникает при ошибке в работе сокета. Следует отметить, что это событие не поможет отловить ошибку в момент открытия сокета (Open). Для того, чтобы избежать выдачи сообщения об ошибке, необходимо заключить операторы открытия сокета в блок try..except (обработка исключительных ситуаций).

Событие OnLookup возникает при попытке получения от DNS IP-адреса указанного хоста.

Событие OnRead возникает, когда удаленный компьютер послал какие-либо данные. При возникновении этого события возможна обработка данных.

Событие OnWrite возникает, когда разрешена запись данных в сокет.

Метод SendBuf(var Buf; Count: Integer) используется при посылке буфера через сокет. Буфером может являться любой тип, будь то структура (record), либо простая переменная типа Integer. Буфер указывается параметром Buf, вторым параметром необходимо указать размер пересылаемых данных в байтах (Count).

Метод SendText(const S: string) используется при посылке текстовой строки через сокет.

SendStream(AStream: TStream) это посылка содержимого указанного потока через сокет. Пересылаемый поток должен быть открыт. Поток может быть любого типа – файловый или из оперативной памяти.

Порядок выполнения работы.

1. Установить сетевое соединение двух компьютеров.

2. Подготовка к игре. Каждый из игроков расставляет на игровом поле (10x10 клеток) корабли (четыре корабля размером в одну клетку, три – размером в две, два корабля – в три клетки и один корабль - 4 клетки (рисунок 54) , в соответствии с правилами игры (корабли не должны располагаться на смежных клетках).

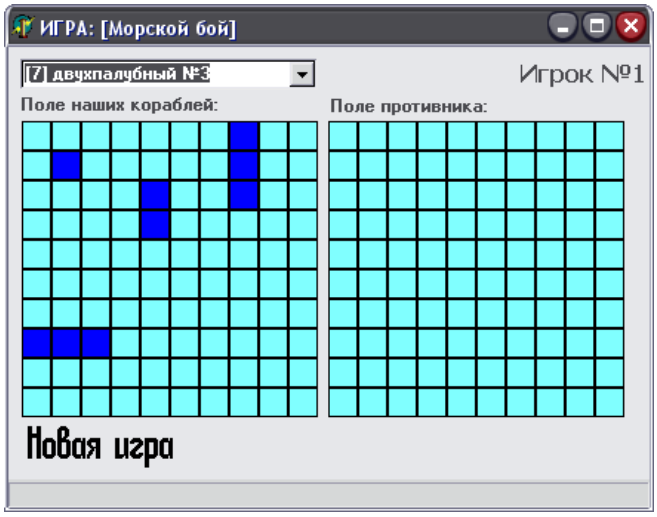


Рисунок 54 – Размещение кораблей на игровом поле

3. Игра. На основе договоренности первого хода (т.е. заранее решено, кто будет делать первый ход: «сервер» или «клиент») игрок №1 делает первый ход – посылает запрос о какой-либо клетке игрового поля на предмет наличия в ней корабля противника. Приложение оппонента (игрок №2) обрабатывает запрос и посылает данные обратно игроку №1. В присланных данных может содержаться два типа сообщения: либо в данной клетке есть корабль (т.е. «попал» или «убил» - в этом случае игрок №1 ходит еще раз и данная клетка помечается «крестиком»), либо в данной клетке нет корабля (т.е. «мимо» - в этом случае игрок №1 теряет право

хода и передает ход игроку №2. Игрок №2 посылает аналогичный запрос игроку №1 (рисунок 55).

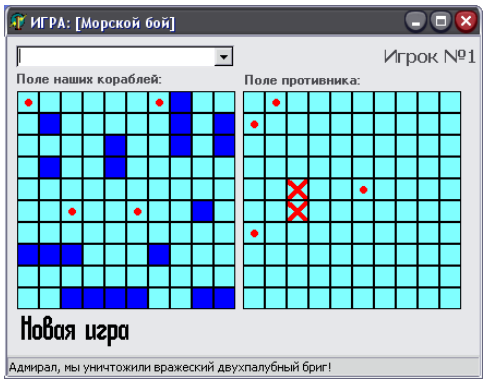


Рисунок 55 - Игра

Игра заканчивается в том случае, когда «потоплены» корабли одного из игроков.

4. Выход из программы или новая игра (возврат к пункту 2).

Содержание отчета по лабораторной работе.

1. Название и цель работы.
2. Руководство для пользователя программы.
3. Руководство для программиста.
4. Алгоритм программы.
5. Листинг программы.
6. Экранные формы работы программы.
7. Вывод по выполненной работе.
8. Список использованной источников.

Контрольные вопросы.

1. Перечислите методы и опишите их характерные особенности, которые используются сокетами для передачи данных между компьютерами?
2. Какие компоненты использовались при построении программы?
3. В каком виде передаются данные через сокеты?
4. Стек TCP/IP?
5. Модель OSI?

22 Лабораторная работа № 22. Аутентификация в компьютерных сетях

Цель работы. Разработать способы аутентификации в компьютерных сетях.

Теоретическая справка.

Средства аутентификации и идентификации относятся к категории классических средств по управлению информационной безопасностью как корпоративных, так и глобальных коммуникационных сетей и включают в себя определение, создание, изменение, удаление и аудит пользовательских учетных записей. Аутентификация в них используется для проверки подлинности входящего в систему пользователя и для избежания отказа в обслуживании зарегистрированного пользователя

Для эффективного построения распределенных информационных технологий необходимо участие пользователя в функциях, выполняемых в распределенных устройствах, часто удаленных от места положения самого пользователя. В связи с этим встает задача идентификации и аутентификации пользователей в различных компонентах распределенной системы и программной инфраструктуры в зависимости от выполняемых функций.

Чтобы обеспечить безопасность информационных ресурсов, устранить возможность несанкционированного доступа, усилить контроль санкционированного доступа к конфиденциальной либо к подлежащей засекречиванию информации, внедряются различные системы опознавания, установления подлинности объекта (субъекта) и разграничения доступа. В основу построения таких систем закладывается принцип допуска и выполнения только таких обращений к информации, в которых присутствуют соответствующие признаки разрешенных полномочий.

Ключевыми понятиями в этой системе являются "идентификация" и "аутентификация". Идентификация - это присвоение какому-либо объекту или субъекту уникального имени или образа. Аутентификация - это установление

подлинности, т.е. проверка, является ли объект (субъект) действительно тем, за кого он себя выдает.

Конечная цель процедур идентификации и аутентификации объекта (субъекта) - допуск его к информации ограниченного пользования в случае положительной проверки либо отказ в допуске в случае отрицательного исхода проверки.

Объектами идентификации и аутентификации могут быть люди (пользователи, операторы и др.), технические средства (мониторы, рабочие станции, абонентские пункты), документы (ручные, распечатки и др.), магнитные носители информации, информация на экране монитора, табло и др.

Один из наиболее распространенных методов аутентификации - присвоение лицу или другому имени пароля и хранение его значения в вычислительной системе. Пароль - это совокупность символов, определяющая объект (субъект). При выборе пароля возникают вопросы о его размере, стойкости к несанкционированному подбору, способам его применения. Естественно, чем больше длина пароля, тем большую безопасность будет обеспечивать система, ибо потребуются большие усилия для его отгадывания. При этом выбор длины пароля в значительной степени определяется развитием технических средств, их элементной базой и быстродействием.

Для идентификации пользователей могут применяться сложные в плане технической реализации системы, обеспечивающие установление подлинности пользователя на основе анализа его индивидуальных параметров: отпечатков пальцев, рисунка линий руки, радужной оболочки глаз, тембра голоса и др. Но пока эти приемы носят скорее рекламный, чем практический характер.

Одно из интенсивно разрабатываемых направлений по обеспечению безопасности информации - идентификация и установление подлинности документов на основе электронной цифровой подписи - ныне простирается от проведения финансовых и банковских операций до контроля за выполнением различных договоров. Естественно, при передаче документов по каналам связи применяется факсимильная аппаратура, но в этом случае к получателю приходит не подлинник, а лишь копия документа с копией подписи, которая в процессе передачи

может быть подвергнута повторному копированию для использования ложного документа.

Порядок выполнения работы.

1. Изучить средства аутентификации и идентификации.

2. Создать сетевое приложение, использующее средства аутентифика-ции и идентификации пользователей («чат», службу SMS- сообщений, FTP-сервер и т. д.).

3. Предоставлять доступ к ресурсу распределенной вычислительной сети.

4. Приложение должно предусматривать регистрацию пользователей.

5. Осуществлять проверку на право доступа к данному ресурсу.

6. Провести тестирование приложения в ЛВС.

Содержание отчета по лабораторной работе.

1.Название и цель работы;

2.Руководство для пользователя программы;

3.Руководство для программиста;

4.Алгоритм программы;

5.Листинг программы;

6.Экранные формы работы программы;

7.Вывод по выполненной работе;

8.Список использованных источников.

Контрольные вопросы.

1. Что такое аутентификация?

2. Что такое идентификация?

3. Какие методы аутентификации и идентификации существуют?

23 Лабораторная работа № 23. Снифферы. Переключение сетевого адаптера в режим прослушивания

Цель работы. Практическое изучение протокола TCP/IP. Получение навыков программирования сетевого уровня модели семиуровневой модели OSI.

Теоретическая справка.

Сниффинг в локальной сети без коммутаторов - хорошо проработанная технология. Большое количество коммерческих и некоммерческих утилит делает возможным прослушивание сетевого трафика и извлечение необходимой информации. Идея заключается в том, что для прослушивания сетевого трафика, сетевая карта компьютера переводится в специальный режим "promisc mode". После этого весь сетевой трафик (несмотря на его предназначение), достигший сетевой карты, может быть доступен снифферу.

В локальной сети с коммутаторами для прослушивания сетевого трафика потребуется больше изобретательности, поскольку коммутатор направляет только тот трафик, который предназначен для конкретного компьютера. Однако, существует ряд технологий, которые позволяют преодолеть это ограничение.

Библиотека packet.dll предоставляет набор функций, которые позволяют принять или отправить пакет произвольной структуры, запросить или установить параметры сетевого адаптера, получить дескрипторы динамически размещаемых структур типа PACKET, установить или снять BPF-фильтр, изменить размер буфера драйвера и получить статистическую информацию о текущей сессии.

Имеются следующие функции.

1. ULONG PacketGetAdapterNames (PTSTR pStr, PULONG BufferSize) – предназначена для получения информации об адаптерах, установленных в системе. Функция опрашивает регистр ОС, производит OID-вызовы драйвера пакетов и записывает имена установленных сетевых адаптеров и их описание в заданный пользователем буфер pStr. BufferSize – размер этого буфера. Формат данных, записываемых в буфер, отличен для версий Windows 95/98 и WindowsNT/2000, из-за разницы в кодировках строк у этих ОС (Windows 95/98 использует кодировку ASCII, Windows NT/2000 – UNICODE).

2. LPADAPTER PacketOpenAdapter (LPSTR AdapterName) – предназ-начена для инициализации адаптера. Функции передается имя адаптера в качестве аргумента AdapterName (получено с помощью PacketGetAdapterNames), результатом функции является указатель на структуру ADAPTER открытого адаптера.

3. VOID PacketCloseAdapter (LPADAPTER lpAdapter) – высвобождает структуру ADAPTER, связанную с указателем lpAdapter, и закрывает адаптер, связанный с ней.

4. LPPACKET PacketAllocatePacket (void) – определяет положение структуры PACKET, инициализированной функцией PacketInitPacket, и возвращает указатель на нее.

5. VOID PacketInitPacket (LPPACKET lpPacket, PVOID Buffer, UINT Length) – инициализирует структуру PACKET и имеет следующие аргументы:

– lpPacket – указатель на инициализируемую структуру;

– Buffer – указатель на буфер, задаваемый пользователем и содержащий данные пакета;

– Length – длина буфера – максимальный размер данных, которые могут быть переданы драйвером приложению за один сеанс чтения.

6. VOID PacketFreePacket (LPPACKET lpPacket) – высвобождает структуру PCAKET, связанную с указателем lpPacket.

7. VOID PacketReceivePacket (lpAdapter AdapterObject, LPPACKET lpPacket, BOOLEAN Sync) – выполняет захват группы пакетов, и имеет следующие аргументы:

– AdapterObject – указатель на структуру ADAPTER, определяющую адаптер, который будет задействован в текущей сессии;

– lpPacket – указатель на структуру PACKET, используемую для записи принятых пакетов;

– Sync – флаг, определяющий режим выполнения операции.

Если выбран синхронный режим (True), функция блокирует программу до завершения операции. Если выбран асинхронный режим (False), блокировки не происходит. В последнем случае необходимо использовать функцию PacketWaitPaket для корректного выполнения операции.

Число принятых пакетов зависит от количества пакетов, сохраненных в буфере драйвера, размера этих пакетов и размера буфера, связанного со структурой lpPacket. Формат передачи данных приложению драйвером приведен на рисунке 56.

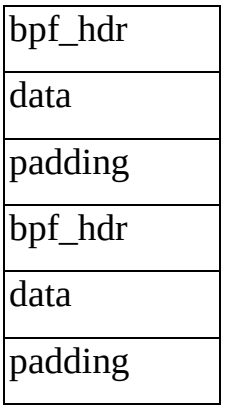


Рисунок 56 - Формат передачи данных приложению драйвером

Пакеты сохраняются в буфере структуры lpPacket. Каждый пакет имеет трейлер, состоящий из структуры bpf_hdr и содержащий информацию о длине пакета и времени его приема. Поле Padding используется для выравнивания данных в буфере. Поля bf_datalen и bf_hdrlen структуры bpf_hdr используются для извлечения пакетов из буфера. Заметим, что Pcap извлекает каждый пакет до того, как передать его приложению.

1. BOOLEAN PacketSetHwFilter (LPADAPTER AdapterObject, ULONG Filter) – устанавливает аппаратный (hardware) фильтр входящих пакетов. Константы, с помощью которых задается фильтр, объявлены в файле ntddndis.h. В качестве аргументов функции задается адаптер, на который устанавливается фильтр, и идентификатор фильтра. Функция возвращает значение True, если операция выполнена успешно. Ниже перечислены наиболее часто используемые фильтры:

2. NDIS_PACKET_TYPE_PROMISCUOUS - каждый входящий пакет принимается адаптером.

3. NDIS_PACKET_TYPE_DIRECTED - принимаются пакеты, предназначенные для данной рабочей станции.

4. NDIS_PACKET_TYPE_BROADCAST - принимаются только широковещательные запросы.

5. NDIS_PACKET_TYPE_MULTICAST - принимаются пакеты, предназначенные группе, которой принадлежит рабочая станция.

6. NDIS_PACKET_TYPE_ALL_MULTICAST: принимаются пакеты любой группы.

7. BOOLEAN PacketSetBuff (LPADAPTER AdapterObject, int dim) - устанавливает новый размер буфера драйвера, связанного с адаптером AdapterObject.dim – новый размер буфера. Функция возвращает True, если операция была выполнена успешно, False – если для выполнения операции недостаточно памяти. При установке нового размера буфера все данные, находящиеся в нем, стираются.

Порядок выполнения работы.

В данной лабораторной работе необходимо выполнить несколько последовательных действий.

1. Необходимо определить текущий сетевой адаптер.
2. При помощи нажатия какой-либо клавиши/кнопки приложения, переключить адаптер в режим «прослушивания».
3. Сохранить несколько пакетов переданных по сети в файл.
5. При помощи другой клавиши/кнопки вернуть исходный режим работы сетевой карты.
6. Обеспечить возможность просмотра сохраненного лога (log) пакетов.

Содержание отчета по лабораторной работе.

1. Название и цель работы.
2. Руководство для пользователя программы.
3. Руководство для программиста.
4. Алгоритм программы.
5. Листинг программы.
6. Экранные формы работы программы.
7. Вывод по выполненной работе.
8. Список использованных источников.

24 Лабораторная работа № 24. Анализ работы вычислительной сети

Цель работы. Практическое изучение работы локальной сети, выявление слабых мест и загруженности сети.

Теоретическая справка.

Данное программное обеспечение предназначено для анализа работы локальной сети, выявления слабых мест и загруженности сети.

Программа может использоваться в различных сетях независимо от топологии с количеством клиентских машин до 100, операционная система Windows 9x/Me/NT/XP.

Программа производит сравнительный анализ сетевого трафика и ведет статистку подключений и загрузки сети. Имеет удобный интерфейс, вся аналитическая информация выводится в графическом виде.

Программа может использоваться не только в локальных сетях, но также и в глобальной сети Internet.

Данное программное обеспечение состоит из двух взаимосвязанных программных модулей: клиентской части и серверной программы.

Клиентская программа устанавливается на все компьютеры сети, которые будут анализироваться. Соответственно серверная программа устанавливается на машину с которой будет анализироваться работа сети.

Клиентская программа практически не нуждается в настройке. В файле serv.ini хранится IP адрес сервера. При запуске программа автоматически пытается подключиться по данному адресу. В случае успешного подключения просто сворачивается на панель задач, иначе выдает сообщение об ошибке.

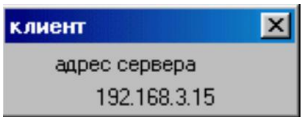


Рисунок 57 – Внешний вид клиентской программы

Серверная программа работает как анализатор тех данных, которые присылают клиентские программы. Сразу после запуска она ожидает сообщения от клиентов, и после получения немедленно реагирует и выводит результаты на экран.

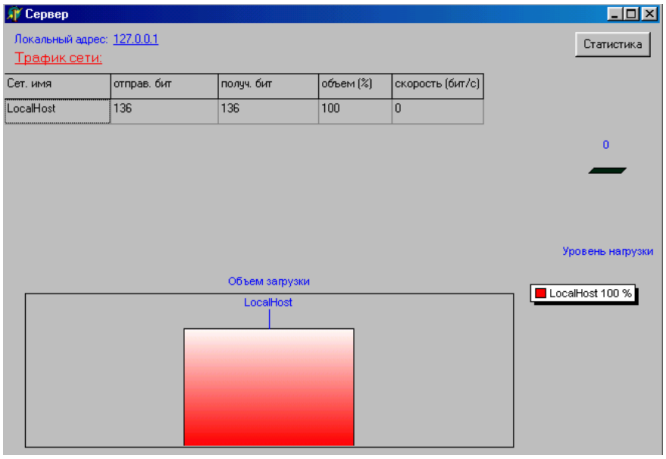


Рисунок 58 – Внешний вид серверной программы

Вверху выводится IP адрес машины, где установлена серверная программа. В таблице выводится список всех подключенных машин и их сетевой трафик. Нижняя диаграмма показывает сравнительную загрузку сети каждой из машин, автоматически присваивая каждой определенный цвет, сбоку от диаграммы выводится расшифровка к каждой колонке диаграммы.

Справа находится индикатор загрузки, каждую секунду показывающий уровень нагрузки на сеть.

Для каждой подключившейся машины можно получить дополнительную информацию, не отображающуюся в таблице подключений. Для этого достаточно выбрать нужного клиента в таблице и дважды щелкнуть левой клавишей мыши по соответствующей строке. При этом высвечивается: IP адрес клиента, время последнего подключения, если на данный момент клиент уже отключился, то время отключения и текущее состояние.

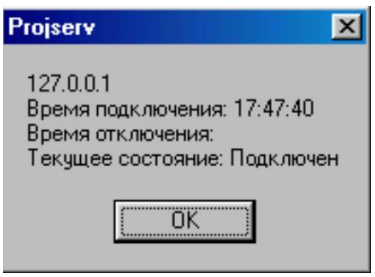


Рисунок 59 – Получение дополнительной информации о клиенте

На протяжении всей работы программа ведет статистику работы сети.

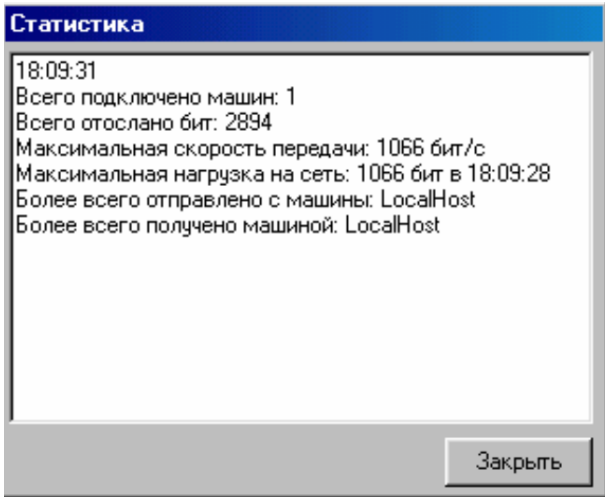


Рисунок 60 – Получение статистической информации.

При этом выводится статистическая информация на текущее время на машине, где установлена серверная программа.

При обрыве связи с одним из клиентов информация о нем не удаляется из списка подключений, но свойство «текущее состояние» становится равным «Отключено». При повторном подключении клиента его свойства просто обновляются.

Клиентская программа (описание процедур и функций).

Основная функция клиентской программы это определение входящего и исходящего трафика локальной машины. Для этого достаточно использовать всего лишь одну функцию библиотеки IPHLPAPI.DLL, которая поставляется со всеми версиями Windows. Рассмотрим ее:

Объявление функции ():

```
var
    GetIfTable:function( pIfTable: PMibIfTable;
                         pdwSize      : PULONG;
                         bOrder  : Boolean ): DWORD; stdcall;
```

Параметры:

pIfTable - должен содержать указатель на структуру;

pdwSize - должен содержать размер структуры;

bOrder - указывает, нужна ли сортировка в возвращаемом массиве.

В качестве первого параметра функция использует указатель на структуру.

Описание структуры.

```
type
 TMibIfTable = packed record
   dwNumEntries  : DWORD;
   Table             : TMibIfArray;
 end;
 PMibIfTable = ^ TMibIfTable;
```

Поля:

– dwNumEntries - определяет размерность массива представленного вторым параметром;

– Table - является массивом структур.

Структура сама по себе крайне неинформативна, нас интересует второе ее поле, также представляющее собой структуру.

```
type
  TMibIfRow = packed record
    wszName             : array[0..255] of WideChar;
    dwIndex             : DWORD;
    dwType              : DWORD;
    dwMtu               : DWORD;
    dwSpeed             : DWORD;
```

```
    dwPhysAddrLen       : DWORD;
    bPhysAddr           : array[0..7] of Byte;
    dwAdminStatus       : DWORD;
    dwOperStatus        : DWORD;
    dwLastChange        : DWORD;
    dwInOctets          : DWORD;
    dwInUcastPkts       : DWORD;
    dwInNUCastPkts      : DWORD;
    dwInDiscards        : DWORD;
    dwInErrors          : DWORD;
    dwInUnknownProtos   : DWORD;
    dwOutOctets         : DWORD;
    dwOutUCastPkts      : DWORD;
    dwOutNUCastPkts     : DWORD;
    dwOutDiscards       : DWORD;
    dwOutErrors         : DWORD;
    dwOutQLen           : DWORD;
    dwDescrLen          : DWORD;
    bDescr              : array[0..255] of Char;
  end;
  TMibIfArray = array [0..512] of TMibIfRow;
  PMibIfRow = ^TMibIfRow;
  PmibIfArray = ^TmibIfArray;
```

Поля:

- wszName - указатель на строку содержащую имя интерфейса;
- dwIndex - определяет индекс интерфейса;
- dwType - определяет тип интерфейса (смотри MSDN);
- dwMtu - определяет максимальную скорость передачи;
- dwSpeed - определяет текущую скорость передачи в битах в секунду;
- dwPhysAddrLen - определяет длину адреса содержащегося в bPhysAddr;

– bPhysAddr - содержит физический адрес интерфейса (его немного видоизмененный, MAC адрес);

– dwAdminStatus - определяет активность интерфейса;

– dwOperStatus - содержит текущий статус интерфейса (смотри MSDN);

– dwLastChange - содержит последний измененный статус;

– dwInOctets - содержит количество байт принятых через интерфейс;

– dwInUcastPkts - содержит количество направленных пакетов принятых интерфейсом;

– dwInNUCastPkts - содержит количество ненаправленных пакетов принятых интерфейсом (включая Броадкаст и т.п.);

– dwInDiscards - содержит количество забракованных входящих пакетов (даже если они не содержали ошибки);

– dwInErrors - содержит количество входящих пакетов содержащих ошибки;

– dwInUnknownProtos - содержит количество забракованных входящих пакетов со структурой неизвестного протокола;

– dwOutOctets - содержит количество байт отправленных интерфейсом;

– dwOutUCastPkts - содержит количество направленных пакетов отправленных интерфейсом;

– dwOutNUCastPkts - содержит количество ненаправленных пакетов отправленных интерфейсом (включая Броадкаст и т.п.);

– dwOutDiscards - содержит количество забракованных исходящих пакетов (даже если они не содержали ошибки);

– dwOutErrors - содержит количество исходящих пакетов содержащих ошибки

– dwOutQLen - содержит длину очереди данных;

– dwDescrLen - содержит размер массива bDescr;

– bDescr - содержит описание интерфейса.

По MSDN интерфейсом является не обязательно какое-либо физическое устройство, например сетевая карта, но также и сетевые службы.

Передача данных по сети осуществляется при помощи сокетов.

При подключении клиентская машина отсылает серверу код #1, при отключении - код #2.

На форме находится элемент **«Таймер»**. Каждые полсекунды по событию таймера клиентская программа проверяет переменные со значениями количества полученных и отосланных бит и если хотя бы одно значение изменилось, то отсылает серверу строку следующего формата:

«*Код*»#«*всего получено бит*»#«*всего отправлено бит*»

Проверка сделана для того, чтобы не загружать сеть сообщениями о нулевом трафике.

Клиентская программа автоматически подключается к серверу при запуске программы, используя в качестве адреса сервера информацию из файла serv.ini и сворачивается на панель задач.

Серверная программа (описание процедур и функций).

Основное назначение серверной программы – анализ тех данных, которые присылают клиентские программы.

Для хранения присланных данных используется массив:

hosts: array [1..100,1..9] of string;

в котором хранятся - сетевое имя клиентской машины, количество отправленных бит машины, количество полученных бит машины, скорость передачи, время подключения, время отключения, IP адрес, текущее состояние.

Для отображения данных используется компонент таблица **StringGrid**. Когда сервер получает очередной пакет данных от одной из клиентских программ происходит обработка полученных данных: в первую очередь проверяется от какой машины пришли данные и есть ли она в списке подключенных машин, если есть, то обновляется соответствующая графа массива, если нет, то данные от машины и информация о ней добавляется в массив и параметр количества подключенных машин увеличивается на единицу.

Данные от машины анализируются в соответствии с шаблоном:

«*Код*»#«*всего получено бит*»#«*всего отправлено бит*»

Информация о сетевом имени и IP адресе машины берется из сокета по соответствующим свойствам RemoteHost и RemoteAddress.

Определяется процент загрузки сети данной машиной в соотношении с общим объемом загрузки. При этом объем загрузки сети конкретной машиной берется.

1. Объем - количество отправленных бит в данный момент времени и количество полученных бит в данный момент времени. В программе используется компонент **«Таймер»**. По событию **OnTimer** происходит анализ скорости передачи (бит/сек) каждой машины.

2. Скорость - объем передачи в данный момент времени - объем передачи секунду назад.

Для графического отображения данных используются компоненты диаграммы. Диаграмма **Chart1** используется для отображения объема загрузки сети каждой машиной в соотношении с другими машинами. При этом каждой колонке диаграммы соответствует объем загрузки конкретной машиной. Обновление диаграммы происходит каждый раз, как приходит пакет данных от какой либо машины.

Диаграмма **Chart2** служит для отображения общего уровня загрузки сети в данный момент времени. Колонке диаграммы соответствует общее количество отправленных и полученных бит за последнюю секунду. Обновление происходит по событию таймера **OnTimer**, при этом максимальным уровнем считается максимальный уровень за все время измерения, если общее количество отправленных и полученных бит за последнюю секунду превышает максимальный уровень, то это значение в дальнейшем будет считаться максимальным уровнем.

Дополнительные свойства каждого клиента выводятся в виде сообщения при двойном нажатии левой клавиши мыши на соответствующем элементе таблицы клиентов. При этом последовательно обрабатываются события таблицы **OnSelectCell** и **OnDblClick**.

В дополнительных свойствах клиента указывается - IP адрес клиента, время подключения, время отключения, текущее состояние.

На протяжении всей работы программы ведется подключений и работы сети. Статистические данные обновляются каждую секунду по событию таймера.

Вывод статистической информации осуществляется при нажатии кнопки «Статистика», при этом обрабатывается событие **Button1Click.**

В статистике указывается: общее количество подключенных машин, общее количество пересланных бит, максимальная скорость передачи (бит/с), максимальная нагрузка на сеть и время когда это произошло, имя машины принявшей наибольшее количество данных и имя машины отправившей наибольшее количество данных.

Основные компоненты клиентской программы.

ClientSocket1: TclientSocket – сокет клиента. Основной компонент для передачи информации серверной программе.

tmrTraffic : Ttimer – таймер с интервалом обновления 0.5 сек. Используется для периодического получения информации с сетевого интерфейса.

Label2: Tlabel – надпись. Используется для вывода информации о сетевом адресе сервера.

Основные компоненты серверной программы.

ServerSocket1: TserverSocket – сокет сервера. Сетевой компонент для получения информации от клиентских машин.

Timer1: Ttimer – таймер с интервалом обновления 1 сек. Используется для периодической обработки информации, полученной от клиентских машин.

net: TstringGrid – таблица. Используется для вывода информации о подключившихся машинах.

Label2: Tlabel – надпись. Используется для вывода сетевого адреса сервера.

Chart1: Tchart – диаграмма. Используется для графического вывода объема сравнительной загрузки сети. Диаграмма настроена таким образом, что каждой колонке автоматически присваивается индивидуальный цвет, а справа выводится комментарии к каждой и значение в проценнтном представлении.

Chart2: Tchart - диаграмма. Используется для индикации уровня загрузки сети. В свойстве LeftAxis.Maximum указывается максимальный уровень за все время работы программы.

Button1: Tbutton – кнопка. Предназначена для вывода статистической информации.

Содержание отчета по лабораторной работе.

1. Название и цель работы;
2. Руководство для пользователя программы;
3. Руководство для программиста;
4. Алгоритм программы;
5. Листинг программы;
6. Экранные формы работы программы;
7. Вывод по выполненной работе;
8. Список использованных источников.

Список использованных источников

1 Олифер, В. Г. Компьютерные сети. Принципы, технологии, протоколы : учеб. пособие для вузов по направлению 552800 "Информатика и вычисл. техника"... / В. Г. Олифер, Н. А. Олифер. - 3-е изд. - СПб. : Питер, 2008. - 957 с.: а-ил. - (Учебник для вузов).

2 Пятибратов, А. П. Вычислительные системы, сети и телекоммуникации: учебник для вузов / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко .- 4-е изд., перераб. и доп. - М. : Финансы и статистика, 2008. - 736 с. - Библиогр.: с. 718 - 721. - Предм. указ.: с. 727-734. - ISBN 978-5-279-03285-3.

3 Бройдо, В. Л. Вычислительные системы, сети и телекоммуникации: учеб. пособие для вузов / В. Л. Бройдо, О. П. Ильина .- 3-е изд. - Санкт Петербург : Питер, 2008. - 766 с. : ил.. - (Учебное пособие). - Библиогр.: с. 756 - 759. - Алф. указ.: с. 760-765. - ISBN 978-5-91180-754-2.

4 Росляков, А. В. Сети доступа: учеб. пособие для студ. вузов, обучающихся по направлению 210400 - "Телекоммуникации" / А. В. Росляков . - М. : Горячая линия-Телеком, 2008. - 96 с. : ил.. - (Учебное пособие для высших учебных заведений. Специальность). - Библиогр.: с. 93 - 94. - ISBN 978-5-9912-0018-9.

5 Никифоров, С. В. Введение в сетевые технологии. Элементы применения и администрирования сетей: учеб. пособие для вузов / С. В. Никифоров .- 2-е изд. - М. : Финансы и статистика, 2007. - 224 с. - Предм. указ.: с. 220 - 223. - ISBN 978-5-279-03280-8.

6 Новиков, Ю. В. Локальные сети: архитектура, алгоритмы, проектирование / Новиков Ю. В. , Кондратенко С. В. . - М. : ЭКОМ, 2001. - 312 с. : ил. - (Современные компьютерные технологии) - ISBN 5-7163-0061-8.

7 Гук, М. Аппаратные средства локальных сетей: энциклопедия / М. Гук . - СПб. : Питер, 2002. - 576 с. : ил. - ISBN 5-8046-0113-X.