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

Факультет компьютерного проектирования

Кафедра проектирования информационно-компьютерных систем

КОМПЬЮТЕРНАЯ ТЕХНИКА, СИСТЕМЫ И СЕТИ. ЛАБОРАТОРНЫЙ ПРАКТИКУМ

Рекомендовано УМО по образованию в области информатики и радиоэлектроники в качестве пособия для специальности 1-39 03 02 «Программируемые мобильные системы»

Под редакцией Е. Н. Шнейдерова

Минск БГУИР 2016

УДК [004.3+004.7](076.5)
ББК 32.973.26я73+32.973.202я73
К63

А в т о р ы:
Е. Н. Шнейдеров, Д. В. Лихачевский,
Д. В. Лукашонок, С. М. Боровиков

Р е ц е н з е н т ы:
кафедра электронных вычислительных машин и систем учреждения образования «Брестский государственный технический университет» (протокол №6 от 25.03.2015);

профессор кафедры радиоэлектроники филиала «Минский радиотехнический колледж» учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», доктор технических наук, профессор Ф. Д. Троян

Компьютерная техника, системы и сети. Лабораторный практикум : пособие / Е. Н. Шнейдеров [и др.] ; под ред. Е. Н. Шнейдерова. – Минск : БГУИР, 2016. – 104 с. : ил.
К63
ISBN 978-985-543-191-7.

Пособие состоит из описания восьми лабораторных работ по учебным дисциплинам «Основы компьютерной техники и программирования мобильных электронных систем» и «Введение в компьютерные системы и сети».

Лабораторные работы могут быть также использованы для подготовки отдельных тем сертификационных экзаменов *CompTIA A+ Certification* и *CCNA R&S: Introduction to Networks*.

УДК [004.3+004.7](076.5)
ББК 32.973.26я73+32.973.202я73

ISBN 978-985-543-191-7

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ

В соответствии с учебным планом специальности 1-39 03 02 «Программируемые мобильные системы» дисциплины «Основы компьютерной техники и программирования мобильных электронных систем» (ОКТиПМЭС) и «Введение в компьютерные системы и сети» (ВвКСиС) предусматривают по 16 ч лабораторных занятий каждая.

Данное пособие включает восемь лабораторных работ, одна из которых является расчётно-исследовательской, четыре – практическими и три – виртуальными. Практикум позволяет сформировать умения и развить навыки практической подготовки студентов по дисциплинам ОКТиПМЭС и ВвКСиС в соответствии с учебными программами дисциплин.

Слово «виртуальные» подчёркивает то, что в указанных лабораторных работах структура и состав компьютерной сети, а также происходящие в сети процессы моделируются в памяти ЭВМ. Моделирование выполняется на базе программного обеспечения *Cisco Packet Tracer*.

Лабораторные работы №1, №2 и №3 позволяют подготовиться к отдельным темам экзамена международно признаваемой сертификации для техников *CompTIA A+*. Лабораторные работы №3, №5, №6, №7, №8 дают базовые знания в области компьютерных сетей и позволяют подготовиться к отдельным темам сертификации *Cisco CCNA R&S: Introduction to Networks*.

Для усложнения при необходимости задания к некоторым из лабораторных работ можно объединять. Например, можно совместно выполнить работы №3 + №4 или №4 + №6 и т. п. Авторы считают, что такой подход позволит студенту одновременно применять как теоретические, так и практические навыки и, следовательно, более эффективно освоить материал дисциплин.

При подготовке лабораторного практикума техническую помощь оказывали студенты группы 413802 специальности «Программируемые мобильные системы» А. И. Коновалов и В. В. Логвинович.

ЛАБОРАТОРНАЯ РАБОТА №1
СБОРКА НАСТОЛЬНОГО КОМПЬЮТЕРА И ИЗУЧЕНИЕ ЕГО КОМПОНЕНТОВ

1.1 Цель работы

Изучить компоненты системного блока настольного персонального компьютера, их совместимость и принцип подбора. Получить навыки сборки-разборки системного блока для его модернизации и технического обслуживания.

1.2 Краткие теоретические сведения

В зависимости от конструкторского исполнения и назначения современные компьютеры бывают различных типов (таблица 1.1).

Таблица 1.1 – Классификация компьютеров по конструкторскому исполнению

Стационарные компьютеры (***stationarity computers***)	**Настольные компьютеры** (***desktop***)	Настольные компьютеры имеют традиционную модульную компоновку и чаще всего состоят из системного блока, устройств ввода-вывода и периферийных устройств. Они достаточно легко модернизируются и обслуживаются. К настольным компьютерам относят **неттопы** (***nettops***, миниатюрные настольные компьютеры) и **приставки** (специализированные компьютеры, например игровые)
	Моноблоки (***all-in-one***)	Моноблоки имеют единый цельный корпус-монитор. Они отличаются малым занимаемым объёмом, сложностью модернизации и обслуживания и хорошими эстетическими показателями
	Стоечные компьютеры (***rack computers***)	Стоечные компьютеры отличаются унифицированными размерами, профессиональной проработкой компонентов и используются, как правило, в качестве серверов
	ЦОД (***data centres***)	Центры обработки данных − совокупность (чаще всего стоечных) компьютеров, объединённых в единый вычислительный комплекс. Они обычно имеют глубоко модульную структуру, а также при правильном администрировании легко модернизируются и обслуживаются
Портативные компьютеры (***portative computers***)	**Ноутбуки** (***notebooks***)	Ноутбуки представляют собой переносные компьютеры, в корпусе которых объединены типовые компоненты настольных компьютеров. Отличаются малым весом и объёмом, длительным временем автономной работы, сложностью модернизации и обслуживания. Различают следующие разновидности ноутбуков: **нетбуки** (***netbooks***, ноутбуки малой мощности), **ультрабуки** (***ultrabooks***, ноутбуки малых размеров), **ноутбуки-трансформеры** (гибриды планшетного компьютера и ноутбука)
	Планшетные компьютеры (***tablet***)	Планшетные компьютеры являются модернизацией ноутбуков, в которых отсутствует физическая клавиатура, а ввод информации осуществляется за счёт сенсорного экрана. Они редко являются модернизируемыми. К планшетным компьютерам относят **электронные книги** (***e-book reader***)
	Смартфоны (***smartphones***)	Смартфоны являются гибридами планшетного компьютера и мобильного телефона

Компоненты настольного компьютера

Системный блок настольного компьютера имеет типовой аппаратный состав, который включает в себя следующие компоненты: системную плату, центральный процессор, энергозависимую оперативную память, энергонезависимую память, видеоподсистему, аудиоподсистему, сетевой адаптер и блок питания. Дополнительно в состав настольного компьютера включены подсистема охлаждения и устройства чтения переносных запоминающих устройств и другие подсистемы, расширяющие его функциональность.

Системная (материнская) плата

На рисунке 1.1 представлен типичный набор компонентов материнской платы, выпускаемых на протяжении достаточно длительного периода времени. Он может отличаться в зависимости от модели или производителя. Тем не менее практически все материнские платы имеют в своём составе бóльшую часть из представленного далее перечня компонентов и разъёмов:

1 Разъём для установки центрального процессора (***socket***). Как правило, число в названии разъёма обозначает количество контактов (для процессоров *Intel*) или порядковый номер разработки (для процессоров *AMD*).

2 Чипсет (***chipset***) – набор микросхем системной платы (иногда их называют микросхемами системной логики), который определяет функциональность и быстродействие этой платы. Чипсет содержит две основные микросхемы, которые называют **северным мостом** (***northbridge***, контроллер-концентратор памяти) и **южным мостом** (***southbridge***, контроллер ввода-вывода). Чипсет обеспечивает совместимость (и синхронность) работы компонентов материнской платы и подключённых к ней устройств.

3 Микросхема базовой системы ввода-вывода (***BIOS***, ***basic input output system***) – микросхема *EEPROM*, содержащая программу, выполняемую первой при включении компьютера. Настройки *BIOS* хранятся в микросхеме энергозависимой *CMOS*-памяти, для питания которой на плате предусмотрена литиевая батарея питания.

4 Разъёмы оперативной памяти, представленной обычно *DIMM*-модулями. Для правильной установки планок памяти в двухканальном режиме работы, разъёмы выкрашены в различные цвета. Отдельный цвет соответствует отдельному каналу.

5 Разъёмы для подключения к внутренним шинам – это разъёмы, предназначенные для установки дополнительных модулей (графического адаптера, сетевой и звуковой плат, модема, тюнера и т. д.), расширяющих функциональные возможности компьютера. На сегодняшний момент универсальным интерфейсом расширения является *PCI Express* *x*1, *x*4, *x*8, *x*16 – последовательная шина для подключения различных плат расширения (заменила собой интерфейсы *PCI*, *PCI-X*, *AGP*).

1 – Разъём для подключения кулера
2 – Разъём для передней панели корпуса
3 – Интерфейсы *USB 2.0*
4 – Четырёхсегментный *POST*-индикатор
5 – Интерфейс *IEEE 1394*
6 – Интерфейс для подключения криптопроцессора
7 – Кнопки перезагрузки/включения
8 – Разъём для аудио передней панели корпуса
9 – Интерфейс *PCI-Express x16*
10 – Литиевая батарея питания
11 – Интерфейс *RS-232* (*COM*)
12 – Переключатель режима диагностики БП
13 – Контроллер аудио
14 – Интерфейс *Thunderbolt*
15 – Интерфейс *PCI-Express x1*
16 – Контроллер *Ethernet*
17 – Разъём питания *EATX12V_1 6pins*
18 – Разъёмы аудио
19 – Интерфейсы 2×*USB 2.0*, *Ethernet* (*RJ-45*)
20 – Кнопка для поддержки *USB BIOS Flashback*
21 – Интерфейсы 2×*USB 3.0*, *Ethernet* (*RJ-45*)
22 – Интерфейсы *HDMI*, *DisplayPort*, *Optical S/PDIF*
23 – Интерфейс *Mini DisplayPort*
24 – Интерфейсы 2×*USB 3.0*, 2×*eSATA*
25 – Интерфейсы 2×*USB 2.0*, *PS/2 combo*
26 – Разъёмы 2×*ATX12V 8pins*
27 – Переключатель *EPU* (пониженное энергопотребление)
28 – Переключатель *TPU* (авторазгон)
29 – Разъём процессора *LGA1150*
30 – Разъёмы 4×*DIMM DDR3* (*RAM*)
31 – Разъём питания *EATXPWR 24pins*
32 – Интерфейс *USB 3.0*
33 – Разъёмы подключения кулеров
34 – Контроллер *BIOS*
35 – Северный мост *Intel Z87*
36 – Интерфейсы 4×*SATA 6Gb/s* (*Marvel*)
37 – Контроллер южного моста
38 – Интерфейс *mSATA*
39 – Интерфейсы 6×*SATA 6Gb/s*
40 – Кнопка сброса настроек *BIOS*

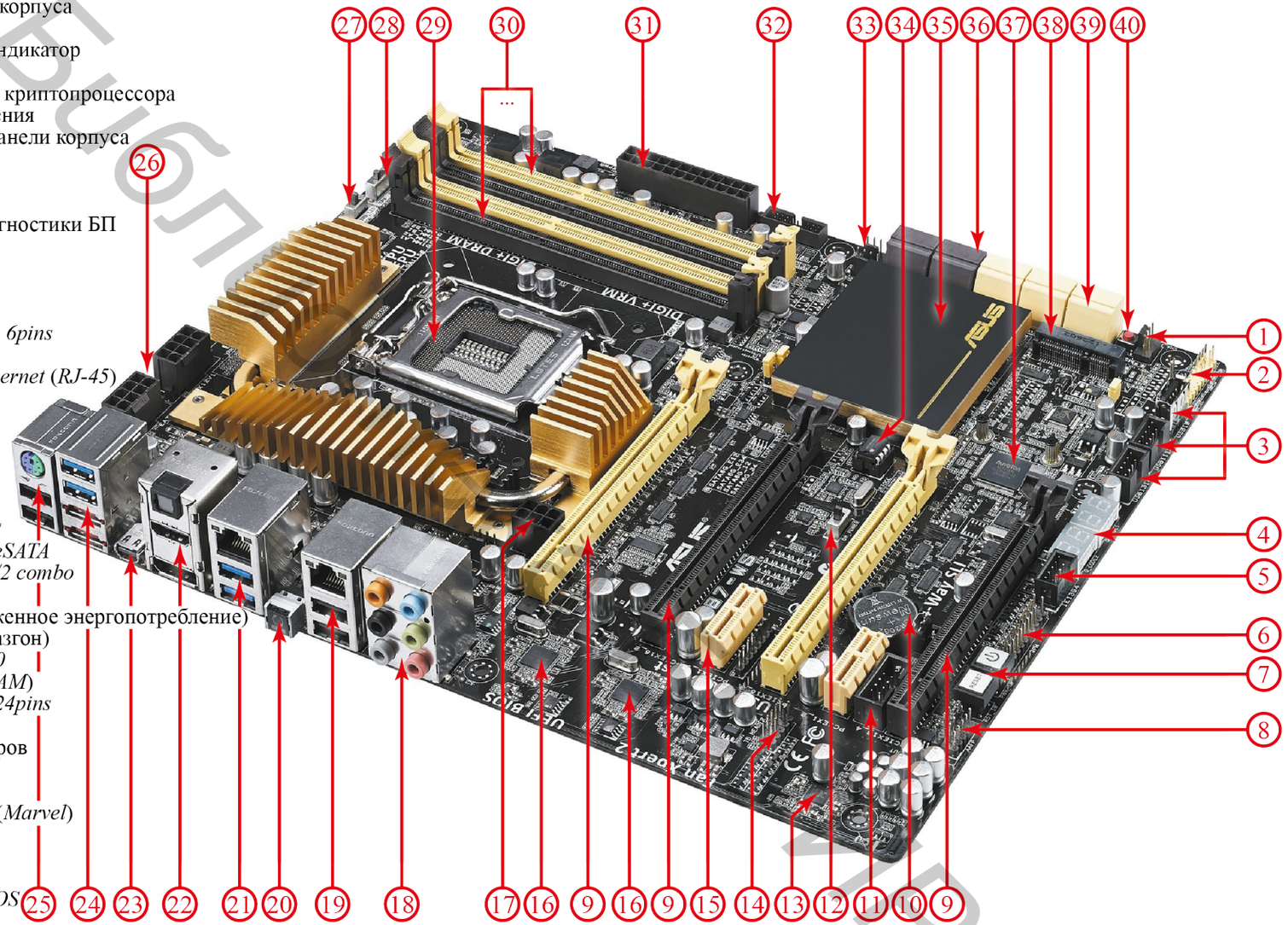
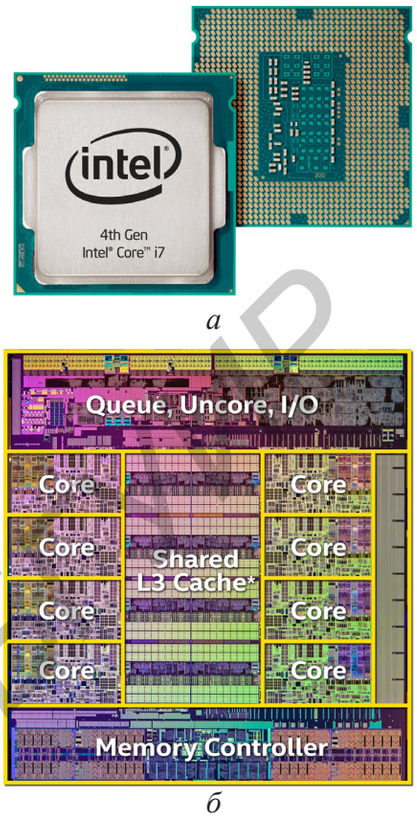


Рисунок 1.1 – Компоненты материнской платы

Процессор

Процессор (***central processing unit***, ***CPU***) представляет собой большую интегральную схему, выполняющую машинные инструкции. Конструктивно является полупроводниковой пластиной размером примерно 20×20 мм, помещённой в плоский металлический корпус, на одной стороне которой располагается множество контактов (рисунок 1.2, *а*).

Использование современных промышленных технологий позволяет разместить на кристалле процессора большое количество функциональных элементов (например, ~2,6 млрд транзисторов для *Intel Core i7-5960X* при площади кристалла всего 17,6×20,2 мм, рисунок 1.2, *б*), обеспечивающих многопоточную и быструю обработку информации.



а

б

а – внешний вид;
б – функциональные области *CPU* на кристалле

Рисунок 1.2 – Процессор

С точки зрения структуры процессор содержит:
– арифметико-логическое устройство;
– устройство управления;
– регистры и счётчики команд;
– шины данных и шины адресов;
– кэш (скоростная память малого объёма);
– графическое ядро (в зависимости от модели).

Основными характеристиками процессора являются:
– количество ядер (от 1 до 8 для процессоров общего применения);
– тактовая частота (как правило, составляет несколько ГГц);
– разрядность (64 или 32 бита, последние являются устаревшими);
– объём кэш-памяти (для *L*2 – до 2 Мбайт, для *L*3 – 2...20 Мбайт);
– максимально поддерживаемый объём и частота адресуемой памяти;
– максимальная рассеиваемая мощность (*thermal design power*, *TDP*);
– поддерживаемые технологии и преимущества.

В вычислительной системе может быть несколько параллельно работающих процессоров. Такие системы называются **многопроцессорными**.

Основными производителями процессоров для персональных компьютеров являются *Intel Corporation* (более 70 % рынка) и *Advanced Micro Devices*, *Inc*. (более 16 %). Для мобильных систем этот список представлен также компаниями *Qualcomm* (более 50 %), *Apple*, *MediaTek*, *Spreadtrum* и *Samsung*.[1]

Для выбора процессора по параметрам, сравнения различных моделей и просмотра спецификаций рекомендуются источники [1, 2].

[1] Информация сформирована по данным *Strategy Analytics* на сентябрь 2014 г.

Оперативная память

Исторически название «оперативная» эта память получила потому, что она имеет достаточно высокую производительность. Она является хранилищем информации, которая была или будет обработана процессором. Все устройства, связываются с оперативной памятью через кэш и **системную шину**.

Оперативную память можно рассматривать как некоторый набор ячеек, каждая из которых способна хранить блок информации. Часто для оперативной памяти используют обозначение ***RAM*** (***random access memory***, память с произвольным доступом), которое означает то, что при необходимости память может напрямую обратиться к одной необходимой ячейке, не затрагивая при этом остальные. При этом скорость произвольного доступа не меняется от места нахождения нужной информации, что является преимуществом. Однако из-за того, что ячейки памяти содержат конденсаторы (наиболее дешёвая и долговечная технология), содержащиеся в них данные сохраняются только пока компьютер работает, т. е. конденсаторы сохраняют заряд.

Оперативная память изготавливается в виде **модулей памяти** (рисунок 1.3). Модули памяти представляют собой пластины, на которых размещаются микросхемы памяти и контакты для подключения. Модули могут различаться между собой по размеру и количеству контактов (*SIMM*, *DIMM* или *SO-DIMM*).

Основными характеристиками модулей *RAM* являются:

– объём памяти модуля;

– тип (конструкторская и аппаратная реализация интерфейса взаимодействия с системной шиной; включает в себя частоту работы, пропускную способность и др.);

– временные задержки адресации и доступа к ячейкам памяти (**тайминги**);

– количество микросхем памяти;

– способность к коррекции ошибок;

– наличие охлаждения.

Наиболее популярной и производительной памятью в персональных компьютерах стала *DDR SDRAM*.

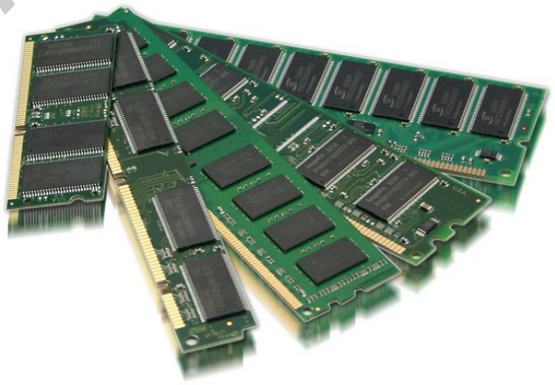


Рисунок 1.3 – Модули оперативной памяти

Жёсткий диск

Жёсткий диск («винчестер», ***HDD*** – ***hard disc drive***) – энергонезависимый накопитель информации, основанный на эффекте магнетизма. Он представляет собой несколько двусторонних магнитных дисков (носителей информации), размещённых на одном валу, заключённых в металлический корпус и вращающихся с большой скоростью (рисунок 1.4). Считывание и запись информации производится специальными головками, расположенными на конце коромысла.

В полугерметичном блоке находятся двусторонние пластины с нанесённым

Рисунок 1.4 – Жёсткий диск

на них магнитным слоем, посаженные на вал двигателя и вращающиеся со скоростью от 3600 оборотов в минуту. Всеми головками чтения управляет специальный привод основанный на электромагнетизме. Вся управляющая диском электроника: кэш-память, микроконтроллер интерфейса и др. – расположены на небольшой плате в нижней части жёсткого диска. Для подключения интерфейса и питания используются стандартные разъёмы *PATA* / *SATA* и *Molex* /*Power SATA*.

Диск называется «жёстким», т. к. в отличие от дискет (гибких дисков) имеет жёсткое основание, которое чаще всего выполнено из алюминия с нанесённым поверх него магнитным слоем.

Основными характеристиками *HDD* являются:

– интерфейс подключения (*PATA*, *SATA*, *eSATA*, *SCSI*, *FireWire* и др.);

– объём памяти (от 40 Гбайт до 8 Тбайт);

– форм-фактор (3.5", 2.5", 1.8", 1.3", 1");

– скорость вращения шпинделя (от 3600 до 15000 об/мин);

– объём кэш-памяти (буфер, от 4 до 256 Мбайт);

– время произвольного доступа (тайминги);

– надёжность в часах наработки (обычно не указывается при использовании технологии ***S.M.A.R.T.*** – *self-monitoring*, *analysis and reporting technology*);

– устоявшаяся скорость передачи данных (от 40 до 500 Мбайт/с);

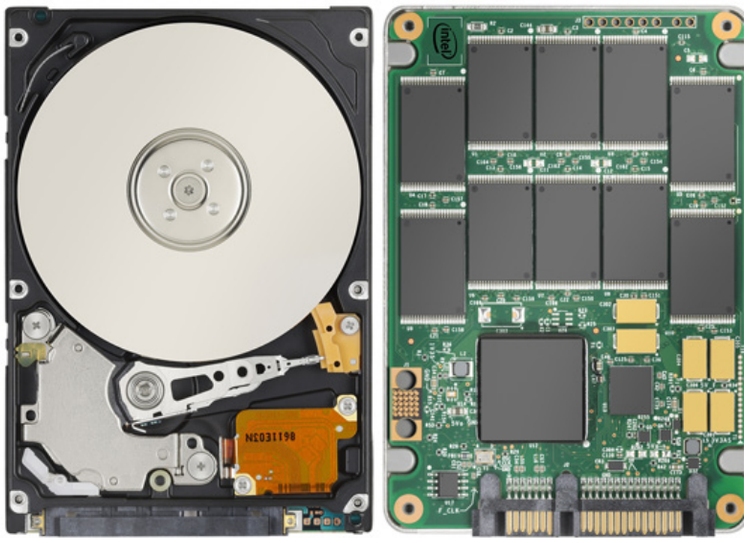
– энергопотребление в режиме записи и в режиме ожидания;

– степень ударопрочности, уровень шума и др.

Твердотельный накопитель

Твердотельный накопитель (***SSD*** – ***solid-state drive***) – энергонезависимый носитель информации на основе микросхем *NAND*-памяти (рисунок 1.5, *б*). В *SSD* не используются движущиеся части и элементы по сравнению с жёсткими дисками, что исключает вероятность их износа механическим путём.

Твердотельный накопитель состоит из самих *NAND*-микросхем, управляющего микроконтроллера, микросхемы энергозависимой кэш-памяти и печатной платы, на которой всё это монтируется.



а *б*

а – HDD; *б – SSD*

Рисунок 1.5 – Сравнение внутреннего устройства энергонезависимых носителей информации

В большинстве *SSD*-накопителей используется дешёвая *MLC*-память (*multi level cell*), которая может вмещать в одну хранимую ячейку более одного бита в отличие от *SLC*-памяти (*single level cell*). Это очень результативно сказывается на цене готового изделия и способствует популяризации данных накопителей. Однако *MLC*-память имеет недостаток – относительно небольшой ресурс перезаписи информации, который компенсируется специальными алгоритмами записи.

SLC записывают только один бит в ячейку и это обеспечивает значительно лучшую долговечность и до двух раз более высокую скорость в сравнении с *MLC*, но цена накопителей на *SLC*-память примерно в два раза выше.

В функции *SSD*-контроллера входит операция *TRIM* (для интерфейсов *ATA*), чтение, запись и кеширование информации, коррекция ошибок (*ECC*), шифрование, *S.M.A.R.T.*, пометка нерабочих блоков и сжатие данных.

Все контроллеры памяти нацелены на параллельно подключённую *NAND*-память. Так как шина памяти одного чипа очень мала (максимум 16 бит), используются шины многих чипов, подключённых параллельно (аналогия *RAID* 0). К тому же отдельно взятый чип отнюдь не обладает отличными характеристиками, а наоборот. Например, высокую задержку ввода-вывода. Когда чипы памяти параллельно объединены, эти задержки скрываются, распределяясь между ними. Да и шина растёт пропорционально каждому добавленному чипу, вплоть до максимальной пропускной способности контроллера.

Самыми распространёнными интерфейсами для *SSD* потребительского класса являются *SATA* 3.0 6 Гбит/с, *PCI-Express* и *USB* 3.0.

Графический адаптер

Графический адаптер (видеокарта, ***GPU*** *– graphic processor unit*) – устройство компьютера (рисунок 1.6), предназначенное для формирования информации, хранимой в памяти, пригодной для отображения на мониторе. В первую очередь под графическим адаптером понимают устройство с **графическим процессором**, который и занимается формированием графического образа. Однако встроенный графический процессор современных видеокарт может производить дополнительную обработку информации, снимая эту задачу с *CPU*. Например, все современные видеокарты *Nvidia* и *AMD* осуществляют рендеринг графического

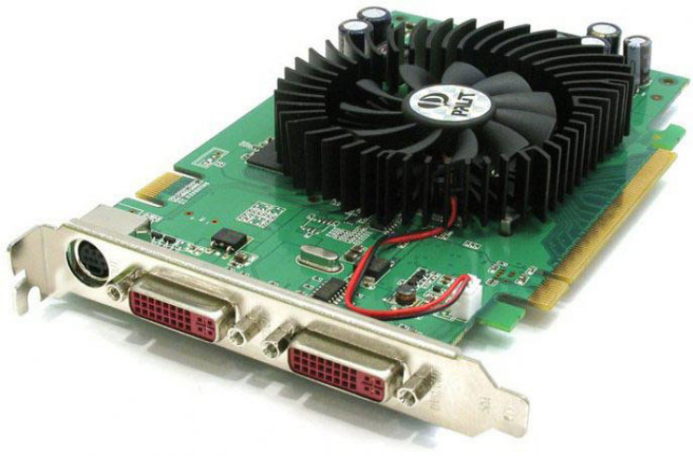


Рисунок 1.6 – Дискретный графический адаптер

конвейера ***OpenGL*** и ***DirectX*** на аппаратном уровне.

Обычно графический адаптер выполнен в виде отдельной печатной платы (**дискретное исполнение**) и вставляется в разъём на материнской плате (*AGP*, *PCI Express*). Такой тип графического адаптера имеет собственную систему питания и охлаждения. Отличается значительно бóльшей производительностью и тепловыделением (топовые видеокарты потребляют более 300 Вт энергии при полной нагрузке). Стоит отметить, что для дискретной видеокарты требуется удовлетворяющий её потребности блок питания.

Также существуют и **встроенные** (интегрированные) видеокарты – как в виде отдельного чипа, так и в качестве составляющей части северного моста чипсета или *CPU*. Такой тип адаптеров является ограниченным по тепловыделению, потому мощные микросхемы для встроенных видеокарт не используются. В качестве буфера они используют оперативную память. Разъёмы встроенных видеокарт установлены непосредственно на материнской плате. Обычно это *VGA*, *DVI*, *HDMI* или *DisplayPort*.

Основными характеристиками *GPU* являются:

– частота графического процессора;
– объём оперативной памяти графического адаптера (*GDDR*);
– ширина шины памяти (в битах);
– интерфейс подключения к материнской плате;
– интерфейсы подключения устройств отображения информации;
– мощность минимального рекомендуемого блока питания;
– тип охлаждения, поддержка сопроцессоров и др.

Звуковой адаптер

Звуковой адаптер (*soundcard*) – устройство компьютера, позволяющее обрабатывать звук. Он содержит в себе два преобразователя сигналов:

– аналого-цифровой (АЦП), преобразующий непрерывный (аналоговый) звуковой сигнал (речь, музыку, шум и др.) в дискретный (цифровой) сигнал для обработки и хранения компьютером;

– цифроаналоговый (ЦАП), выполняющий обратное преобразование сохранённого в цифровом виде звука в аналоговый сигнал, который затем воспроизводится с помощью **акустической системы**.

На момент появления звуковые платы представляли собой отдельные устройства, но в современных материнских платах они, как правило, интегрированы в материнскую плату.

Сетевой адаптер

Сетевой адаптер (***NIC*** – *network interface card*) – устройство компьютера, позволяющее ему взаимодействовать с другими устройствами посредством компьютерной сети. В настоящее время в персональных компьютерах и ноутбуках контроллер и компоненты, выполняющие функции сетевого адаптера, довольно часто интегрированы в системную плату для удобства и удешевления всего компьютера в целом.

Основными характеристиками *NIC* являются:

– интерфейс подключения к системной плате (*PCI*, *PCI Ex.*, *USB* и др.);

– сетевая среда передачи данных (*network media*);

– поддерживаемые стандарты передачи данных, которые в свою очередь определяют максимальную пропускную способность (10 Мбит/с, 100 Мбит/с, 1 Гбит/с и др.).

Блок питания

Блок питания (**БП**, ***PSU*** – *power supply unit*) – устройство (вторичный источник электропитания), предназначенное для снабжения компонентов компьютера электроэнергией постоянного тока путём преобразования сетевого напряжения до требуемых значений (см. лабораторную работу №2, рисунок 2.2).

Основные характеристики блоков питания приведены в теоретической части лабораторной работы №2.

Аппаратные интерфейсы персонального компьютера

Аппаратный интерфейс, порт (*port*) – специализированный разъём чаще всего на системной плате, предназначенный для подключения оборудования определённого типа. Обычно портами называют разъёмы, предназначенные для работы периферийного оборудования, существенно отделённого от архитектуры компьютера (например, не называют портами разъёмы *PCI*-, *ISA*-, *AGP*- и *PCI*-шин, а также разъёмы для оперативной памяти и процессора).

Некоторые порты допускают **«горячее» подключение** и **отключение**, другие – обязательно требуют предварительного отключения соединяемого оборудования от электросети.

Ниже приведён обзор часто используемых в системных платах аппаратных интерфейсов (рисунок 1.7).

RS-232 (***COM*-порт**) – порт для универсальной синхронной передачи сигнала (рисунок 1.7, *а*). Широко известен как последовательный порт компьютера. Используется для подключения различного вида оборудования по собственным протоколам передачи данных.

Parallel port (***LPT***), **параллельный порт** – тип интерфейса (рисунок 1.7, *б*), разработанный для подключения различных периферийных устройств (принтеров, модемов, звуковых адаптеров и др.).


а


б


в


г


д


е


ж


з


и


к


л


м

Рисунок 1.7 – Аппаратные интерфейсы

Universal Serial Bus (***USB***) – последовательный интерфейс передачи данных для среднескоростных и низкоскоростных периферийных устройств (рисунок 1.7, *в*). Он имеет различные спецификации: *USB* 1.1, *USB* 2.0, *USB* 3.0. Существуют различные типы разъёмов: *A* – на стороне ведущего устройства и *B* – на стороне ведомого устройства.

IEEE 1394 (***FireWire***, ***i-Link***) – универсальный интерфейс, предназначенный для обмена информацией между компьютером и другими электронными устройствами (рисунок 1.7, *г*). *FireWire* может использоваться как для создания компьютерной сети равнозначных устройств, так и для подключения аудио- и видеоустройств, принтеров и сканеров, подключения устройств хранения информации и др.

External SATA (***eSATA***) – интерфейс подключения внешних устройств, поддерживающий режим «горячей замены» (рисунок 1.7, *д*). Средняя скорость передачи данных выше, чем у *USB* 2.0 или *IEEE 1394*.

Video Graphics Array (***VGA***) – разъём для передачи аналогового сигнала к графическим устройствам (рисунок 1.7, *е*). На подключение по *VGA* могут влиять внешние помехи.

Digital Visual Interface (***DVI***) – интерфейс, предназначенный для передачи цифрового сигнала к графическим устройствам (рисунок 1.7, *ж*). Также *DVI* позволяет выполнять автоматическую коррекцию картинки.

High Definition Multimedia Interface (***HDMI***) – интерфейс для передачи мультимедийных данных высокой чёткости с защитой от копирования (рисунок 1.7, *з*).

DisplayPort – сигнальный интерфейс для цифровых мониторов (рисунок 1.7, *и*). Он является наиболее современным интерфейсом для подключения аудио- и видеоаппаратуры к компьютерной технике.

PS/2 – порт, применяемый для подключения клавиатуры и мыши к компьютеру (рисунок 1.7, *к*). Его спецификация определяет стандартные цвета для разъёмов в системном блоке и кабелей подключаемых устройств: сиреневый – клавиатура, зеленый – мышь.

Registered Jack (***RJ***) – интерфейс, используемый для соединения различного оборудования в сети (рисунок 1.7, *л*). Различают типы интерфейсов: *RJ*-11, *RJ*-14, *RJ*-25, *RJ*-45 и др. *RJ*-45 наиболее распространен, потому что применяется для построения компьютерных сетей.

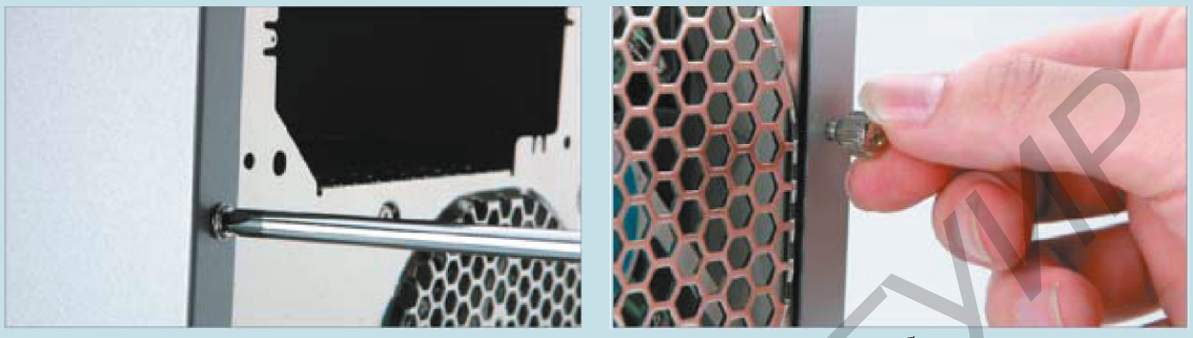
Audio I/O – порты для передачи аудиосигнала к звуковой аппаратуре или от неё (рисунок 1.7, *м*).

Процесс сборки настольного компьютера

Перед сборкой компьютера необходимо убедиться в правильности подбора комплектующих и их совместимости друг с другом.

Сборка компьютера выполняется в следующей последовательности:

1 С помощью отвёртки выкрутить винты крепления боковых панелей корпуса на задней стенке (рисунок 1.8) и снять боковые панели, сдвигая их назад.



а *б*

а – выкручивание с помощью отвёртки; *б* – выкручивание вручную

Рисунок 1.8 – Съём винтов боковых панелей корпуса компьютера

2 Установить блок питания в верхнюю или нижнюю часть корпуса (в зависимости от типа корпуса) и закрепить его винтами (рисунок 1.9).

а *б*

а – установка блока питания; *б* – закрепление блока питания

Рисунок 1.9 – Установка и закрепление блока питания компьютера в корпусе

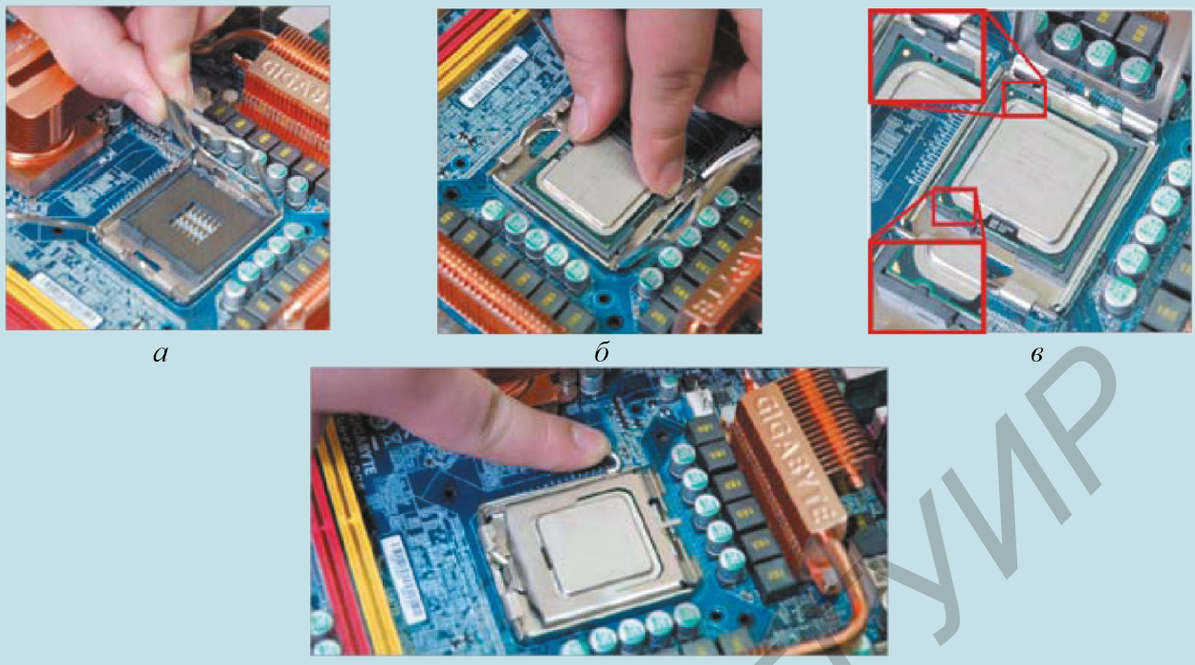
3 Далее необходимо отдельно собрать системную (материнскую) плату.

3.1 Освободить крепёжную рамку разъёма процессора, точно в пазы установить процессор в разъём и зафиксировать его крепёжной рамкой (рисунок 1.10 на примере процессора *Intel* и разъёма *LGA775*).

3.2 Нанести термопасту на поверхность процессора. Паста должна тонким непрерывным слоем покрывать металлическую поверхность (рисунок 1.11, *а*, *б*). Назначение термопасты – заполнить тонкий зазор между процессором и системой охлаждения для обеспечения лучшей теплоотдачи и защиты от перегрева.

При выполнении этого действия нельзя допускать попадания пасты на контакты процессора и других окружающих его элементов.

3.3 При необходимости нанести термопасту на поверхность охлаждающего радиатора. Стоит отметить, что в большинстве случаев радиаторы уже продаются с нанесённой на них тонкой плёнкой термопасты.

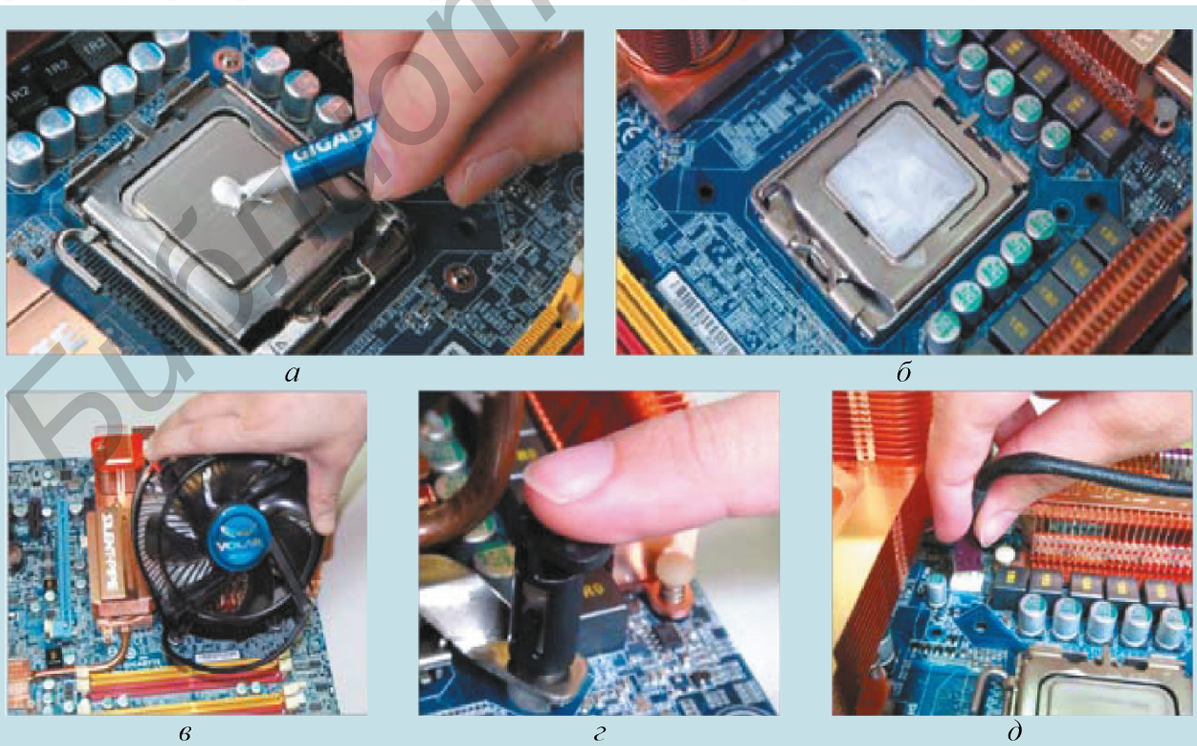


а *б* *в*

г

а – освобождение крепёжной рамки; *б*, *в* – установка процессора в разъём;
г – фиксация процессора крепёжной рамкой
Рисунок 1.10 – Установка процессора в разъём

3.4 Установить систему охлаждения, вставив крепёжные штырьки в отверстия на печатной плате либо защёлкнув замки крепления радиаторов (в зависимости от типа крепления системы охлаждения), как показано на рисунке 1.11, *в*, *г*. Затем необходимо подключить питание к системе охлаждения, подсоединить провод кулера к разъёму с маркировкой *FAN* (см. рисунок 1.11, *д*).



а *б*

в *г* *д*

а, *б* – нанесение термопасты; *в*, *г*, *д* – установка системы воздушного охлаждения на плату
Рисунок 1.11 – Установка системы воздушного охлаждения

3.5 В *DIMM*-разъёмы, добившись совпадения ключевого выреза, установить модули оперативной памяти, как показано на рисунке 1.12. Для этого защёлки отжимаются в стороны и модуль устанавливается в пазы. При полном вхождении модуля в разъём защёлки автоматически закрываются.

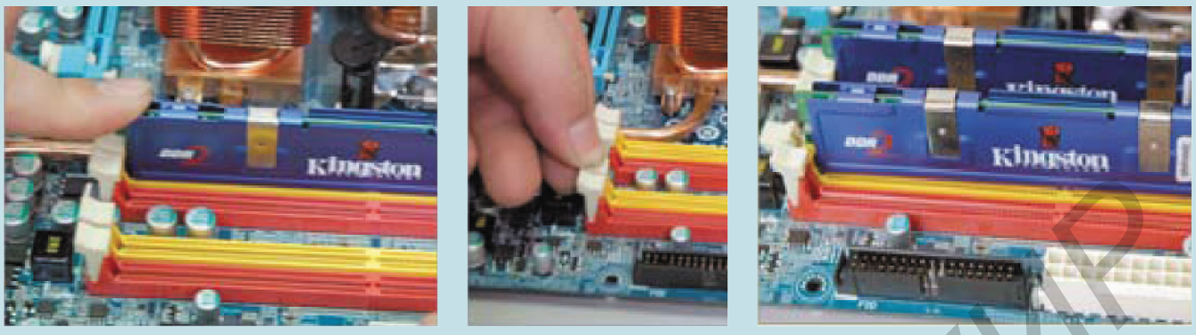


Рисунок 1.12 – Установка модулей оперативной памяти

Чтобы память могла работать в двухканальном режиме, необходимо установить пару модулей в разъёмы одного цвета.

4 В корпусе установить маленькие медные стойки (обычно их 8...10 шт.) согласно отверстиям в системной плате (рисунок 1.13).

Рисунок 1.13 – Установка медных стоек для крепления системной платы

5 Установить на корпусе заглушку для торцевых разъёмов системной платы. Эта заглушка, как правило, идёт в комплекте с платой.

6 Совмещая торцевые разъёмы ввода-вывода с отверстиями в заглушке, а также отверстия для крепления, вставить системную плату в корпус (рисунок 1.14). Прилагаемыми к плате винтами зафиксировать её в корпусе. Для лучшей изоляции рекомендуется использовать шайбы.

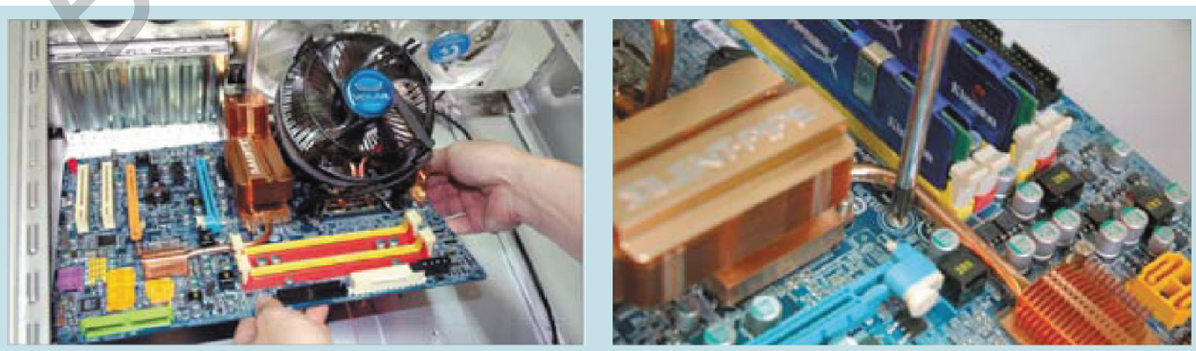


Рисунок 1.14 – Установка системной платы в корпус компьютера

7 Подключить жгут с 24-контактным разъёмом питания *ATX* от блока питания и 4-контактный (или 8-контактный) разъём +12 В к системной плате (некоторые системные платы требуют подключения отдельного кабеля +12 В), как это показано на рисунке 1.15. Ключи на разъёмах не позволят подключить их в неправильном положении.

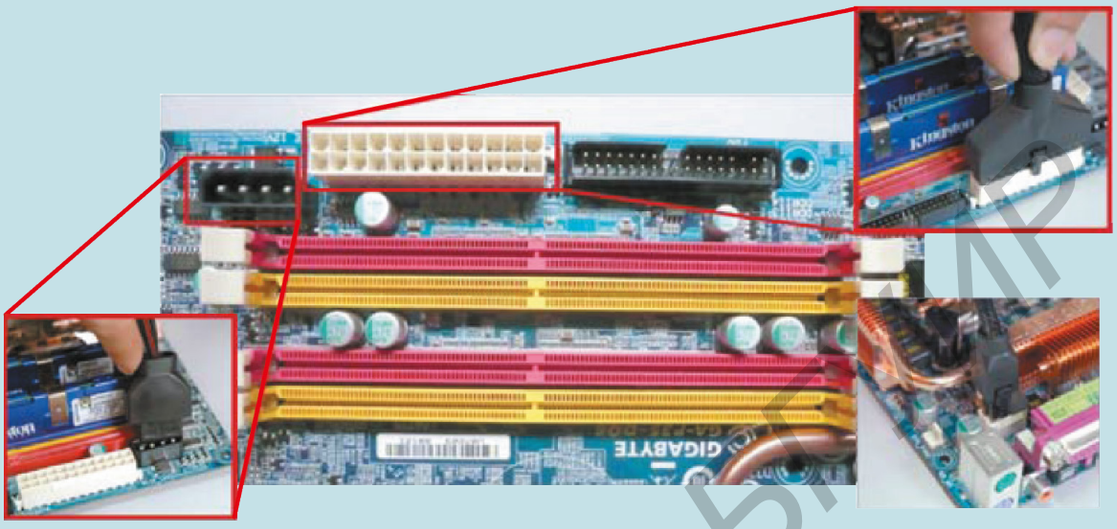


Рисунок 1.15 – Подключение питания к системной плате

8 В соответствии с разметкой и цветовой маркировкой подключить провода от кнопок питания и перезагрузки на передней панели корпуса. Важным является соблюдение полярности контактов; стрелка на разъёме соответствует положительному проводу. Назначение контактов указано в документации к системной плате.

Таким же образом необходимо подключить сигнальные *USB*, *IEEE* 1394 и аудиокабели (разъёмы имеют ключи для защиты от неправильного подключения) от передней панели корпуса (рисунок 1.16).

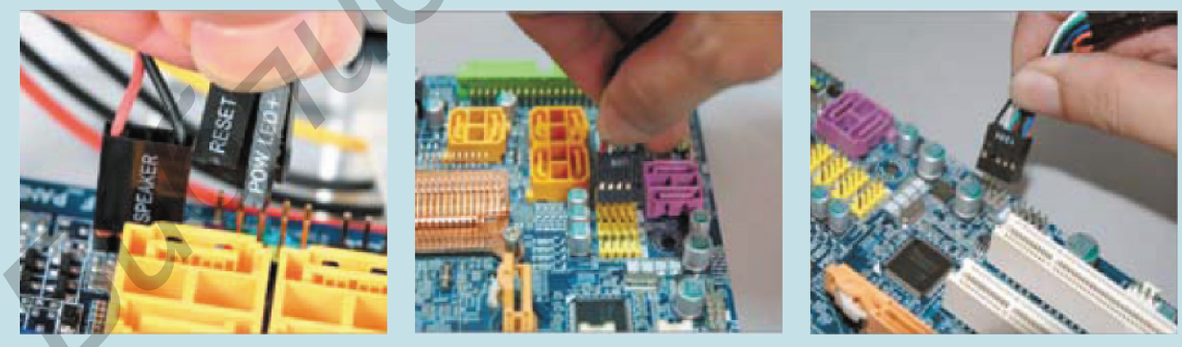


Рисунок 1.16 – Подключение компонентов передней панели корпуса к системной плате

9 Многие корпуса имеют разъёмы для наушников и микрофона на передней панели. На этом этапе необходимо подключить соответствующие сигнальные кабели в соответствии с документацией к системной плате.

1.3 Практическая часть лабораторной работы

Для выполнения практической части лабораторной работы необходимо:

1 Получить у преподавателя индивидуальное задание на подбор компонентов персонального компьютера, выполняющего определённые функции.

2 С помощью известных торговых площадок (например, *catalog.onliner.by* или *migom.by*) выбрать нужные для работы компоненты таким образом, чтобы результирующий компьютер смог выполнять возложенные на него функции, но в то же время общая стоимость всех компонентов не превышала указанную преподавателем в задании.

3 Пользуясь выданными преподавателем инструментами и соблюдая технику безопасности, выполнить разборку и сборку настольного персонального компьютера, после чего проверить его работоспособность.

4 Оформить отчёт по лабораторной работе.

1.4 Содержание отчёта

Отчёт о выполненной работе должен содержать:

1 Титульный лист. Цель лабораторной работы. Задание преподавателя.

2 Таблицу с выбранными компонентами для сборки компьютера без учёта блока питания (компоновку таблицы студентам выбрать самостоятельно, руководствуясь СТП БГУИР 01–2013).

3 Рекомендации по оптимизации выданного преподавателем компьютера.

4 Выводы по лабораторной работе.

1.5 Контрольные вопросы

1 Какова техника безопасности при выполнении работ по сборке-разборке персонального компьютера?

2 Каково назначение форм-фактора системной платы компьютера?

3 Какой инструмент и в каком случае используется для работы с компонентами компьютера?

4 Какие компоненты персонального компьютера имеют строгие требования к совместимости? Укажите эти требования.

5 Какие условия обязывают пользователя отдать предпочтение дискретному графическому адаптеру, а не встроенному?

1.6 Литература

1 Собери компьютер за 30 минут. Руководство GIGABYTE по сборке компьютера. – М. : Gigabyte Technology, 2012. – 100 с.

2 Meyers, M. CompTIA A+® Guide: Essentials (Exam 220-701). Third Edition / M. Meyers. – New York : McGraw-Hill / Osborne Media, 2010. – 713 p.

3 Брукс, Ч. CompTIA A+. Устройство, настройка, обслуживание и ремонт ПК / Ч. Брукс. – 3-е изд. перераб. и доп. ; пер. с англ. – СПб. : БВХ-Петербург, 2010. – 1232 с.

ЛАБОРАТОРНАЯ РАБОТА №2
ИССЛЕДОВАНИЕ ИСТОЧНИКА ПИТАНИЯ КОМПЬЮТЕРНОЙ СИСТЕМЫ

2.1 Цель работы

Изучить основные особенности электропитания компьютерных систем: характеристики источников питания, методы расчёта потребляемой мощности и принципы подбора компьютерного оборудования.

2.2 Краткие теоретические сведения

Источники питания компьютерных систем

Любое компьютерное устройство функционирует на основе электрической энергии. Источником этой энергии для неё может служить аккумулятор или переменное напряжение ~220 В, 50 Гц, получаемое из бытовой розетки (в некоторых странах бытовая электросеть может иметь другие характеристики).

Аккумулятор (рисунок 2.1) имеет постоянное выходное напряжение и ток, и его использование эффективно в мобильных вычислительных устройствах и устройствах с резервированием источника питания. При этом напряжение аккумулятора, как правило, не изменяется, а аккумулятор подбирается таким образом, чтобы изначально обеспечить нужные электрические характеристики, не прибегая к дополнительным приспособлениям.

Рисунок 2.1 – Аккумулятор мобильного вычислительного устройства

При питании вычислительных систем от бытовой электросети в системе предусматривается блок питания (БП), формирующий постоянное напряжение (точнее сказать несколько значений постоянного напряжения), которое требуется для работы аппаратуры (рисунок 2.2).

Рисунок 2.2 – Блок питания внутреннего исполнения настольного персонального компьютера

Любой блок питания описывается рядом электрических, технических и экономических характеристик. Применительно к компьютерной системе основополагающими характеристиками при выборе блока питания являются его электрические характеристики и типы соединителей, которыми он оснащён.

Характеристики источников питания

Основные электрические характеристики:

1 **Диапазон рабочих напряжений** (*AC input*) – интервал значений входного переменного напряжения, при котором блок питания сохраняет работоспособность и значения своих выходных параметров. На современном рынке различают БП с малым диапазоном рабочих напряжений (220...240 В) и расширенным диапазоном (110...230 В). Создание БП расширенного диапазона стало возможным благодаря использованию активного корректора коэффициента мощности.

2 **Мощность** (*power*) блока питания характеризует, сколько он может отдать электрической энергии подключаемым к нему приборам (материнская плата, видеокарта, жёсткий диск и др.) за единицу времени. Мощность часто обозначают на наклейке БП большим шрифтом. Мощность современных блоков питания настольных ПК варьируется в диапазоне 100...1500 Вт. Мощность распространённых БП составляет 400...700 Вт.

3 **Максимальный выходной ток** (*max DC output*) определяет возможности элементов блока питания (выходных каскадов БП, проводов и т. п.) при большой нагрузке. При превышении этого параметра (подключении к выходу БП большого количества вычислительных устройств) срабатывает встроенная защита и БП отключается. При некачественном БП защита может не сработать и последствия могут быть более значительными (рисунок 2.3). Большинство разработчиков БП предусматривают токи до 40 А в наиболее нагруженных линиях питания.

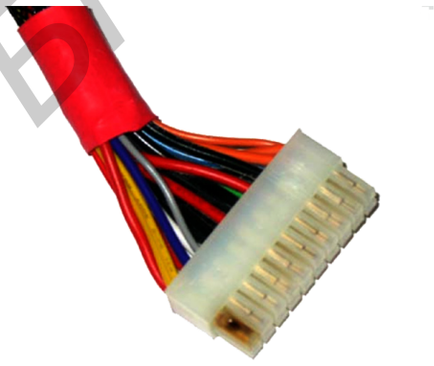


Рисунок 2.3 – Выгорание контактов БП вследствие большой токовой нагрузки

Основные характеристики, как правило, указываются производителем на информационной наклейке устройства (рисунок 2.4).

Дополнительные электрические характеристики:

1 **Внутреннее сопротивление** (*resistance*) блока питания характеризует потери мощности внутри БП при преобразовании им тока и напряжения.

2 **Стабильность выходного сигнала** (*output stability*) определяет постоянство значений выходных напряжений в течение времени при возможном изменении нагрузки БП.

3 **Коэффициент полезного действия** (*efficiency*) определяет отношение выходной мощности блока питания к потребляемой.

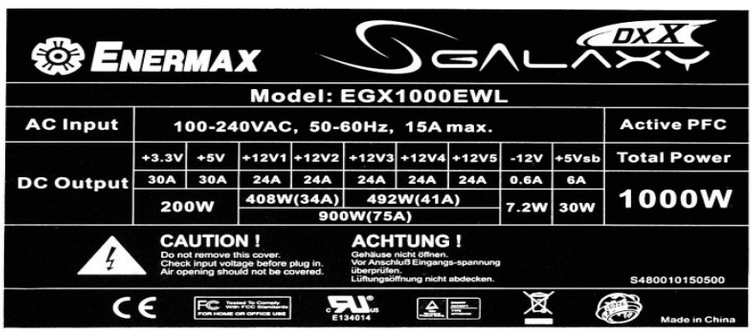


Рисунок 2.4 – Информационная наклейка блока питания

Основные технические характеристики источников питания:

1) диапазон рабочих температур;
2) надёжность блока питания (время наработки на отказ);
3) уровень шума, создаваемый блоком питания при работе;
4) частота вращения вентилятора блока питания;
5) масса блока питания;
6) длина питающих кабелей;
7) удобство в использовании (модульность);
8) экологичность блока питания;
9) соответствие государственным и международным стандартам.

Основное назначение блока питания – формирование напряжения питания, которое необходимо для функционирования всех устройств в составе ПК. Основные напряжения питания компонентов: +12, +5, +3,3 В. Существуют также дополнительные уровни напряжения: –12 и –5 В. Также блок питания осуществляет гальваническую развязку между сетью ~220 В и компонентами компьютера. Это необходимо для устранения токов утечек, например, чтобы корпус ПК не бился током, а также препятствует возникновению паразитных токов при сопряжении устройств.

Типы электрических соединителей источников питания

На задней стенке блока питания размещается разъём для подключения сетевого кабеля и выключатель. Опционально могут присутствовать и другие элементы: индикаторы сетевого напряжения или состояния работы блока питания, кнопки управления режимом работы вентилятора и кнопка переключения входного сетевого напряжения ~110 / 220 В.

На задней стенке всё реже размещают кулер, выдувающий из блока питания нагретый воздух. Всё чаще вентилятор размещают в верхней части блока питания из-за большего пространства для установки вентилятора, что позволяет установить большой и тихий активный элемент охлаждения (см. рисунок 2.2). На некоторых блоках питания устанавливают оба вентилятора (сверху и сзади).

К передней стенке подключается провод с разъёмом подключения питания материнской платы. В модульных блоках питания он, как и другие провода, подключается через разъём. Ниже на рисунке 2.5 указано назначение контактов всех основных разъёмов стандарта *ATX*.

Следует отметить, что каждое напряжение имеет свой цвет провода: +12 В – жёлтый; +5 В – красный; +3,3 В – оранжевый; *GND* – чёрный. Для остальных напряжений цвета проводов у каждого производителя могут отличаться.

Основные разъёмы для подключения блока питания к компонентам ПК представлены на рисунке 2.6 (*а* – привод гибких дисков (*FDD*); *б*, *в* – устройства энергонезависимой памяти (*HDD*, *SSD*, *CD*/*DVD*-приводы); *г*, *д*, *е* – графический адаптер; *ж* – процессор; *з* – системная плата).

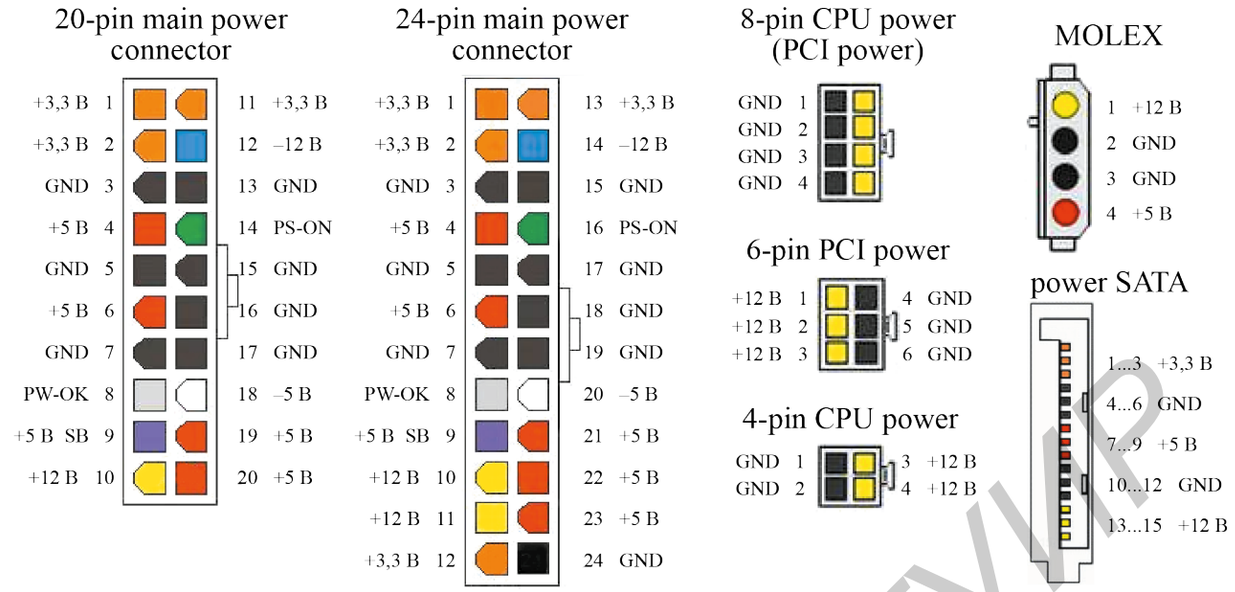


Рисунок 2.5 – Назначение выводов разъёмов питания стандарта *ATX*

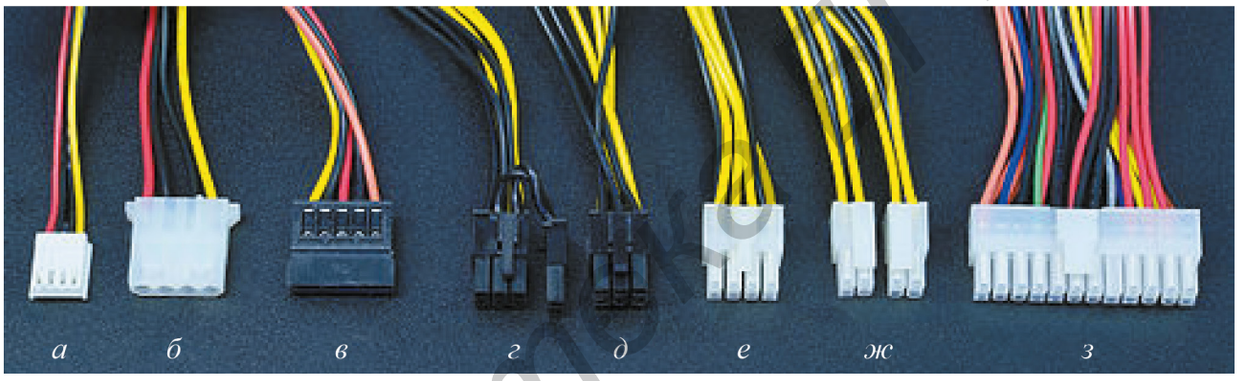


Рисунок 2.6 – Разъёмы питания стандарта *ATX*

Оценка необходимой мощности блока питания

Общая мощность блока питания (P) – самая распространённая характеристика, отображающая сумму мощностей каждого канала (уровня напряжения). Самым нагруженным каналом является уровень +12 В.

Необходимая (рассчитываемая) мощность блока питания состоит из пиковой мощности ($P_{ПИК}$), обусловленной максимальным потреблением аппаратной конфигурации, запаса пиковой мощности для будущей реконфигурации компьютера ($P_{ЗАП}$) и запаса мощности на подключение съёмных носителей информации ($P_{ИНФ}$).

Пиковая (максимальная) мощность – мощность, которую способен выдать блок питания в моменты максимальной нагрузки на питаемые устройства (компоненты компьютера). Такие моменты случаются редко и не являются продолжительными: включение компьютера (раскручивание шпинделей жёстких дисков, загрузка операционной системы, продув вентиляторов), работа игр в момент сложного графического рендеринга и др. Пиковая мощность составляет 120…130 % номинальной мощности ($P_{НОМ}$):

$$P = P_{ЗАП} + P_{ПИК} + P_{ИНФ} = P_{ЗАП} + 1{,}3 \cdot P_{НОМ} + P_{ИНФ}.$$

Номинальная мощность – гарантированная мощность, которую способен выдавать блок питания в течение длительного времени. Для идеального соотношения цены и возможностей БП номинальная мощность должна совпадать с потребляемой мощностью компьютера и быть равна

$$P_{\text{НОМ}} = \sum_i P_{\text{УСТР}\,i} \ ,$$

где $P_{\text{УСТР}\,i}$ – мощность отдельно взятого компонента (устройства) системного блока персонального компьютера.

Ниже приведён список устройств, которые можно рассматривать как отдельные компоненты в составе ПК с точки зрения энергопотребления:

– центральный процессор ($P_{\text{ЦП}}$) и его система охлаждения ($P_{\text{ЦП ОХЛ}}$);
– видеокарта ($P_{\text{ВК}}$) и её система охлаждения ($P_{\text{ВК ОХЛ}}$);
– оперативная память ($P_{\text{ОЗУ}}$) и её система охлаждения ($P_{\text{ОЗУ ОХЛ}}$);
– жёсткие диски ($P_{\text{ПЗУ}}$), включая *HDD* и *SSD* и их охлаждение;
– общая система охлаждения ($P_{\text{ОХЛ}}$);
– материнская плата ($P_{\text{МП}}$) и её система охлаждения ($P_{\text{МП ОХЛ}}$);
– приводы оптических носителей информации и картридеров ($P_{\text{ВН ИНФ}}$);
– платы расширения, адаптеры, игровые контроллеры и др.

Наибольшее энергопотребление в современном компьютере имеют центральный процессор ($P_{\text{ЦП}}$) и видеокарта ($P_{\text{ВК}}$), поэтому в первую очередь рассмотрим энергопотребление данных компонентов.

Потребление центрального процессора. Производители по тем или иным причинам не указывают потребляемую мощность процессоров. Возможно, это связано с маркетинговой политикой предприятий или же со сложностью расчёта этой величины (потребляемая мощность меняется с течением времени в зависимости от вычислительной нагрузки на процессор). Однако требования к системам охлаждения заставляют их приводить в технической документации характеристику, описывающую **требования к теплоотводу** (***thermal design power***, ***TDP***) и показывающую, на какую среднюю тепловую мощность должна быть рассчитана система охлаждения.

Несмотря на то, что *TDP* не отражает точные характеристики самого процессора, а является лишь величиной, способствующей организации охлаждения, от этого значения можно отталкиваться в расчётах. Это объясняется тем, что напряжения и токи информационных сигналов, принимаемые и выдаваемые процессором, крайне малы и компенсируют друг друга по входам-выходам. А значит, можно утверждать, что практически вся потребляемая мощность по питанию выделяется на процессоре в виде тепла:

$$P_{\text{ЦП}} = TDP.$$

Следует учесть, что не все производители вкладывают одинаковый смысл в этот параметр. Некоторые из них определяют *TDP* как максимальную, а некоторые ошибочно как среднюю. Для определения средней рассеиваемой мощности существует отдельная характеристика – ***SDP*** (***scenario design power***). Некоторые

производители называют её ***ACP*** (***average CPU power***). В этом случае необходимо предусмотреть увеличение значения на 20…30 % для вычисления потребляемой мощности:

$$P_{\text{ЦП}} = 1{,}3 \cdot SDP.$$

Значения *TDP* и *SDP* можно посмотреть в технических характеристиках к процессорам на сайтах производителей.

Потребление мощности систем воздушного охлаждения (кулеров) зависит от потребляемого ими тока (напряжение составляет +12 В). У большинства кулеров надёжных производителей в технической документации приведены значения максимального входного тока (или максимальной потребляемой мощности). Как правило, для современных конструкций с гидродинамическим подшипником ток варьируется в диапазоне 0,1…0,4 А, что определяет потребляемую мощность как $U \times I = 1\ldots5$ Вт на каждый мотор:

$$P_{\text{ОХЛ}} = 12(\text{В}) \cdot \sum_{i=1}^{n} I_{\text{МОТ}\,i},$$

где n – количество моторов в системе охлаждения, шт.; $I_{\text{МОТ}\,i}$ – максимальный ток потребления i-го мотора, А.

Системы жидкостного охлаждения требуют большего питания, т. к. способствуют переносу более плотного вещества – хладагента. Потребляемая мощность таких систем складывается из мощности насоса и мощности подсистемы воздушного обдува радиаторов. Расчёт потребляемой мощности такой системы повторяет расчёт систем воздушного охлаждения, но токи теперь определяются отдельно для насоса (обычно 0,3…0,8 А) и отдельно для подсистемы обдува (обычно 0,1…0,4 А). Среднее значение потребляемой мощности здесь составляет 4…14 Вт в зависимости от назначения и конструкции самих систем.

При принудительном увеличении рабочей тактовой частоты процессора – **разгоне** (***overclock***) – любыми из возможных методов увеличивается и потребляемая им мощность. Точного закона, описывающего эту зависимость, не существует. Однако много источников ссылается на формулу

$$P_{\text{ЦП}} \sim C_{\text{ДИН}} \cdot U^2 \cdot F,$$

где $C_{\text{ДИН}}$ – динамическая ёмкость процессора – константа, определяемая числом транзисторов на кристалле и активностью их переключения; U – напряжение питания процессора, В; F – тактовая частота работы процессора, Гц.

Анализируя физику работы процессора, можно вывести зависимость $F \sim U$. Это значит, что $P \sim F^3$. На основе этого выражения несложно рассчитать увеличение потребления мощности процессором, зная увеличение его тактовой частоты.

Потребление графического адаптера. В зависимости от типа графического адаптера потребление мощности может значительно различаться. Потребление интегрированных графических систем (видеокарта в процессоре) уже включено в значение мощности центрального процессора и не считается отдельно. Дискретный графический адаптер представляет собой сложный модуль, ко-

торый имеет собственный графический процессор, память и другие компоненты. Расчёт потребляемой мощности для него ничем не проще расчёта для всего компьютера. Поэтому производители видеокарт всегда указывают в технических характеристиках одну из двух характеристик (а иногда и обе):

– рекомендуемую мощность блока питания для компьютера, в составе которого используется графический адаптер ($P_{\text{РЕК БП}}$);

– номинальную потребляемую мощность ($P_{\text{ВК НОМ}}$).

Первая характеристика не является значительно информативной и обычно указывается для неопытных пользователей, не способных самостоятельно оценить общую мощность системы. Вторая же, наоборот, отражает нужную для расчёта среднюю потребляемую мощность устройства. Стоит отметить, что, как и большинство устройств в составе ПК, графический адаптер характеризуется пиковой мощностью, которая составляет 120…130 % от номинальной:

$$P_{\text{ВК}} = 1{,}3 \cdot P_{\text{ВК НОМ}}.$$

В расчёте нужно использовать именно её.

Система охлаждения видеокарты рассчитывается аналогично ранее рассмотренным системам охлаждения.

Потребление оперативной памяти. Потребление оперативной памяти зависит от количества модулей (планок) и числа микросхем памяти в этих модулях. Как правило, при отсутствии операций записи, потребляемая мощность невысока. Если на плате памяти расположено 4 микросхемы, её потребляемая пиковая мощность составляет 3...4 Вт, 8 микросхем – 5...6 Вт, 16 микросхем – 10...12 Вт. В скоростных типах памяти, предназначенных для игр или высоконагруженных приложений (*PC3-16000* и больше), пиковая мощность может доходить до 20 Вт на планку.

Так как потребление мощности оперативной памяти определяется только микросхемами (больше активных элементов на плате нет), то точное значение потребления можно определить из документации производителя микросхем, на базе которых собирают интересующую оперативную память.

Потребление материнской платы. Материнская плата представляет собой печатную плату, посредством которой объединяются остальные компоненты системного блока персонального компьютера. Говорить о точном значении потребляемой ею мощности неуместно. Даже отключение аудиосистемы от материнской платы (а значит, уменьшение нагрузки на выходном каскаде) снижает потребляемую ею мощность.

Для оценки максимальной мощности необходимо задействовать в расчёте все её компоненты (чипсет, аудиоподсистема, сетевой адаптер, контроллер *USB* и т. п.). Чаще всего за их работу отвечают соответствующие микропроцессоры и микроконтроллеры (северный мост, южный мост, аудиоконтроллер, контроллер интерфейса *USB* и т. д.). Каждый такой микропроцессор или микроконтроллер имеет свою максимальную потребляемую мощность. Интересующая мощность будет определяться их суммой. Ниже приведён общий список микросхем, которые могут входить в расчёт:

– северный мост материнской платы (обязательно);
– южный мост материнской платы (обязательно);
– контроллер аудиоподсистемы (при использовании);
– контроллер компьютерной сети (при использовании);
– контроллер ввода-вывода, определяющий работу скоростных интерфейсов, если они не поддерживаются *CPU* напрямую (обязательно);
– микросхема *BIOS* (обязательно);
– контроллеры интерфейсов (как правило, отдельный на каждый интерфейс) подключения внешних накопителей данных (при использовании);
– стабилизаторы и преобразователи напряжения (обязательно);
– нерассмотренные выше вспомогательные микросхемы (при наличии), суммарная мощность которых обычно не превышает 10 Вт.

При использовании в расчёте указанных компонентов для подавляющего большинства материнских плат результирующее значение потребляемой мощности ляжет в диапазоне 15...30 Вт. Исключение составят материнские платы производства старше 5 лет, платы специального назначения, экспериментальные платы, не вышедшие в массовое производство и платы для «разгона». Большинство инженеров по сборке персональных компьютеров, не вникая в точный расчёт, просто руководствуется верхней границей этого диапазона.

Потребление накопителей информации. *HDD*-диски имеют различные характеристики потребления питания в моменты ожидания, чтения и поиска информации. Как правило, эти численные параметры указаны в спецификации производителя. Стоит отметить, что даже пиковая мощность жёсткого диска в момент раскрутки шпинделя редко превышает 13...15 Вт.

SSD-диски представляют собой набор микросхем *flash*-памяти, управляемые одним микроконтроллером. Оценка их потребляемой мощности так же, как и в случае материнской платы, сводится к сумме потребляемых мощностей микросхем в составе *SSD*.

Точное измерение мощности

Стоит отметить, что любые попытки рассчитать вручную точное значение мощности, потребляемой системой в течение длительного времени, не дадут результата. Это объясняется тем, что компьютер – сложная электронная система, которая в различные моменты времени задействует различные свои компоненты, а значит, и потребляет различную мощность.

На первый взгляд решением вопроса точной оценки потребляемой компонентами компьютера мощности может стать бытовой измеритель мощности (рисунок 2.7), который подключается в разрыв электрической цепи между блоком питания и бытовой розеткой (рисунок 2.8). Однако, работая даже в режиме холостого хода, блок питания потребляет электроэнергию, близкую к своему номинальному значению, что не позволяет использовать эту схему для получения интересуемых значений. Поэтому оптимальная схема для оценки мощности, потребляемой компонентами компьютера, изображена на рисунке 2.9.

Рисунок 2.7 – Прибор для измерения потребляемой мощности

Рисунок 2.8 – Схема измерения мощности ПК

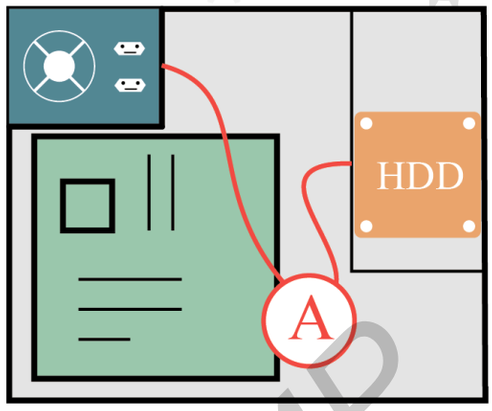


Рисунок 2.9 – Схема измерения мощности отдельного компонента ПК

2.3 Практическая часть лабораторной работы

Для выполнения практической части лабораторной работы необходимо:

1 Ознакомиться с выданным преподавателем настольным персональным компьютером, производителями и моделями компонентов, входящих в его состав, и оборудованием для измерения мощности. По желанию преподавателя задание может быть выдано не примером физического компьютера, а перечнем его компонентов.

2 Пользуясь различными калькуляторами мощности (например [2, 3] и др.), выполнить не менее трёх различных расчётов необходимого блока питания для выданного персонального компьютера или перечня компонентов.

3 Выполнить уточнённый расчёт мощности вручную по описанному алгоритму, оценив потребление энергии каждым контроллером на системной плате.

4 С помощью выданного преподавателем измерительного инструмента оценить потребляемую мощность выданного персонального компьютера или отдельных его компонентов в моменты загрузки операционной системы, ожидания, выполнения расчётов, игры.

5 Выполнить сравнение полученных результатов и на их основании оформить отчёт по лабораторной работе.

2.4 Содержание отчёта

Отчёт о выполненной работе должен содержать:

1 Титульный лист. Цель лабораторной работы. Задание преподавателя.

2 Снимки страниц *online*-калькуляторов мощности, позволяющие понять, как были получены результирующие значения.

3 Уточнённый расчёт потребляемой мощности.

4 Выводы о причинах наличия или отсутствия различий в значениях мощности, полученных различными способами.

2.5 Контрольные вопросы

1 Можно ли и каким образом осуществить питание персонального компьютера от двух различных источников питания, если по отдельности каждому из них не хватает мощности?

2 Можно ли, имея в руках обычный мультиметр, измерить потребление мощности персональным компьютером?

3 Как, зная только модель контроллера порта *DisplayPort*, выяснить потребляемую им электрическую мощность?

4 Устройства какой мощности можно без последствий подключать к порту *USB* 3.0? В чём могут проявляться эти последствия?

2.6 Литература

1 Современные блоки питания ATX и их характеристики [Электронный ресурс]. – Режим доступа : http://ru.gecid.com/power/sovremennyee_bloki_pitaniya_atx_i_ih_harakteristiki/.

2 eXtreme Power Supply Calculator [Электронный ресурс]. – Режим доступа : http://outervision.com/power-supply-calculator.

3 Power Supply Calculator [Электронный ресурс]. – Режим доступа : http://www.coolermaster.com/power-supply-calculator/.

ЛАБОРАТОРНАЯ РАБОТА №3
ВВЕДЕНИЕ В СТРУКТУРИРОВАННЫЕ КАБЕЛЬНЫЕ СИСТЕМЫ ИНФОРМАЦИОННО-КОМПЬЮТЕРНЫХ СЕТЕЙ

3.1 Цель работы

Ознакомиться с понятием структурированной кабельной системы (СКС) локальной информационно-компьютерной сети, а также получить практические навыки монтажа отдельных её элементов.

3.2 Краткие теоретические сведения

Структурированная кабельная система (***structured cabling system***, ***SCS***) – это набор кабелей, кроссового оборудования и коммутационных элементов (разъёмов, коннекторов), а также правила их совместного использования, которые позволяют создавать надёжные, удобные и легко масштабируемые информационно-компьютерные сети. Зачастую для определения СКС используют термин **пассивное сетевое оборудование**.

Современные здания содержат множество кабелей разного типа и назначения. Их состав предусматривает как кабели для обеспечения подачи электричества, так и кабели для слаботочных телекоммуникационных инженерных систем (телефонной связи, сигнализации, кабельного телевидения, сети и т. п.).

Применительно к информационно-компьютерным сетям СКС предусматривает проектирование, построение и эксплуатацию кабельных слаботочных систем в соответствии с принципами и нормами, заложенными в разработанных стандартах.

В 1983 г. *AT&T* установила первую структурированную кабельную систему. В 1991 г. на телекоммуникационные кабельные системы американскими Ассоциацией электронных отраслей промышленности (*EIA*) и Ассоциацией индустрии связи (*TIA*) был введён стандарт *EIA/TIA 568*. В условиях практического отсутствия национальных альтернатив стандарт *EIA/TIA 568A* широко распространился по миру. В настоящее время он пересмотрен, дополнен и разделён на несколько отдельных стандартов. На его основе были разработаны и приняты международные (*ISO/EIC 11801*) и европейские (*EN 50173*) стандарты, которые в настоящее время находят всё более широкое применение на практике в Республике Беларусь. Актуальные редакции стандартов применительно к сетям на основе медных кабелей называются:

– ISO/IEC 11801: Ed 2.2:2011-06 Information technology – Generic cabling for customer premises;

– ISO/IEC 24764: Ed 1.0:2010-04 Generic Cabling Systems for Data Centres;

– ANSI/TIA/EIA 568-C.0:2009 Generic Telecommunications Cabling for Customer Premises;

– ANSI/TIA/EIA 568-C.1:2009 Commercial Building Telecommunications Cab ling Standard;

– ANSI/TIA/EIA 568-C.2:2011 Balanced Twisted-Pair Telecommunications Cab ling and Components;

– EN 50173:2007 Information Technology – Generic cabling systems.

Цель указанных стандартов – дать рекомендации (в ряде случаев обязать) по созданию структурированной кабельной системы, которая может поддерживать любые приложения передачи данных и являться частью инфраструктуры офиса или промышленного здания.

Мировыми лидерами по производству компонентов СКС являются *AMP Netconnect*, *Eurolan*, *Legrand*, *Molex PN*, *ExaLan Plus* и *Siemon*.

Признаки структурированных кабельных систем

Признаками структурированных кабельных систем являются:

1 Структурированность.

По назначению структурированную сеть принято разделять на подсистемы (организовывать иерархию или, как это видно из названия, структуру). Ряд международных стандартов разделяет СКС на три подсистемы: магистраль комплекса, **вертикальную подсистему** (магистраль здания) и **горизонтальную подсистему**. Такое разделение поддерживают не все стандарты. На практике разделение мнений настолько значительно, что в проспектах ряда компаний можно обнаружить четыре, пять, восемь и даже девять подсистем.

2 Универсальность (унифицированность).

Универсальность в СКС достигается за счёт соответствия стандартам, которые позволяют перейти от частных (систем произвольной реализации) к открытым системам с унифицированными параметрами, поддерживающими работу оборудования любых производителей.

3 Избыточность.

Классическая структурированная кабельная система монтируется на этапе строительства или капитального ремонта здания и должна служить без изменений до следующего капитального ремонта (обычно 15...20 лет). Достигается это путём выполнения монтажа системы не из расчёта на существующие потребности, а исходя из общих требований нормативов (как правило, со значительным запасом). Следовательно, практически любые изменения организационной структуры заказчика не требуют модернизации СКС.

Достоинства избыточности выражаются в возможности строителям создавать СКС прежде, чем станут известны требования пользователей, и обеспечивать длительный срок службы всей телекоммуникационной инфраструктуры здания. Недостатком же является экономическая неэффективность.

Иерархия структурированной кабельной системы

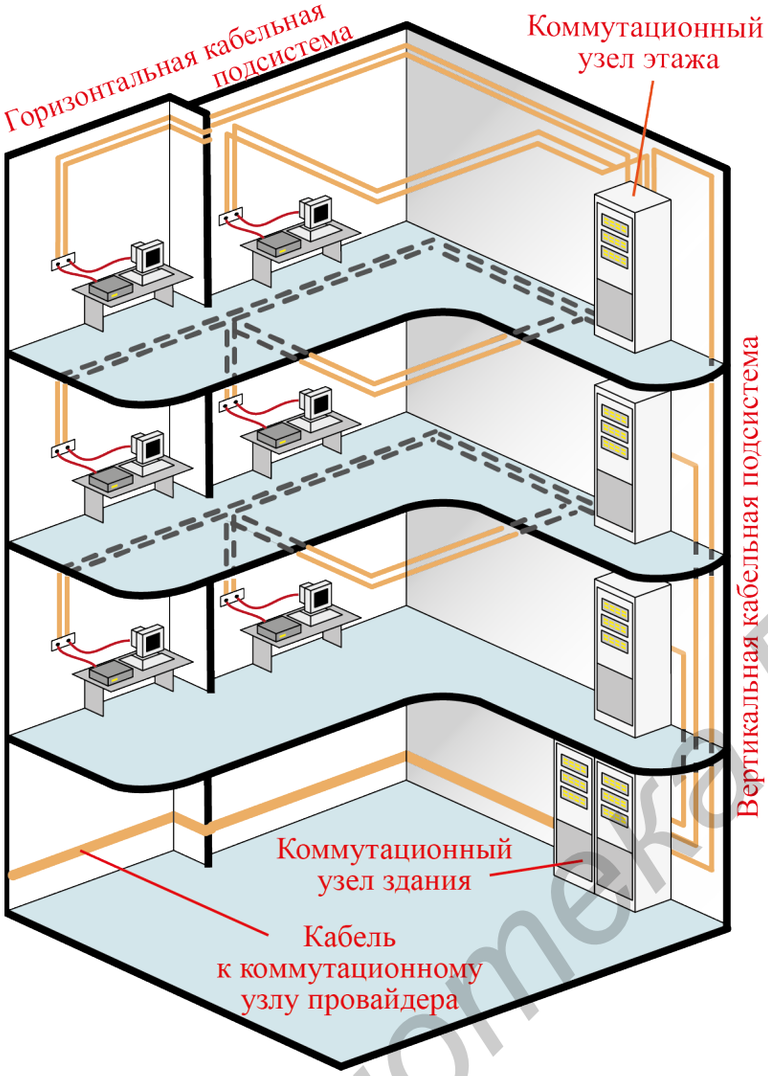


Рисунок 3.1 – Классическая топология СКС

Классическая иерархия СКС содержит следующие подсистемы (рисунок 3.1):

1 Магистраль комплекса, которая служит для объединения в сеть различных зданий. Как правило, она реализуется на оптоволоконном (реже медном) кабеле и позволяет соединять между собой здания, расположенные на расстоянии до нескольких десятков километров. Данная подсистема не рассматривается в рамках лабораторной работы.

2 Вертикальная подсистема (магистраль здания), соединяющая этажи здания, обеспечивает связь между распределительной панелью здания и панелями этажей. Она должна включать кабель, установленный вертикально между этажными панелями, главную или промежуточную панель в многоэтажном здании, а также кабель, установленный горизонтально между панелями в длинном одноэтажном здании. Данная подсистема также не рассматривается в лабораторной работе.

3 Горизонтальная подсистема, которая прокладывается между этажной распределительной панелью и телекоммуникационной розеткой на рабочем месте. Каждый этаж здания оснащается собственной горизонтальной подсистемой.

Горизонтальная подсистема СКС обычно включает в себя:

– коммутационный узел этажа, в котором размещается активное сетевое оборудование и коммутационные панели;

– кабельную систему, соединяющую коммутационные панели и розетки этажа;

– соединительные кабели (*patch cord*), которые связывают конечные устройства с коммутационными розетками, а также которые образуют структуру сети на коммутационной панели и соединяют через неё кабельную систему этажа с активным сетевым оборудованием.

Коммутационный узел этажа – место расположения коммутационного оборудования этажа, объединяющее все линии горизонтальной подсистемы. Коммутационный узел этажа монтируется в монтажном шкафу (рисунок 3.2) или стойке. В нём размещаются коммутационные панели, оборудованные разъёмами для подключения сетевых кабелей, и активное сетевое оборудование (рисунок 3.3). Кабели, соединяющие коммутационный узел и розетки этажа, монтируются одним концом на коммутационной панели, а другим – на коммутационной розетке этажа. Эти кабели прокладываются от коммутационного узла ко всем точкам этажа, в которых необходимо подключить компьютеры и другое оборудование. В каждой точке подключения монтируется коммутационная розетка для подключения к сети компьютеров и другого оборудования.

Рисунок 3.2 – Настенный коммутационный узел (шкаф)

Коммутационные панели и коммутационные розетки снабжены одинаковыми разъёмами для подключения соединительных кабелей. Разъёмы на панелях и розетки маркируются для идентификации соединений.

Кабельная система представляет собой собственно кабели и другие компоненты для их крепления и укладки (короба, стяжки, колодцы и т. п.).

В настоящее время в горизонтальных подсистемах чаще всего используется неэкранированная (*unshielded twisted pair*, *UTP*), фольгированная (*foiled twisted pair*, *FTP*) или экранированная (*shielded twisted pair*, *STP*) категории *5* или *5e* «витая пара» с одножильными (монолитными) проводниками. Такой кабель представляет собой собранные в одной изоляционной оболочке четыре скрученных пары медных изолированных проводников и может использоваться в сетях *Ethernet* стандартов *10BaseT*, *100BaseTX*, *100BaseT4*, *1000BaseT*.

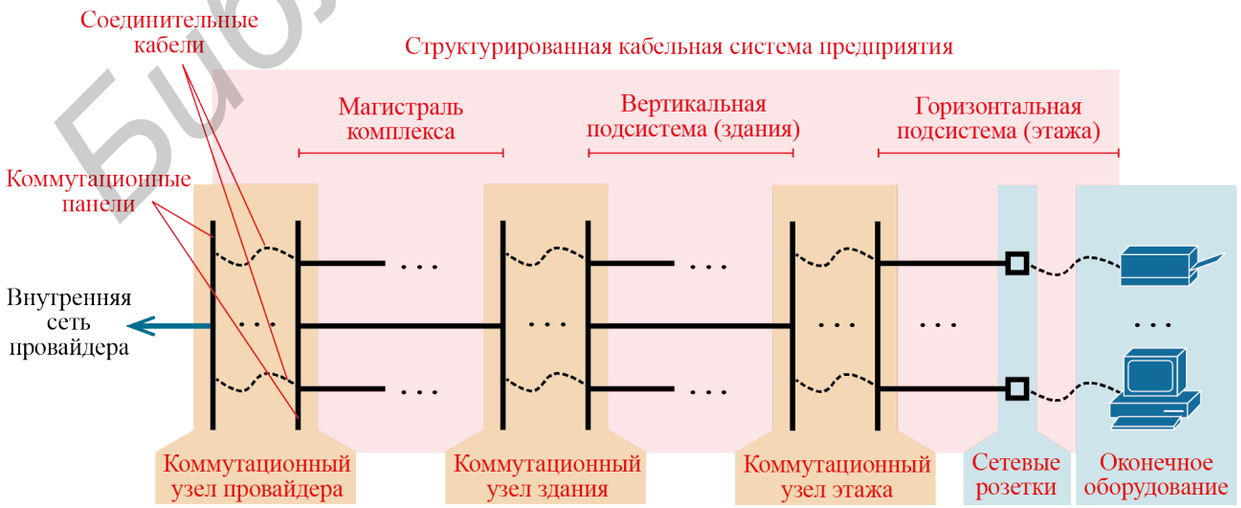


Рисунок 3.3 – Общая схема коммутационного узла

Каждая пара проводников «витой пары» маркируется своим цветом. При этом один проводник пары целиком окрашен в соответствующий цвет (этот проводник называется основным), а другой проводник окрашен в белый цвет и имеет полоски соответствующего цвета (этот проводник называется дополнительным). Стандартные цвета пар – зеленый, оранжевый, синий, коричневый. Существуют стандартные схемы расположения (разводки) проводников по цветам в соединительных разъёмах.

Экранированная «витая пара» часто используется для вертикальной подсистемы или в местах, где существует значительный уровень внешних помех, а также в целях обеспечения более низкого излучения в среду при реализации высокоскоростных сетей. Для всех линий кабеля экранированной «витой пары» в одной точке экрана обеспечивается надёжное его заземление.

При использовании «витой пары» категории *5* длина горизонтальных кабелей должна составлять не более 100 м. В коммутационных панелях и розетках кабель монтируется в 8-контактные разъёмы *RJ-45* (*8P8C* – в соответствии со стандартом). При монтаже кабеля необходимо, чтобы кабель на панели и в соответствующей розетке был разведён по одной и той же схеме.

Соединительные кабели (***patch cord***) служат для подключения конечного (пользовательского) оборудования и для создания структуры сети на коммутационной панели. Они представляют собой кабели, снабжённые с двух сторон вилками для подключения к разъёмам коммутационных панелей, розеток и сетевого оборудования. Для изготовления соединительных кабелей используется многожильный (гибкий) кабель *UTP*. Разводка проводников в разъёмах также производится по стандартным схемам.

Монтаж горизонтальной подсистемы СКС

Монтаж кабеля на коммутационной панели

Коммутационная панель (*patch panel*) имеет две стороны: лицевую и обратную. На её лицевой стороне располагаются разъёмы *RJ-45* для подключения обжатых кабелей (рисунок 3.4). С обратной стороны панели, как правило, выведены универсальные врезные контакты *IDC* (*insulation displacement contact*) для монтажа необжатой «витой пары». Разъёмы и группы контактов пронумерованы. Контакты также имеют цветовую маркировку для визуального контроля правильности реализации той или иной схемы разводки проводников кабеля. Кабель монтируется на контактах *IDC* при помощи специального инструмента (рисунок 3.5) и стяжек по алгоритму, представленному на рисунке 3.6.

Часто коммутационная панель оснащается задней кабельной поддержкой (металлической планкой для фиксации кабеля) и отдельными проводами заземления, которые замыкаются на общий контур коммутационного шкафа. Для удобства разделения кабелей в два жгута на рынке присутствуют угловые версии панелей.

Панели устанавливаются в специальную стойку или настенный шкаф (см. рисунок 3.2).

Рисунок 3.4 – Коммутационная панель

Рисунок 3.5 – Инструмент врезки кабеля

Монтаж кабеля до коммутационных розеток этажа

Монтаж горизонтального кабеля производится в зависимости от планировки этажа и здания в коммуникационных каналах, монтажных коробах, гофротрубах или каким-либо иным подобным способом. При прокладке кабеля не допускается образование перегибов кабеля с радиусом, меньшим допустимого спецификацией, и «петель», растяжка кабеля (укладка с механическим сильным натяжением), а также нарушение внешней изоляции. Недопустимо укладывать в одном коробе (кабель-канале) силовые кабели и кабели для слаботочных сетей. Фиксация кабеля в желобах часто осуществляется с помощью стяжек, хомутов или других фиксирующих элементов. Каждый кабель маркируется на обоих концах (ставится идентификатор с записью в документацию для дальнейшей эксплуатации).

Монтаж кабеля на коммутационных розетках этажа

Коммутационная розетка представляет собой смонтированный в пластмассовом корпусе разъём *RJ-45* для подключения соединительного кабеля. Розетка закрыта крышкой. Под крышкой размещается группа из восьми контактов *IDC*, аналогичных контактам коммутационной панели. Контакты снабжены цифровой и (или) цветовой маркировкой. Горизонтальный кабель монтируется на этих контактах аналогично монтажу на коммутационной панели. В корпусе или крышке розетки имеется прорезь для вывода горизонтального кабеля. Розетка обычно закрепляется на стене комнаты или на корпусе мебели. Алгоритм монтажа кабеля на коммутационных розетках этажа представлен на рисунке 3.9.

Сборка сети коммутационного узла этажа

Сборка сети осуществляется при помощи соединительных кабелей, которые соединяют нужные разъёмы коммутационных панелей, розеток и активного сетевого оборудования.

В коммутационном узле размещаются панели, к которым подключаются горизонтальные кабели этажа (как описано выше), и активное сетевое оборудование. Если сеть имеет топологию типа «звезда», в которой все устройства подключаются к одному концентратору или коммутатору, то сборка заключается в соединении портов концентратора или коммутатора с разъёмами коммутационной панели. При более сложной структуре сети может понадобиться соединение между собой активных сетевых устройств и (или) соединение разъёмов панелей между собой (см. рисунок 3.3).

Алгоритм монтажа кабеля на контактах *IDC*:

1 Выбрать на лицевой панели разъём, к которому будет относиться кабель, и определить на обратной стороне соответствующую ему группу контактов, при необходимости промаркировать монтируемый кабель.

2 Снять изоляцию с монтируемого конца кабеля на длину, удобную для работы с проводниками (обычно это 2...3 см, рисунок 3.6, *а*), удалить пластиковый сепаратор или корд при его наличии (рисунок 3.6, *б*).

3 Развести проводники кабеля в соответствии с выбранной маркировкой контактов на обратной стороне коммутационной панели (рисунок 3.6, *в*) и вставить соответствующие проводники в прорези соответствующих контактов (рисунок 3.6, *г*). При этом важно по-максимуму сохранить целостность свития пар.

4 Установив ножи врезного инструмента перпендикулярно коммутационной панели, до упора надавить на них (рисунок 3.6, *д*). Нож *IDC* контакта прорежет изоляцию и врежется в металл жилы, что гарантирует надёжный контакт.

5 Если во врезной инструмент не встроен нож для автоматической обрезки неиспользуемой длины жил, то концы проводников следует срезать с помощью кусачек. Внешний вид правильно монтированного кабеля представлен на рисунке 3.6, *е*.

6 Смонтировав все кабели на обратной стороне коммутационной панели (рисунок 3.6, *ж*), необходимо установить поддержку кабеля и закрепить каждый кабель на нём кабельными стяжками (рисунок 3.6, *з*). Неиспользуемые концы кабельных стяжек удалить с помощью кусачек.

7 Аккуратно сгруппировать все кабели и закрепить их пластиковыми стяжками (рисунок 3.6, *и*). При жгутировании кабелей недопустимо их пережатие.

Рисунок 3.6 – Алгоритм монтажа кабеля на коммутационной панели

Подключение конечного оборудования к розеткам этажа

Подключение пользовательского оборудования к розеткам осуществляется при помощи соединительных кабелей, аналогичных кабелям для сборки структуры сети в коммутационном узле. Подключаемые устройства (сетевые платы компьютеров, принтеров и т. д.) снабжены разъёмом *RJ-45*. Этот разъём соединяется кабелем с аналогичным разъёмом коммутационной розетки.

Монтаж разъёмов *RJ-45* на кабеле *UTP-5cat* (обжим «витой пары»)

При обжиме «витой пары» придерживаются двух стандартных схем раскладки проводников по номерам контактов разъёмов *RJ-45* (*EIA/TIA 568*):

1 При соединении конечного оборудования с коммутатором/концентратором используется схема **прямого кабеля** (***straight through cable***). На обоих концах прямого кабеля раскладка проводников должна совпадать, при этом используют цветовую раскладку стандарта *EIA/TIA 568B* или *EIA/TIA 568A* (рисунок 3.7). Чаще используется схема раскладки по *EIA/TIA 568B*.

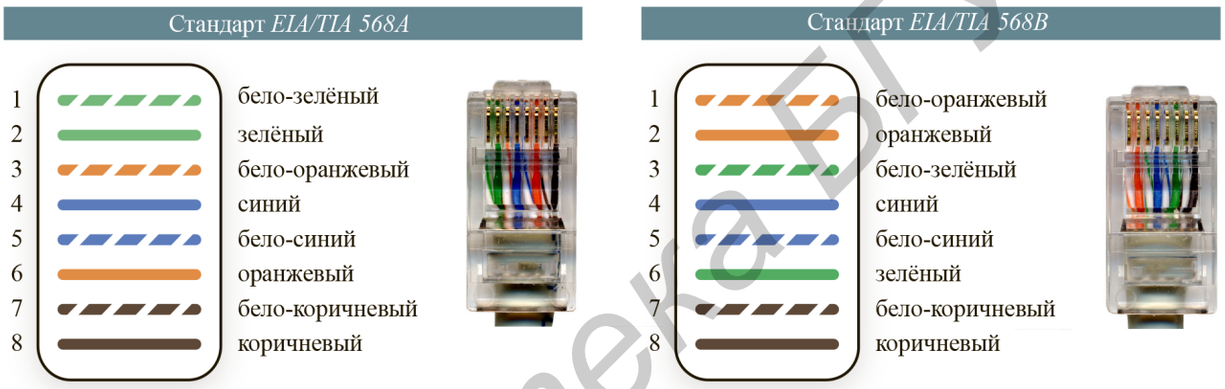


Рисунок 3.7 – Цветовая раскладка проводников прямого кабеля

2 При соединении конечного оборудования с маршрутизатором или другим конечным оборудованием, маршрутизатора с маршрутизатором или коммутатора с коммутатором используется схема **кроссоверного кабеля** (***crossover cable***). На одном конце такого кабеля используется схема *568A*, а на другом – *568B*. Цветовая раскладка такого кабеля представлена на рисунке 3.8.

Разъёмы *RJ-45* представляют собой полый прозрачный пластиковый корпус с фиксирующим замком, внутри которого расположено восемь подвижных металлических контактов. В необжатом разъёме контакты выходят за пределы корпуса, после обжима они вдавливаются внутрь, прорезая наружный изолирующий слой на проводниках, расположенных внутри кабеля «витая пара», и замыкаясь на токопроводящую жилу. Различают два типа разъёмов *RJ-45*: с контактной вставкой и без неё. Алгоритм обжима кабеля «витая пара» представлен на рисунке 3.10.

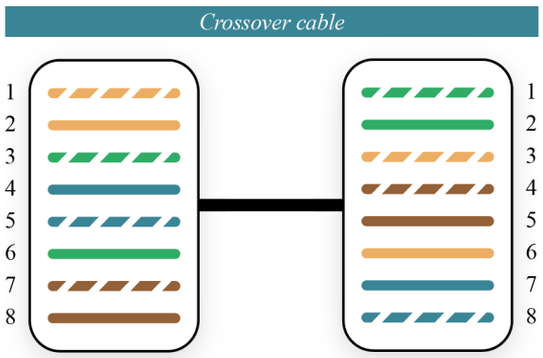


Рисунок 3.8 – Цветовая раскладка проводников кроссоверного кабеля *GigabitEthernet*

Алгоритм монтажа настенной розетки:

1 Так как розетка, как правило, поставляется в разобранном виде, необходимо присоединить печатную плату с коннектором, экранированным разъёмом и зажимным кольцом к основанию корпуса розетки (рисунок 3.9, *а*).

2 Нанести цифровые обозначения на кабель с помощью маркеров, соответствующих диаметру кабеля (рисунок 3.9, *б*).

3 Выполнить кольцевую подрезку оболочки кабеля и снять верхнюю изоляцию. Далее снять верхнюю изоляцию, не повредив экран из фольги, расположенный под оболочкой (рисунок 3.9, *в*).

4 Развернуть фольгу и загнуть её вниз на внешнюю изоляцию (рисунок 3.9, *г*). После чего следует обмотать дренажный провод вокруг фольги.

5 Далее необходимо срезать кусачками лишнюю часть фольги, оставив примерно 1,5 см для осуществления контакта (рисунок 3.9, *д*).

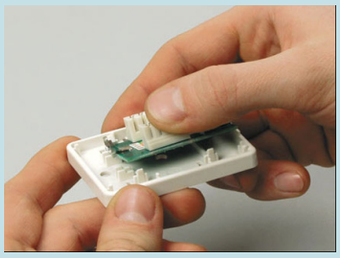
6 Завести кабель на *IDC*-модуль розетки (рисунок 3.9, *е*). Зажим модуля должен соприкасаться с фольгой и дренажным проводом. Для обеспечения наилучшего контакта необходимо зафиксировать кабель в зажиме плоскогубцами.

7 Витые пары разложить по цветовой маркировке на печатной плате модуля в соответствии с выбранным вариантом *T568A* или *T568B*. При этом изоляцию жил зачищать не нужно (рисунок 3.9, *ж*).

8 Устройством для врезки кабеля надавить на проводники до упора. При этом нож розетки прорезает изоляцию и врезается в металл жилы, что гарантирует надёжный контакт (рисунок 3.9, *з*).

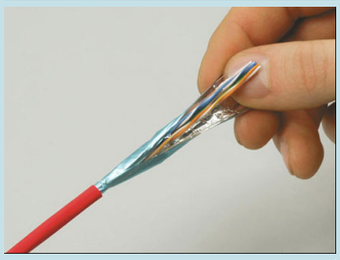
9 Присоединить крышку к основанию корпуса розетки. Внешняя розетка крепится на стену с помощью двусторонней клеевой площадки (рисунок 3.9, *и*).

а

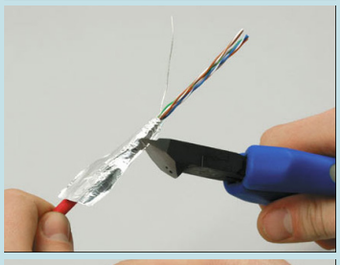


б

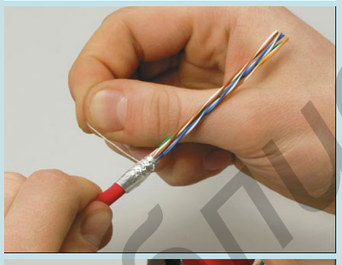
в



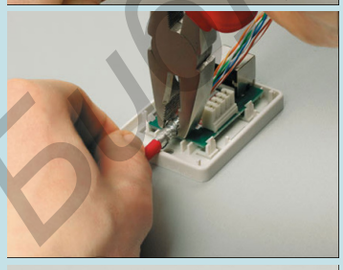
г



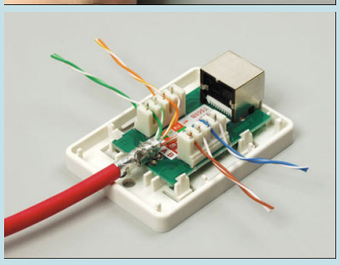
д



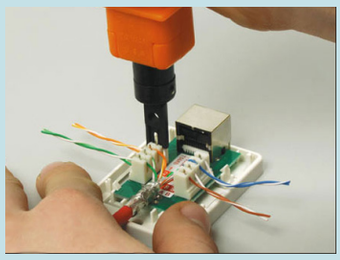
е



ж



з



и

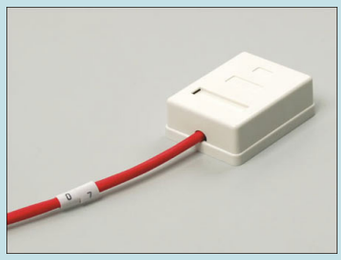


Рисунок 3.9 – Алгоритм монтажа настенной розетки

Алгоритм обжима кабеля *FTP-cat 5e*:

1 Отрезать ненужную часть кабеля, ограничив его до необходимой длины, надеть защитный колпачок, предотвращающий чрезмерные перегибы кабеля и обеспечивающий удобство коммутации (рисунок 3.10, *а*).

2 Снять изоляцию с монтируемого конца кабеля на длину, удобную для работы с проводниками (обычно это 2...3 см, рисунок 3.10, *б*). Развернуть фольгу и загнуть её вниз на внешнюю изоляцию и, добившись нужной длины, накрутить на неё дренажный провод.

3 Удалить защитную плёнку (рисунок 3.10, *в*).

4 Развести проводники кабеля вплоть до края оболочки, которому нужно придать плоскую форму, чтобы было удобно расположить пары в один ряд (рисунок 3.10, *г*).

5 С помощью кримпера подрезать проводники на расстояние около 14 мм от края оболочки кабеля (рисунок 3.10, *д*).

6 Разложить проводники в направляющих каналах вставки в том порядке цветов, который соответствует выбранной схеме. Во время укладки проводников вставку нужно продвинуть максимально близко к оболочке кабеля (рисунок 3.10, *е*).

7 Поместить вставку в корпус разъёма до упора. После того как вставка плотно вошла в корпус, при взгляде с торца должно быть видно, как жилы всех проводников равномерно упираются в торец разъёма.

8 Вставить конструкцию в кримпер и свести до упора ручки инструмента. В этот момент ножи прокалывают оболочку проводников и входят в металлическую жилу, создавая электрический контакт (рисунок 3.10, *ж*).

9 Сдвинуть на разъём защитный колпачок (рисунок 3.10, *з*) и протестировать кабель (рисунок 3.10, *и*).

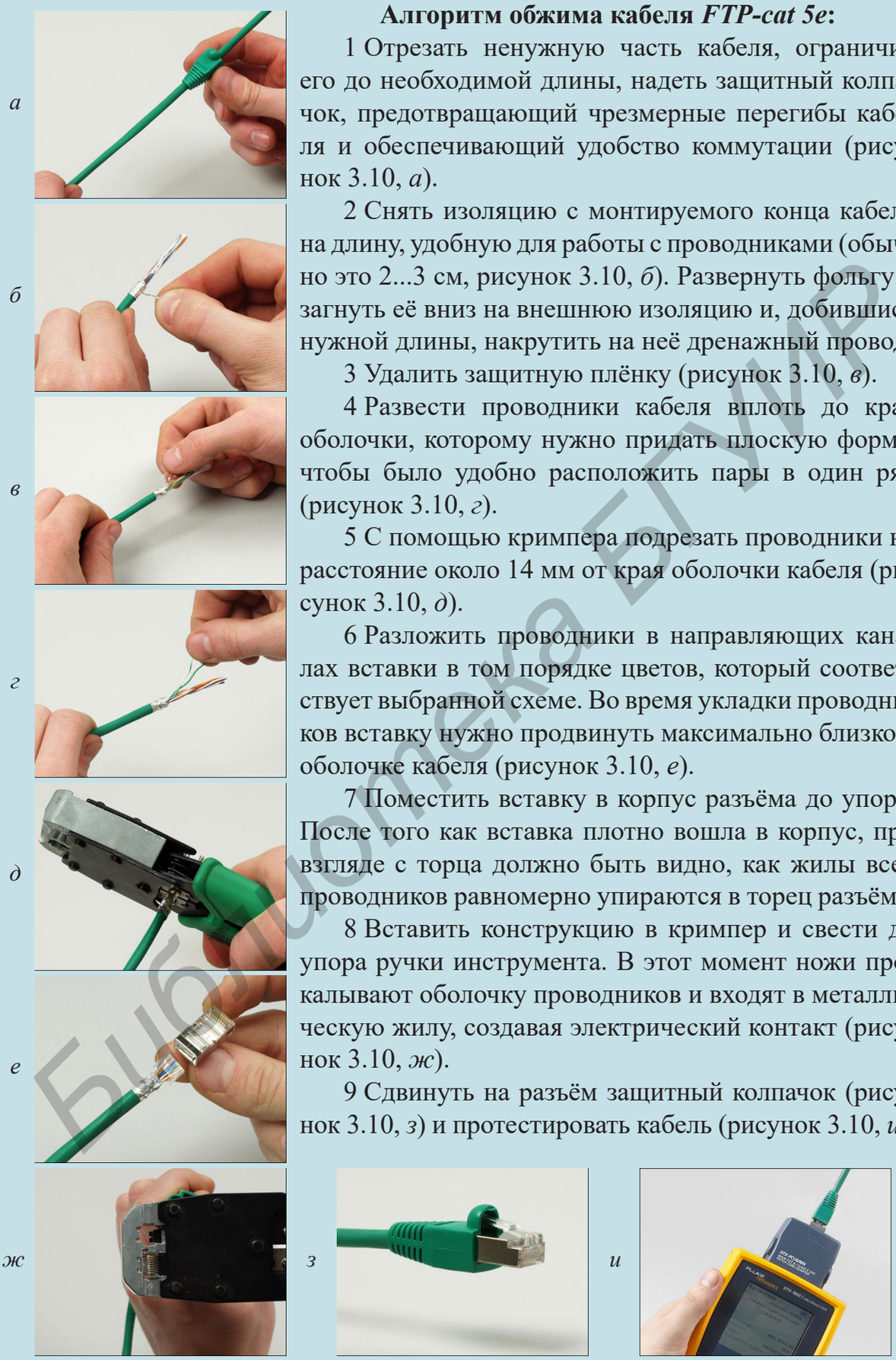


Рисунок 3.10 – Алгоритм обжима сетевого кабеля

Инструменты для монтажа и тестирования СКС

Кроме инструмента, указанного на рисунке 3.5, при монтаже СКС используются следующие типы инструментов начального уровня:

– универсальный инструмент для зачистки кабеля типа «витая пара» *UTP/STP*, коаксиального и телефонного кабеля (рисунок 3.11, *а*);

– многофункциональный инструмент для зачистки и обрезки кабелей (рисунок 3.11, *б*);

– боковые кусачки для обрезания проводников (рисунок 3.11, *в*);

– универсальный обжимной инструмент (кримпер) для разъёмов формата *RJ*, снятия внешней оболочки и изоляции, а также обрезки кабеля и проводников (рисунок 3.11, *г*);

– прибор для тестирования коаксиального, *USB* и сетевого кабеля на обрывы, короткие замыкания, целостность изоляции, перепутанные пары и жилы (рисунок 3.11, *д*).

а

б

в

г

д

Рисунок 3.11 – Инструменты для монтажа и тестирования СКС начального уровня

3.3 Практическая часть лабораторной работы

Для выполнения практической части лабораторной работы необходимо:

1 Ознакомиться с теорией монтажа СКС, особое внимание уделить алгоритмам монтажа кабеля на коммутационную панель и розетку, а также алгоритму обжима сетевого кабеля.

2 Ознакомиться с инструментами и материалами, выданными преподавателем и использующимися для выполнения лабораторной работы.

3 Выполнить монтаж одного конца кабеля на коммутационной панели или розетке согласно указанной преподавателем цветовой схеме. На другом конце кабеля смонтировать разъём *RJ-45*.

4 Изготовить соединительный кроссоверный кабель.

5 С помощью измерительного (тестирующего) оборудования проверить качество изготовленных компонентов СКС.

6 Соединить сетевые адаптеры персональных компьютеров учебной аудитории или ноутбуков студентов по схеме «Адаптер ПК1 ↔ *patch cord* ↔ *patch panel* ↔ Адаптер ПК2».

7 Настроить на компьютерах соединение типа «точка-точка», взяв нужную информацию из лабораторной работы №5.

8 Оформить отчёт по лабораторной работе.

3.4 Содержание отчёта

Отчёт о выполненной работе должен содержать:

1 Титульный лист. Цель лабораторной работы.

2 Описание выданных преподавателем инструментов и материалов.

3 Изображения (фотографии) выполненных компонентов СКС с указанием их составных частей, а также цветовой схемы монтажа кабеля.

4 Подробную схему соединения персональных компьютеров.

5 Вывод по работе, в котором необходимо указать наиболее сложный этап лабораторной работы.

3.5 Контрольные вопросы

1 Укажите известные стандарты проектирования и монтажа структурированных кабельных систем. Какими из этих стандартов руководствуются на территории Республики Беларусь?

2 Какие основные признаки имеет структурированная кабельная система? Позволительно ли спроектировать СКС, лишённую одного из перечисленных признаков?

3 Поясните понятие «пассивное сетевое оборудование». Какие элементы СКС относятся к пассивному сетевому оборудованию?

4 Назовите известные типы кабеля «витая пара» по конструкторскому исполнению. Какие категории «витой пары» соответствуют каждому из названных типов?

5 Перечислите инструменты для монтажа и тестирования СКС. Укажите, для каких операций они применяются.

6 Укажите цветовые схемы раскладки проводников «витой пары» для технологий *FastEthernet* и *GigabitEthernet*, а также поясните, когда применяется каждая из них.

3.6 Литература

1 Самарский, П. А. Основы структурированных кабельных систем / П. А. Самарский. – М. : Компания АйТи; ДМК Пресс, 2005. – 216 + 12 с.

2 Семёнов, А. Б. Проектирование и расчёт структурированных кабельных систем и их компонентов / А. Б. Семёнов. – М. : Компания АйТи; ДМК Пресс, 2003. – 416+16 с.

3 Инструкции по монтажу компонентов кабельной системы [Электронный ресурс]. – 2015. – Режим доступа : http://www.hyperline.ru/info/.

ЛАБОРАТОРНАЯ РАБОТА №4
СЛУЖБЫ INTERNET INFORMATION SERVICES MICROSOFT WINDOWS

4.1 Цель работы

Ознакомиться и приобрести навыки работы со службами *Internet Information Services* операционной системы *Microsoft Windows* для их использования при реализации информационных ресурсов предприятия.

4.2 Краткие теоретические сведения

Internet Information Services (***IIS***) – набор проприетарного программного обеспечения, распространяющегося в составе операционных систем семейства *Windows* и позволяющего реализовать набор веб-сервисов (сервер приложений).

Основным компонентом *IIS* является веб-сервер, поддерживающий протоколы *HTTP*, *HTTPS*, *FTP*, *POP3*, *SMTP*, *SNMP*. По данным компании *Netcraft* на февраль 2015 г. более 19,5 млн активных веб-сайтов обслуживаются веб-сервером *IIS*, что составляет 11 % от общего числа активных веб-сайтов. Это ставит *Microsoft IIS* на третье место после серверов *Apache* (50,64 %) и *nginx* (14,94 %).

Установка служб *IIS*

Установка служб *IIS* производится аналогично установке других компонентов операционной системы *Windows* через панель управления. Для начала установки необходимо открыть *Панель управления* → *Программы и компоненты* → *Включение или отключение компонентов Windows*. В появившемся диалоговом окне необходимо выбрать для установки *Службы IIS*. По умолчанию поддержка протокола *FTP* отключена, поэтому необходимо развернуть дерево компонентов и отметить компонент *FTP-сервер* (*Служба FTP* и *Расширяемость FTP*) для установки (рисунок 4.1).

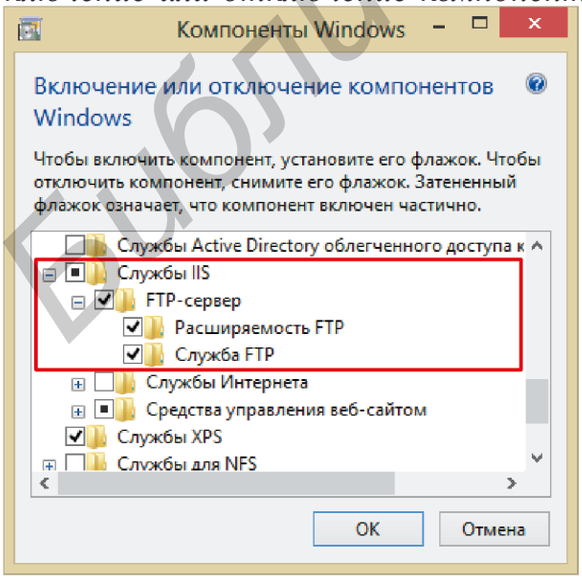


Рисунок 4.1 – Выбор компонентов для установки служб *IIS*

При добавлении поддержки протокола *FTP*, автоматически создаётся каталог с именем **C:\inetpub\ftproot**. Все файлы в этом каталоге будут отображаться, как будто они находятся в корневом каталоге вашего *FTP*-сервера. В дальнейшем рекомендуется сменить этот каталог на более удобный.

Прежде чем начать работу, нужно запустить *Диспетчер служб IIS* (*IIS Manager*). Его можно найти в меню *Пуск*. Конкретное расположение может зависеть от используемой версии *Windows* (обычно *IIS → Диспетчер служб IIS*). Ярлык программы будет располагаться в разделе *Все программы* (*Programs*) или *Администрирование* (*Administrative Tools*). Начальная страница *Диспетчера служб IIS* представлена на рисунке 4.2.

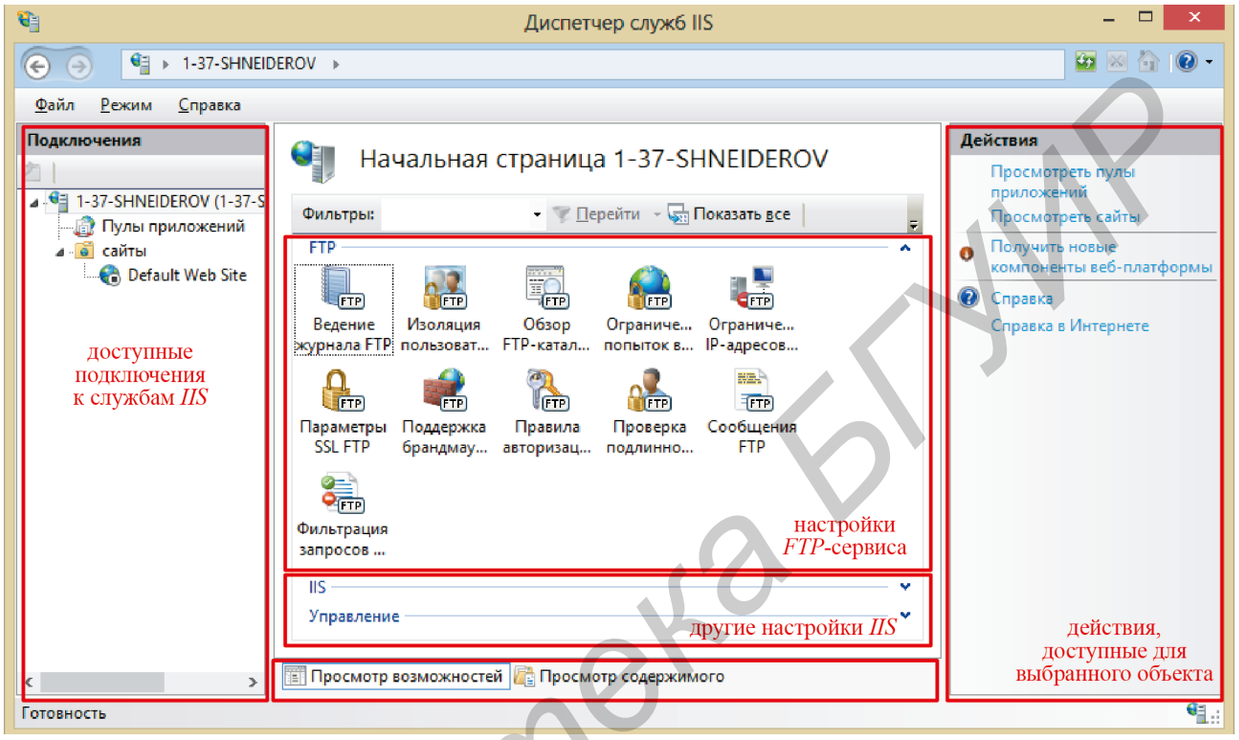


Рисунок 4.2 – Начальная страница *IIS Manager*

В левой части окна *Диспетчера служб IIS* отображается запись с именем используемого сервера и доступными для него сайтами (приложениями). На рисунке 4.2 сервер имеет имя **1-37-SHNEIDEROV**. В центральной области в виде набора значков отображаются настройки сервера. В правой части экрана расположен список доступных действий для выбранного объекта.

Архитектура *Internet Information Services*

IIS последних версий (позднее версии 6) отделяет код веб-сервера (режим ядра) от кода поддержки приложений (режим пользователя) при помощи **слушателя режима ядра** **http.sys** и **службы веб-администрирования** (***web administration service***, *WAS*), которая является диспетчером процесса и конфигурации пользовательского режима (рисунок 4.3). Эти программы не выполняют никакого стороннего кода, поэтому ошибка веб-сайта не оказывает на них никакого воздействия. Код выполняется в **рабочем процессе**. Рабочие процессы выполняются с помощью приложения **w3wp.exe**. Каждая копия **w3wp.exe** представляет собой отдельный рабочий процесс. Рабочие процессы существуют отдельно друг от друга и от ядра, поэтому их можно изолировать и от операционной системы.

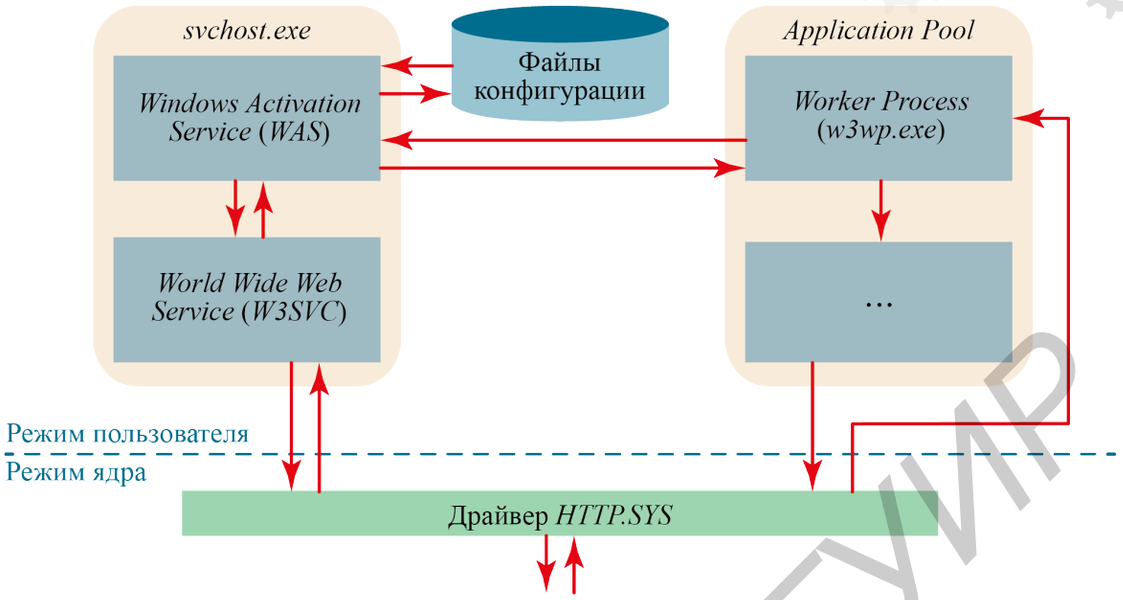
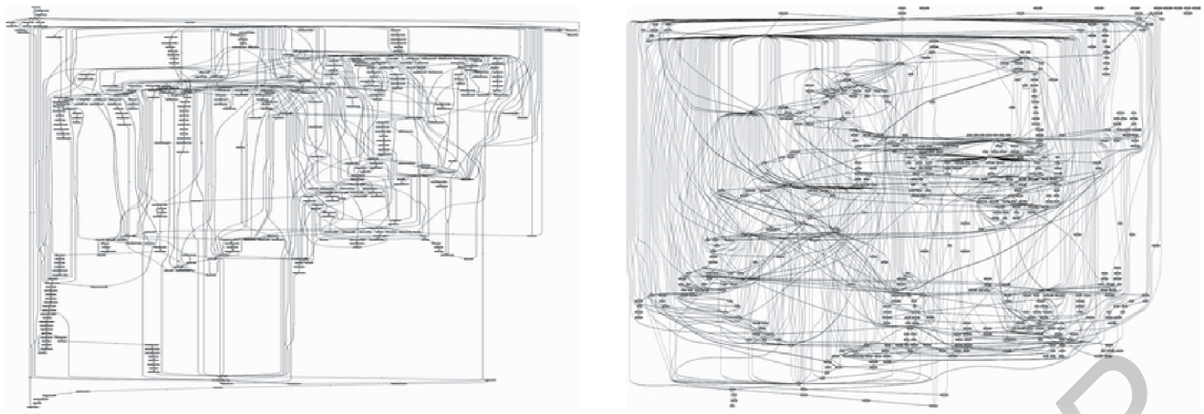


Рисунок 4.3 – Упрощённая архитектура сервера *IIS*

IIS вводит особенный способ поддержки приложений – **пулы приложений**, позволяющие выполнять код в изолированной среде. Каждый пул приложений обслуживается одним или несколькими рабочими процессами. При запуске *IIS* служба веб-администрирования инициализирует таблицу маршрутизации пространства имён **http.sys** с одной записью для каждого приложения. Эта таблица определяет, к какому пулу приложений должно быть маршрутизировано приложение. При получении запроса **http.sys** даёт *WAS* команду на запуск одного или нескольких рабочих процессов для поддержки этого пула приложения. Такая изоляция процессов в целом повышает стабильность веб-сервера.

WAS имеет возможность контролировать общее состояние *IIS* посредством отслеживания рабочих процессов и мониторинга степени их исправности. Это позволяет предотвратить отказы. *WAS* отслеживает состояние рабочих процессов посредством отправки запросов через установленные промежутки времени. Если рабочий процесс не отвечает на запрос, *WAS* завершает процесс и запускает новый, что сохраняет возможность системы отвечать на запросы даже в случае «зависания» рабочего процесса. При возникновении сбоя в рабочем процессе и его «зависании» **http.sys** будет выдавать последовательные запросы до тех пор, пока *WAS* не запустит новый рабочий процесс для поддержки этого пула приложений. Конечный пользователь в таком случае столкнётся с временной потерей обслуживания в данном пуле приложений, в то время как ключевые веб-службы и другие приложения продолжат свою работу.

Microsoft Internet Information Services часто подвергается критике со стороны системных аналитиков. Объектом критики выступает безопасность, сложность в настройке и бóльшее время отклика сайта. В [2] автор графически представил системные вызовы *Apache* (ОС *Linux*) и *IIS* (ОС *Microsoft*) в случае ответа веб-сервером на запрос (рисунок 4.4). Ответ представляет собой *HTML*-страницу с картинкой.



а *б*

а – Apache (ОС *Linux*); *б – IIS* (ОС *Windows*)

Рисунок 4.4 − Графическое отображение системных вызовов при запросе *HTML*-страницы

Каждый лишний вызов представляет собой дополнительную точку, в которой может возникнуть ошибка. Неправильная передача параметра, недостаточный контроль за диапазоном значений, переполнение стека и т. п. – всё это потенциальные проблемы, которые должны быть подвергнуты тестированию и анализу. И это те проблемы, которые потенциально могут быть использованы при взломе.

К сожалению (или к счастью), детальный разбор архитектуры и принципов функционирования *Microsoft IIS* не представляется возможным в рамках лабораторной работы.

Создание *FTP*-сайта

В рамках лабораторной работы предполагается создание *FTP*-сайта. Ниже указаны его предполагаемые основные параметры:

1) имя сервера – student_ftp;
2) домашний каталог сервера – *C:\inetpub\ftp*;
3) сервер должен быть доступен любому сетевому интерфейсу учебного ПК и стандартному порту 21;
4) использование *SSL* должно быть отключено;
5) анонимные пользователи должны иметь полный доступ к каталогу сервера.

До начала добавления сайта необходимо выполнить следующие подготовительные действия: создать каталог по адресу **C:\inetpub\ftp** и присвоить разрешения, позволяющие полный анонимный доступ к созданной папке.

Создание нового *FTP*-сайта будем производить с помощью *Мастера создания FTP-сайтов*. Для запуска мастера необходимо:

1) открыть *Диспетчер служб IIS*;
2) на панели *Подключения* в дереве развернуть узел *Сайты*;
3) щёлкнуть правой кнопкой мыши по узлу *Сайты* и выбрать *Добавить FTP-сайт*. Альтернативным способом является нажатие одноимённой ссылки на панели *Действия*. В появившемся диалоговом окне необходимо ввести имя сайта и указать путь к заранее созданному каталогу (рисунок 4.5).

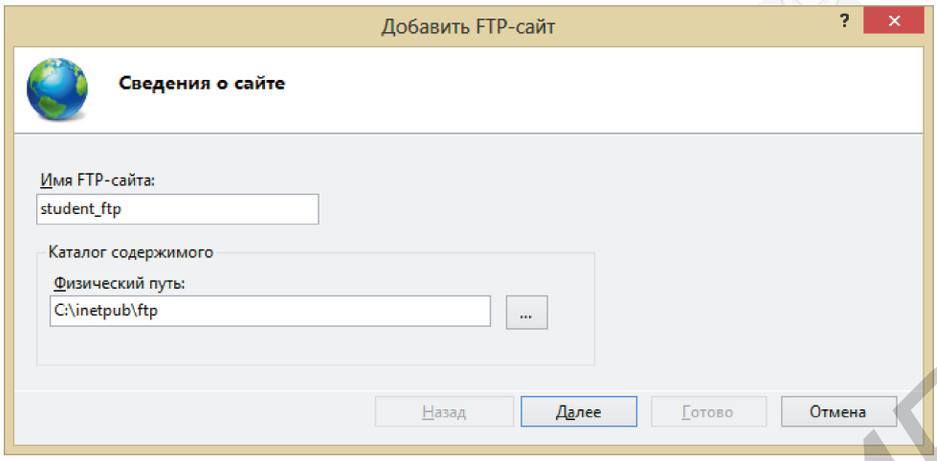


Рисунок 4.5 – Окно добавления *FTP*-сайта

В следующем окне мастера необходимо задать сетевые настройки будущего *FTP*-сайта, выполнив следующие действия:

1) в списке *IP-адрес* необходимо выбрать *IP*-адрес для вашего *FTP*-сайта или оставить значение по умолчанию *Все свободные*. Так как в дальнейшем придётся использовать учётную запись администратора, рекомендуется убедиться, что доступ к серверу ограничен и введён локальный адрес *loopback* (127.0.0.1). При использовании *IPv6* также рекомендуется указать адрес *loopback IPv6* (::1);

2) необходимо ввести порт *FTP*-сайта в поле *Порт*. В данной лабораторной работе используется стандартный порт 21;

3) в лабораторной работе *Виртуальное имя узла* не используется, поэтому необходимо убедиться, что соответствующее поле не заполнено;

4) убедиться, что использование *SSL* отключено.

Следующий шаг мастера предполагает настройку параметров безопасности. С целью упрощения настройки рекомендуется привести к виду, показанному на рисунке 4.6.

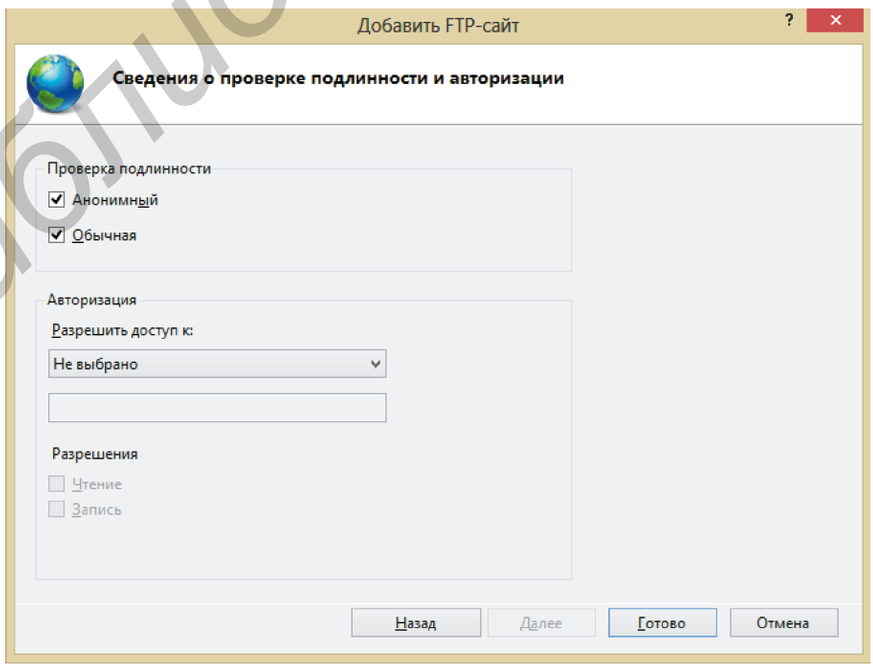


Рисунок 4.6 – Завершение настройки добавляемого *FTP*-сайта

После нажатия кнопки *Готово* *FTP*-сайт будет создан и отобразится в дереве на панели *Подключения*. К нему можно будет получить доступ, набрав в адресной строке браузера *ftp://localhost* (*ftp://127.0.0.1*).

Создание веб-сайта

Для добавления нового веб-сайта с помощью *Мастера добавления* необходимо выполнить следующие действия:

1) на панели *Подключения* в дереве развернуть узел *Сайты*;

2) щёлкнуть правой кнопкой мыши по узлу *Сайты* и выбрать *Добавить веб-сайт*. Альтернативным способом является нажатие одноимённой ссылки на панели *Действия*;

3) в появившемся диалоговом окне (рисунок 4.7) необходимо указать выбранное имя сайта и путь к домашнему каталогу. В рамках данной лабораторной работы используется каталог *FTP*-сайта. В группе параметров *Привязка* можно выбрать тип, *IP*-адрес и порт. Первые два значения можно оставить по умолчанию, а в качестве порта выбрать любой из свободных в рамках операционной системы. Если опция *Запустить веб-сайт немедленно* установлена, то веб-сайт станет доступен сразу после завершения работы мастера.

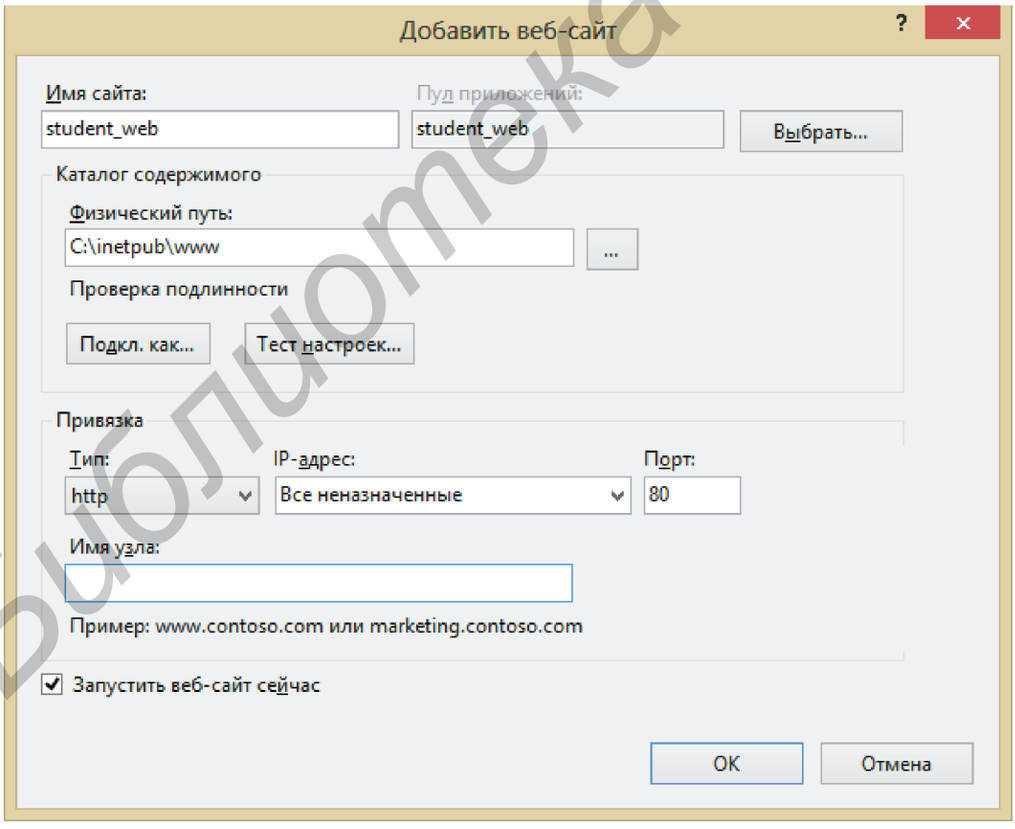


Рисунок 4.7 – Параметры создания веб-сайта

После нажатия кнопки *Готово* веб-сайт будет создан и доступен по адресу *http://localhost:80* (*http://127.0.0.1:80*).

Загрузка файлов веб-сайта на *FTP*-сайт

После создания *FTP*-сайта и веб-сайта можно загрузить тестовую страницу на сервер. Для этого необходимо открыть проводник *Windows* и в строке адреса указать *Uniform Resource Locator* (*URL*, состоящий из протокола, адреса и порта) и нажать *Enter*.

После этого в проводнике отобразится доступное содержимое (указанный ранее каталог) *FTP*-сайта. Копирование может быть осуществлено с помощью стандартных средств *Windows*, например пунктами меню *Правка* → *Копировать* и *Правка* → *Вставить*.

Как только файлы веб-страниц будут загружены на сервер, их можно будет просмотреть с помощью браузера.

4.3 Практическая часть лабораторной работы

Для выполнения практической части лабораторной работы необходимо:

1 По аналогии с лабораторной работой №3, используя выданный преподавателем готовый сетевой кабель (а также в отдельных случаях – коммутатор), объединить рабочие места в сеть[1] и настроить их взаимное сетевое взаимодействие в рамках сформированных студентами бригад.

2 Установить *Microsoft Internet Information Services* на каждой клиентской машине с операционной системой *Microsoft Windows*.

3 Каждому студенту создать и настроить *FTP*-сайт со следующими параметрами: имя сервера – *фамилия_студента*, домашний каталог сервера – *C:\inetpub\фамилия_студента.* Сервер должен быть доступен для анонимных пользователей сети по стандартному порту 21 без использования *SSL*.

4 Каждому студенту создать и настроить веб-сайт со следующими параметрами: имя сервера – по усмотрению студента, путь к данным сайта должен совпадать с открытым каталогом *FTP*-сайта.

5 Разбившись в пределах бригады (сети) на пары, каждому студенту создать простейшую *HTML*-страницу, содержащую его имя и фамилию, загрузить её на *FTP*-сайт товарища и проверить результат работы веб-сервера, отобразив страницу в окне браузера.

6 Написать отчёт по лабораторной работе.

4.4 Содержание отчёта

Отчёт о выполненной работе должен содержать:

1 Титульный лист. Цель лабораторной работы.

2 Схему реализованной сети с адресами узлов и указанием параметров *FTP*-сайта и веб-сайта.

3 Код загружаемой на веб-сайт *HTML*-страницы.

4 Вывод по работе, в котором необходимо указать наиболее сложный этап лабораторной работы.

[1] Преподаватель может заменить физические рабочие места на виртуальные машины с операционной системой семейства *Windows*. В этом случае объединение виртуальных машин будет осуществляться в гипервизоре.

4.5 Контрольные вопросы

1 Какие дополнительные компоненты, кроме служб *Microsoft Internet Information Services*, поставляются с операционными системами семейства *Windows* и могут быть установлены в процессе работы?

2 Укажите назначение служб *Microsoft Internet Information Services.*

3 Охарактеризуйте протоколы передачи данных, поддержку которых реализуют службы *Microsoft Internet Information Services.*

4 Укажите и поясните основные параметры *FTP*- и *HTTP*-сайтов в общем (не привязываясь к службам *Microsoft Internet Information Services*).

5 Поясните, какое назначение имеют представленные в *Диспетчере служб IIS* значки (см. рисунок 4.2).

4.6 Литература

1 Professional IIS 7.0 / K. Schaefer [et al.]. – Wiley Publishing, Inc., Indianapolis, Indiana, 2008. – 812 p.

2 Apache vs IIS [Электронный ресурс]. – 2007. – Режим доступа : http://habrahabr.ru/post/3033/.

3 Internet Information Services (IIS) 7.0 Resource Kit / M. Volodarsky [et al.]. – Microsoft Press; PAP/CDR edition, 2008. – 779 p.

4 Основы архитектуры IIS, или запросопровод для ASP.NET [Электронный ресурс]. – 2013. – Режим доступа : http://habrahabr.ru/post/189086/.

ЛАБОРАТОРНАЯ РАБОТА №5
ИЗУЧЕНИЕ СРЕДЫ МОДЕЛИРОВАНИЯ КОМПЬЮТЕРНЫХ СЕТЕЙ «CISCO PACKET TRACER»

5.1 Цель работы

Изучить типы сетевого и вычислительного оборудования, используемого в компьютерных системах и сетях, а также ознакомиться с программным обеспечением *Cisco Packet Tracer*.

5.2 Краткие теоретические сведения

Cisco Packet Tracer – программное обеспечение, разработанное компанией *Cisco Systems, Inc.* (далее *Cisco*), позволяющее моделировать работу оборудования *Cisco* и различные топологии вычислительных сетей, использующие это оборудование, а также поддерживающее множество сетевых протоколов.

Packet Tracer позволяет моделировать работу в сети большого перечня оборудования: маршрутизаторов, коммутаторов, точек беспроводного доступа, оконечных устройств и т. д. Программа позволяет проводить аппаратную комплектацию оборудования (добавлять платы расширения) и осуществлять настройку сетевого оборудования как с помощью графического интерфейса, так и с помощью командной строки. Графический интерфейс представлен на рисунке 5.1.

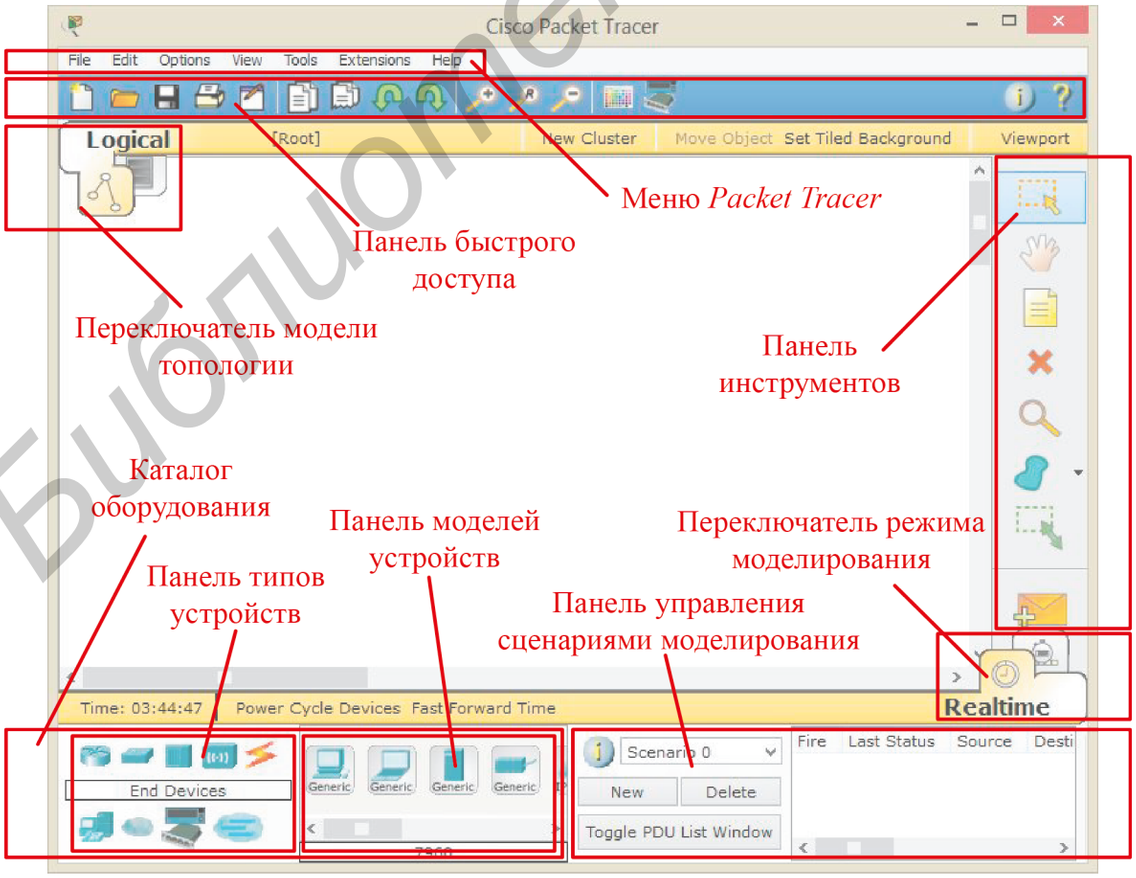


Рисунок 5.1 – Графический интерфейс *Cisco Packet Tracer*

На графическом интерфейсе *Packet Tracer* выделяют следующие области:

1 **Меню**, содержащее основные команды, такие как открытие и сохранение проектов, операции правки и работы с буфером обмена, масштабирование рабочего пространства, отображение панелей инструментов, операции режимов использования программы и др.

2 **Панель быстрого доступа**.

3 **Переключатель модели топологии**, позволяющий работать с логической и физической моделями компьютерных сетей.

4 **Панель инструментов**, содержащая инструменты выделения, удаления, перемещения, масштабирования объектов, а также формирования пакетов.

5 **Переключатель режима моделирования**, позволяющий осуществлять моделирование в режиме реального времени и моделирование в пошаговом режиме, в котором визуально отображается процесс передачи информации по линиям связи.

6 **Каталог оборудования**, содержащий устройства и соединения, которые можно использовать в моделировании (сетевое оборудование в каталоге предусматривает только оборудование *Cisco*).

7 **Панель типов устройств**.

8 **Панель моделей устройств**, выбранных для указанного типа.

9 **Панель создания пользовательских сценариев моделирования**.

Типы оборудования, доступные в *Cisco Packet Tracer*

Доступные в *Packet Tracer* типы оборудования приведены на рисунке 5.2.

Рисунок 5.2 – Обозначение типов оборудования, доступных в *Packet Tracer*

Маршрутизатор (*router*) – сетевое устройство, основной задачей которого является выбор оптимального маршрута передачи данных. Маршрутизатор работает на сетевом (третьем) уровне модели *OSI*.

Коммутатор (***switch***) – сетевое устройство, основной задачей которого является объединение узлов в пределах широковещательного домена. Коммутатор работает на канальном (втором) уровне модели *OSI*.

Коммутатор *L3* (***switch L3***) – сетевое устройство, являющееся усложнённым вариантом обычного коммутатора, позволяющее выполнять некоторые основные функции маршрутизатора. Коммутатор *L3* работает на сетевом (третьем) и канальном (втором) уровнях модели *OSI*.

Мост (***bridge***) – сетевое устройство, основной задачей которого является объединение физических сегментов сети в единую сеть. Мост работает на канальном (втором) уровне модели *OSI*.

Концентратор (***hub***) – сетевое устройство, основной задачей которого, как и коммутатора, является объединение узлов в сеть, однако в отличие от него не является адресным устройством. Концентратор работает на физическом (первом) уровне модели *OSI*.

Повторитель (***repeater***) – сетевое устройство, предназначенное для увеличения расстояния сетевого соединения путём повторения электрического сигнала. Повторитель работает на физическом (первом) уровне модели *OSI*.

Беспроводная точка доступа (***wireless access point***) – сетевое устройство, основной задачей которого является объединение различных сред передачи данных: проводных и беспроводных.

Беспроводной маршрутизатор (***wireless router***) – сетевое устройство, выполняющее те же задачи, что и беспроводная точка доступа, но позволяющее также объединять сегменты в одну сеть и выполнять поиск оптимального маршрута.

Персональный компьютер (***personal computer***), **ноутбук** (***laptop***), **принтер** (***printer***) – вычислительные устройства, оборудованные сетевым адаптером и являющиеся с точки зрения построения сети оконечным оборудованием.

Создание топологии «точка-точка» (*Point-to-Point Link*)

Топология «точка-точка» применяется, как правило, между устройствами одного вида: сетевое-сетевое или вычислительное-вычислительное. Приведём пример такой топологии, применительно к двум компьютерам.

Для объединения двух компьютеров в сеть необходимо добавить на рабочую область *Packet Tracer* два элемента *PC-PT* из каталога оборудования, выбрав в типах устройств *End Devices*. Далее, выбрав в каталоге оборудования *Connections* → *Automatically* ..., соединить эти элементы через интерфейсы *FastEthernet0*. Изображение результата действий приведено на рисунке 5.3.

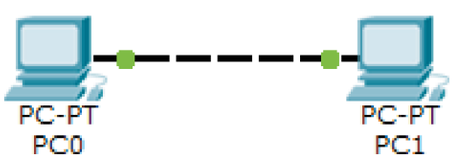


Рисунок 5.3 – Топология «точка-точка» для двух вычислительных устройств

Чтобы обеспечить обмен данными между компьютерами по получившейся сети, необходимо настроить их интерфейсы. Для этого необходимо открыть окно управления устройством (щёлкнуть по устройству левой кнопкой мыши) на вкладке *Desktop* и выбрать *IP Configuration*.

В открывшемся окне *IP Configuration* (рисунок 5.4) необходимо указать такие *IP*-адрес и маску подсети, чтобы оба компьютера входили в один сегмент.

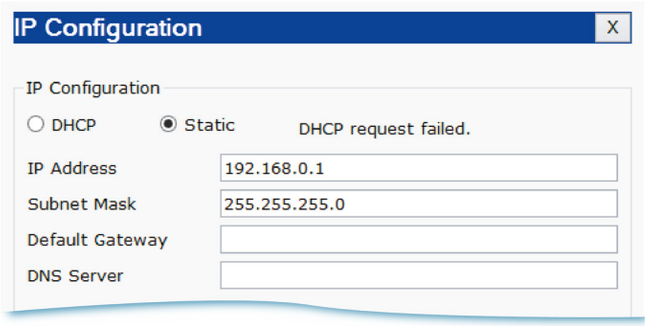


Рисунок 5.4 – Окно *IP Configuration* для компьютера *PC0*

Для быстрого просмотра настройки интерфейсов устройства необходимо навести указатель мыши над ним до появления всплывающей подсказки (рисунок 5.5).

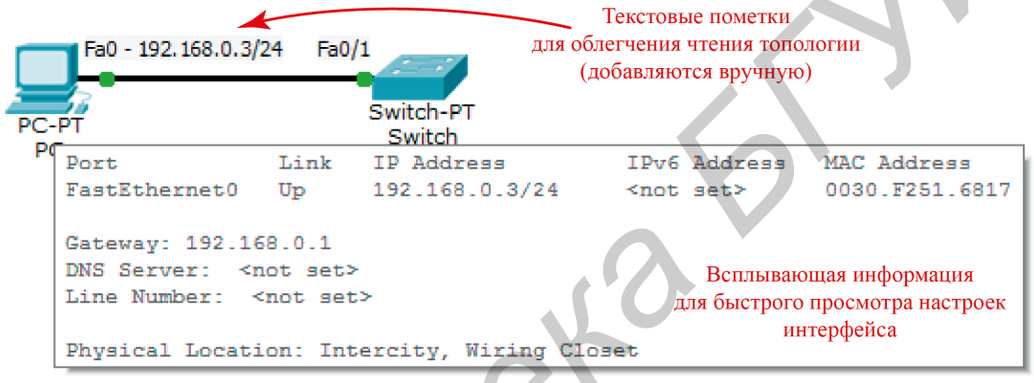


Рисунок 5.5 – Быстрый просмотр настроек устройства

При реализации сетевой топологии в *Packet Tracer* для удобства рекомендуется в места соединений наносить текстовые пометки (инструмент *Place Note*), как показано на рисунке 5.5. Обычно в них указывают название физического интерфейса, сетевой адрес, маску и другую вспомогательную информацию.

На этом настройка устройств закончена. Теперь необходимо проверить работоспособность построенной сети.

Вспомогательные сетевые утилиты ОС *Windows*

Для настройки и диагностики сети в операционной системе *Windows* предусмотрены консольные утилиты **hostname**, **ipconfig**, **ping**, **tracert**, **route**, **net view**, **arp** и **netstat**. Утилиты запускаются из командной строки. Командная строка в *Windows* открывается из меню *Пуск* (*Пуск* → *Все программы* → *Стандартные* → *Командная строка*) или через вызов утилиты *Выполнить* (сочетание клавиш *Win+R* и ввод **cmd** в открывшееся окно). В *Packet Tracer* функции командной строки предоставляет *Command Prompt* (окно работы с устройством, вкладка *Desktop*).

Утилита **hostname** предназначена для отображения имени текущего хоста. Пример использования:

```
C:\example>hostname
example-pc
```

Утилита **ipconfig** предназначена для управления настройками сетевых интерфейсов на текущем хосте. Синтаксис команды[1]:

```
ipconfig [/all] [/renew] [/release],
```

где /all – вывод детальной информации о конфигурации сетевых подключений;
/renew – обновление сведений для всех сетевых адаптеров;
/release – освобождение *IP*-адреса для всех адаптеров (при использовании сервиса *DHCP)*.

Пример использования:

```
C:\example>ipconfig
Ethernet adapter Ethernet:
    DNS-суффикс подключения. . . . :
    IPv4-адрес . . . . . . . . . . . . . . . . . . : 192.168.1.2
    Маска подсети  . . . . . . . . . . . . . . . . : 255.255.255.0
    Основной шлюз. . . . . . . . . . . . . . . . . : 192.168.1.1
```

Утилита **ping** используется для тестирования соединения методом отправки *ICMP*-пакетов к указанному хосту и ожидания ответа. При получении ответа на экране отображается время ожидания и значение *TTL* (*time to live* – времени жизни пакета). Синтаксис команды:

```
ping [-t] [-a] [-n число] [-l размер] [-i TTL] хост,
```

где -t – непрерывная отправка пакетов на указанный хост до ручной остановки процесса нажатием сочетания клавиш *Ctrl+C*, для получения информации о выполненных запросах пользуются сочетанием клавиш *Ctrl+Break*;
-a – отображение совместно с адресом имени хоста;
-n число – число отправляемых запросов;
-l размер – размер объёма данных для отправки в байтах (по умолчанию 32);
-i TTL – время жизни отправляемых пакетов (количество передач между маршрутизаторами перед уничтожением пакета).

Пример использования:

```
C:\example>ping -n 3 -l 1024 -i 20 onliner.by

Обмен пакетами с onliner.by [178.124.129.14] с 1024 байтами данных:
Ответ от 178.124.129.14: число байт=1024 время=8мс TTL=5
Ответ от 178.124.129.14: число байт=1024 время=3мс TTL=5
Ответ от 178.124.129.14: число байт=1024 время=4мс TTL=5

Статистика Ping для 178.124.129.14:
    Пакетов: отправлено = 3, получено = 3, потеряно = 0
    (0% потерь)
Приблизительное время приёма-передачи в мс:
    Минимальное = 3мсек, Максимальное = 8мсек, Среднее = 5мсек
```

[1] Некоторые аргументы в синтаксисе команд в дальнейшем могут быть опущены.

Утилита **tracert** используется для отображения всех промежуточных хостов, расположенных на маршруте к запрашиваемому хосту. Синтаксис команды:

```
tracert [-d] [-h максЧисло] хост,
```

где -d – запрет на отображение имён хостов (отображаются только адреса);
-h максЧисло – максимальное число передач пакета при поиске хоста.
Пример использования:

```
C:\example>tracert –d -h 10 tut.by

Трассировка маршрута к tut.by [178.124.133.65]
с максимальным числом прыжков 10:
 1    1 ms    1 ms    1 ms  172.23.170.0
 2    2 ms    2 ms    3 ms  195.50.30.1
 3    *       *       *     Превышен интервал ожидания для запроса.
 4    *       5 ms    4 ms  195.50.30.254
 5    *       *       *     Превышен интервал ожидания для запроса.
 6    6 ms    4 ms    5 ms  195.137.180.125
 7    3 ms    2 ms    3 ms  178.124.133.65
```

Команда **route** используется для управления таблицей маршрутизации на текущем хосте. При использовании с аргументом **print** показывает содержимое таблицы. Пример использования:

```
C:\example>>route print

===========================
Список интерфейсов
 40 . . . . . . . . . . . . . . . . . . . НИКС VPN
 28 . . 00 ff 78 4d dc 7f . . . . . TeamViewer VPN Adapter
 12 . . 08 60 6e 73 cc 70 . . . . . Контроллер семейства Realtek PCIe GBE
  1 . . . . . . . . . . . . . . . . . . . Software Loopback Interface 1
  . . .
 18 . . 00 00 00 00 00 00 00 e0 Адаптер Microsoft ISATAP #3
===========================

IPv4 таблица маршрута
===========================
Активные маршруты:
Сетевой адрес          Маска сети      Адрес шлюза     Интерфейс       Метрика
          0.0.0.0          0.0.0.0     192.168.1.1     192.168.1.2        4245
          0.0.0.0          0.0.0.0         On-link   172.24.224.10          21
         10.0.0.0        255.0.0.0     192.168.1.1     192.168.1.2       14244
        127.0.0.0        255.0.0.0         On-link       127.0.0.1        4531
      192.168.0.0      255.255.0.0     192.168.1.1     192.168.1.2       14244
      192.168.1.0    255.255.255.0         On-link     192.168.1.2        4501
    192.168.1.255  255.255.255.255         On-link     192.168.1.2        4501
              . . .
     217.21.42.0  255.255.255.224     192.168.1.1     192.168.1.2       14244
   217.21.42.192  255.255.255.224     192.168.1.1     192.168.1.2       14244
255.255.255.255  255.255.255.255         On-link       127.0.0.1        4531
===========================
```

```
Постоянные маршруты:
  Сетевой адрес          Маска        Адрес шлюза  Метрика
        10.0.0.0        255.0.0.0    192.168.1.1     9999
      192.168.0.0      255.255.0.0    192.168.1.1     9999
                . . .
     217.21.42.0  255.255.255.224    192.168.1.1     9999
   217.21.42.192  255.255.255.224    192.168.1.1     9999
=======================
```

Утилита **net view** отображает список хостов рабочей группы или общих ресурсов на запрашиваемом хосте. Синтаксис команды:

```
net view [\\хост | /workgroup[:хост],
```

где \\хост – задаёт имя хоста для просмотра общих ресурсов;

/workgroup[:хост] – задаёт рабочую группу, для которого выводится список хостов, или хост для просмотра общих ресурсов в указанной рабочей группе.

Пример использования:

```
C:\example>net view \\example-pc

Общие ресурсы на \\example-pc
Имя общего ресурса     Тип      Используется как    Комментарий
-------------------------------------------------------------------------------
Documents              Диск
Music                  Диск
         . . .
temp disk              Диск
```

Утилита **netstat** выводит статистику использования протоколов или список текущих подключений сети *TCP/IP* в зависимости от значений аргументов. Синтаксис команды:

```
netstat [-a] [-e] [-n] [-p протокол] [-r] [интервал],
```

где -a – отображает все подключения и сетевые порты;

-e – отображает статистику *Ethernet*;

-n – отображает адреса вместо имён хостов;

-p протокол – отображает соединения для заданного протокола;

-r – отображает таблицу маршрутизации (аналог **route print**);

интервал – обновляет отображённую статистику с заданным в секундах интервалом. Для остановки мониторинга используется сочетание клавиш *Ctrl+B*.

Пример использования:

```
C:\example>netstat

Активные подключения
  Имя          Локальный адрес        Внешний адрес          Состояние
  TCP          127.0.0.1:39000        lmlicenses:53144       ESTABLISHED
  TCP          127.0.0.1:53144        lmlicenses:39000       ESTABLISHED
  TCP          127.0.0.1:54198        lmlicenses:54204       ESTABLISHED
```

Утилита **arp** используется для управления *ARP*-таблицей (соотношений *IP*-адресов и *MAC*-адресов для текущего хоста). При использовании с ключом -a позволяет увидеть текущее состояние таблицы. Пример использования:

```
C:\example>arp -a

Интерфейс: 172.16.240.216 --- 0xc
  адрес в Интернете     Физический адрес      Тип
  172.16.0.1            00-15-17-be-d0-c6     динамический
  172.16.0.8            00-15-17-be-d0-c6     динамический
                  . . .
  172.16.20.2           00-10-4b-97-89-ad     динамический
```

Проверка работоспособности сети в режиме отслеживания пакетов

Для детальной проверки работоспособности созданной ранее сети можно воспользоваться её моделированием в пошаговом режиме. Для этого необходимо включить этот режим в *Packet Tracer* (см. рисунок 5.1). Алгоритм выполняемых действий следующий (рисунок 5.6):

1) выбрать режим *Simulation Mode*;

2) выбрать инструмент *Add Simple PDU* (*Protocol Data Unit*). Сначала щёлкнуть по ПК-источнику *ICMP*-пакета, потом – по ПК-получателю;

3) настроить фильтр пакетов (*Edit Filters*) таким образом, чтобы отображались только *ICMP*-пакеты;

4) для начала автоматической передачи *ICMP*-пакета нажать *Auto Capture → Play*, для пошаговой ручной передачи – *Capture → Forward*. В таблице *Event List* отображаются события, происходящие при передаче пакетов.

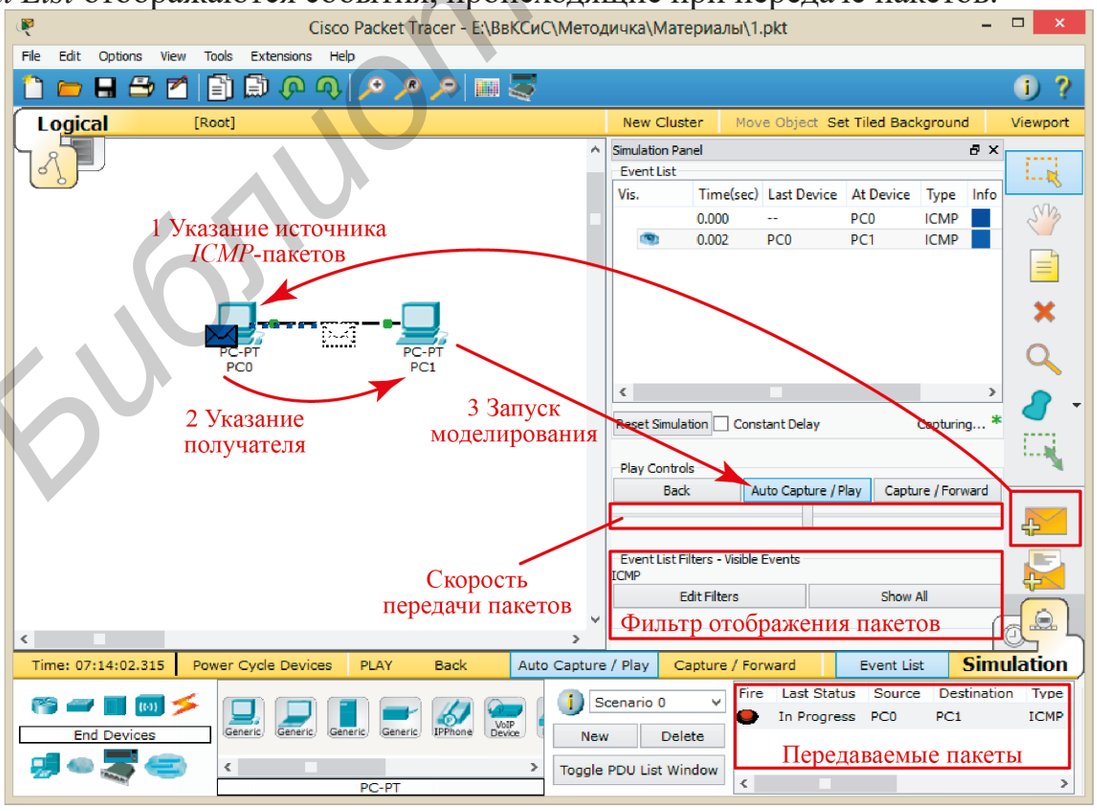


Рисунок 5.6 – Моделирование в пошаговом режиме

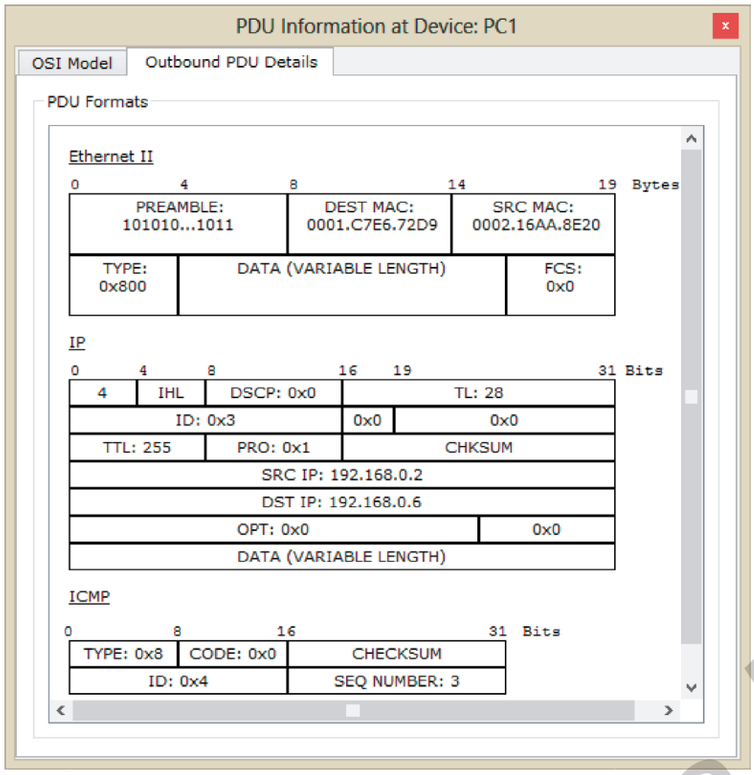


Рисунок 5.7 – Заголовки *PDU*

Отправляемый пакет отображается в виде конверта (см. рисунок 5.6), цвет которого соответствует указанному в области *Панель управления сценариями моделирования*. В случае успешной передачи пакета в графе *Last Status* области передаваемых пакетов отобразится надпись *Successful* (успех), в случае неудачи – *Failure* (неудача). При щелчке левой кнопкой мыши по изображению пакета отобразится окно, содержащее информацию заголовков *PDU* в текущий момент (рисунок 5.7).

Для перезапуска моделирования необходимо воспользоваться *Reset Simulation*.

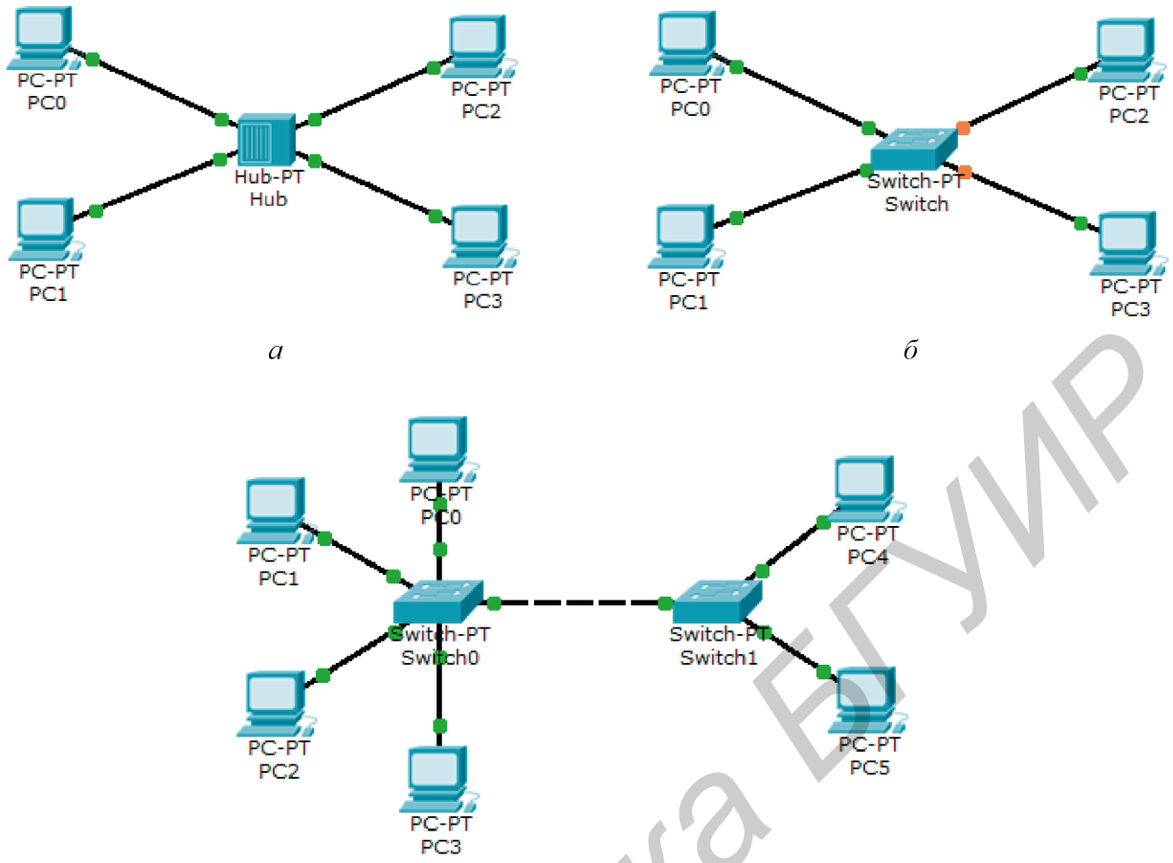
Типовые схемы объединения устройств в сеть

Для объединения вычислительных устройств в сеть используется, как правило, коммутатор или концентратор (рисунок 5.8, *а*, *б*). При этом устройства, подключённые к сетевому оборудованию, образуют **широковещательный домен**. В том случае, если количество единиц вычислительной техники превышает количество физических интерфейсов, сетевое оборудование может объединяться, при этом все вычислительные устройства остаются в одном домене.

Стоит отметить, что настройка статических *IP*-адресов должна быть выполнена в одном сегменте сети, в противном случае устройства будут недоступны для обмена информацией между собой.

При реализации топологии нужно помнить о том, какие линии связи подключаются к тем или иным интерфейсам. *Packet Tracer* поддерживает множество различных линий связи, которые для удобства отличаются своим изображением. В данной лабораторной работе необходимо использовать линии связи *Copper Straight-Trough* (для связи разных типов оборудования) и *Copper Crossover* (для связи одного типа оборудования).

Для объединения вычислительных устройств в один сегмент сети настройка коммутаторов и концентраторов, а также их интерфейсов не требуется. Настройке подлежат только подключаемые вычислительные устройства.



а *б*

в

а – с помощью концентратора; *б* – с помощью коммутатора;
в – с помощью двух сетевых устройств

Рисунок 5.8 – Типовые схемы объединения нескольких ПК в сеть

5.3 Практическая часть лабораторной работы

Для выполнения практической части лабораторной работы необходимо:

1 Изучить работу консольных утилит **hostname**, **ipconfig**, **ping**, **tracert**, **route**, **net view**, **arp** и **netstat** при работе в сетевой среде учебной аудитории. Научиться объяснять результаты их выполнения.

2 Ознакомиться с графическим интерфейсом *Cisco Packet Tracer*.

3 Реализовать в *Packet Tracer* топологию «точка-точка» между двумя компьютерами. Проверить работоспособность сети с помощью консольных утилит. Выяснить, какие консольные утилиты не реализованы или реализованы не в полном объёме в среде моделирования сетей *Packet Tracer*.

4 Проверить работоспособность сети в режиме отслеживания пакетов.

5 Проанализировать заголовки исходящего и входящего *ICMP*-пакетов и инкапсулирующих их *PDU*.

6 Без использования автоматического выбора типа линии связи реализовать сетевые топологии, показанные на рисунке 5.8, *а*, *б*. На практике подтвердить отличие передачи пакетов концентратором и коммутатором (для этого рекомендуется выполнить моделирование сети в режиме отслеживания пакетов).

7 Без использования автоматического выбора типа линии связи реализовать сетевую топологию, основанную на использовании двух сетевых устройств (см. рисунок 5.8, *в*). Для удобства назовём эти сетевые устройства СУ1 и СУ2. Для определения точного состава устройств в моделируемой сети необходимо воспользоваться таблицей 5.1. Для адресации использовать подсеть с адресом **192.168.X.0/24**, где **X** – номер варианта.

Таблица 5.1 – Устройства для реализации топологии сети

Типы сетевых устройств	Устройства, подключаемые к СУ 1	Устройства, подключаемые к СУ2
Для нечётных вариантов: *Switch* + *Switch*	Для вариантов 1...15: 3 *PCs* + 1 *Laptop* + 1 *Printer*	Для вариантов, кратных 3: 2 *PCs* + 1 *Laptop*
Для чётных вариантов: *Switch* + *Hub*	Для вариантов 16...30: 1 *PC* + 2 *Laptops* + 2 *Printers*	Для вариантов, не кратных 3: 1 *PC* + 2 *Laptops*

Проверить работоспособность сети в режиме отслеживания пакетов, пересылая *ICMP*-пакет от ПК, подключённого к СУ1, к ПК, подключённому к СУ2.

8 Проанализировать заголовки *PDU* первых трёх слоёв модели *OSI* на участках сети «ПК – СУ1», «СУ1 – СУ2» и «СУ2 – ПК».

9 Написать отчёт по лабораторной работе.

5.4 Содержание отчёта

Отчёт о выполненной работе должен содержать:

1 Титульный лист. Цель лабораторной работы.

2 Результаты использования консольных утилит **hostname**, **ipconfig**, **ping**, **tracert**, **route**, **net view**, **arp** и **netstat** в реальной сетевой среде учебной аудитории. Если результаты использования громоздки и непригодны для включения в отчёт, позволяется искусственно сократить их объём, но не приводя к потере их смысла. Рекомендуемый объём – 2...3 страницы.

3 Изображение реализованной сетевой топологии «точка-точка».

4 Заполненную таблицу, содержащую следующие столбцы: консольная утилита, назначение, реализация в ОС *Windows*, реализация в *Packet Tracer*.

5 Изображения заголовков входящего и исходящего *PDU* с указанием различий между ними и пояснением этих различий.

6 Аргументированные пояснения в различиях передачи пакетов коммутатором и концентратором.

7 Изображение реализованной сетевой топологии на базе двух сетевых устройств согласно варианту (на изображении с помощью текстовых пометок необходимо указать адреса сетевых интерфейсов).

8 Изображения заголовков *PDU* на участках сети «ПК – СУ1», «СУ1 – СУ2» и «СУ2 – ПК» с указанием различий между ними и пояснением этих различий.

9 Вывод по работе, в котором необходимо указать наиболее сложный этап лабораторной работы.

5.5 Контрольные вопросы

1 В чём заключается функциональное отличие коммутатора от концентратора? В каких сетях применяются эти сетевые устройства? Использование какого из этих устройств обеспечивает лучшую безопасность в сети?

2 Каким образом в *Packet Tracer* реализована генерация большого объёма сетевого трафика от какого-либо оконечного устройства? Какие основные параметры имеет эта функция?

3 Что такое сетевая топология? Какие виды топологий существуют и каково их назначение?

4 Какой логический смысл несут в себе процессы инкапсуляции и декапсуляции блоков информации, передающихся по стеку уровней модели *OSI*?

5 В чём заключается назначение протокола *ICMP* и каков формат его заголовка? Приведите практические примеры его использования в реальной жизни.

5.6 Литература

1 Одом, У. Официальное руководство по подготовке к сертификационным экзаменам CCENT/CCNA ICND1 / У. Одом ; пер. с англ. – 2-е изд. – М. : ООО «И. Д. Вильямс», 2010. – 672 с.

2 Хилл, Б. Полный справочник по Cisco / Б. Хилл ; пер. с англ. – М. : ООО «И. Д. Вильямс», 2004. – 1088 с.

3 Станек, У. Командная строка Microsoft Windows. Справочник администратора / У. Станек ; пер. с англ. – М. : Издательско-торговый дом «Русская Редакция», 2004. – 480 с.

ЛАБОРАТОРНАЯ РАБОТА №6
АНАЛИЗ И ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ СЕТИ С ИСПОЛЬЗОВАНИЕМ АНАЛИЗАТОРА ТРАФИКА WIRESHARK

6.1 Цель работы

Получить навыки практического применения анализатора сетевого трафика *Wireshark* при администрировании компьютерной сети.

6.2 Теоретические сведения

Анализатор трафика (**сниффер**, ***network analyser***, ***sniffer***) – программно-аппаратный комплекс, предназначенный для перехвата и анализа трафика, проходящего через сетевой интерфейс. Область его использования следующая:

– поиск неисправностей в вычислительной сети (*network troubleshooting*);

– обнаружение вторжений в сеть (*network intrusion*);

– мониторинг действий пользователей сети (*network usage monitoring*);

– сбор статистических данных о работе сети (*network statistic*);

– сетевой шпионаж (*network espionage*).

Как правило, аппаратная часть комплекса представляет собой сетевой адаптер ЭВМ, на которой устанавливается соответствующее программное обеспечение. Поэтому производители ограничиваются разработкой только высокоуровневого ПО, которое представляет собой программный декодер и анализатор *PDU*.

Перехват трафика может осуществляться:

– подключением нового узла со сниффером в сеть, что эффективно при использовании в сегменте концентраторов, но малоэффективно при использовании коммутаторов;

– подключением сниффера в разрыв канала;

– программным или аппаратным зеркалированием портов (*port mirroring*) и направлением копии трафика на сниффер;

– через анализ побочных электромагнитных излучений;

– через подмену исходного сетевого узла другим методом программной установки на его интерфейс исходного физического (*MAC-spoofing*) или сетевого адреса (*IP-spoofing*).

Наиболее известными и распространёнными анализаторами трафика являются:

– *tcpdump* (*the Tcpdump team*, платформа *Linux*, *BSD license*);

– *Capsa* (*Colasoft*®, платформа *Windows*, *freeware*/*shareware*);

– *Xplico* (персональная разработка, платформа *Ubuntu*/*Linux*, *GNU GPL*);

– *Wireshark* (*Wireshark team*, платформы *Windows*/*Linux*, *GNU GPLv*2);

– *LanGrabber* (*SkySoftware*®, платформа *Windows*, *shareware*);

– *CommView* (*TamoSaft*®, платформа *Windows*, *shareware*);

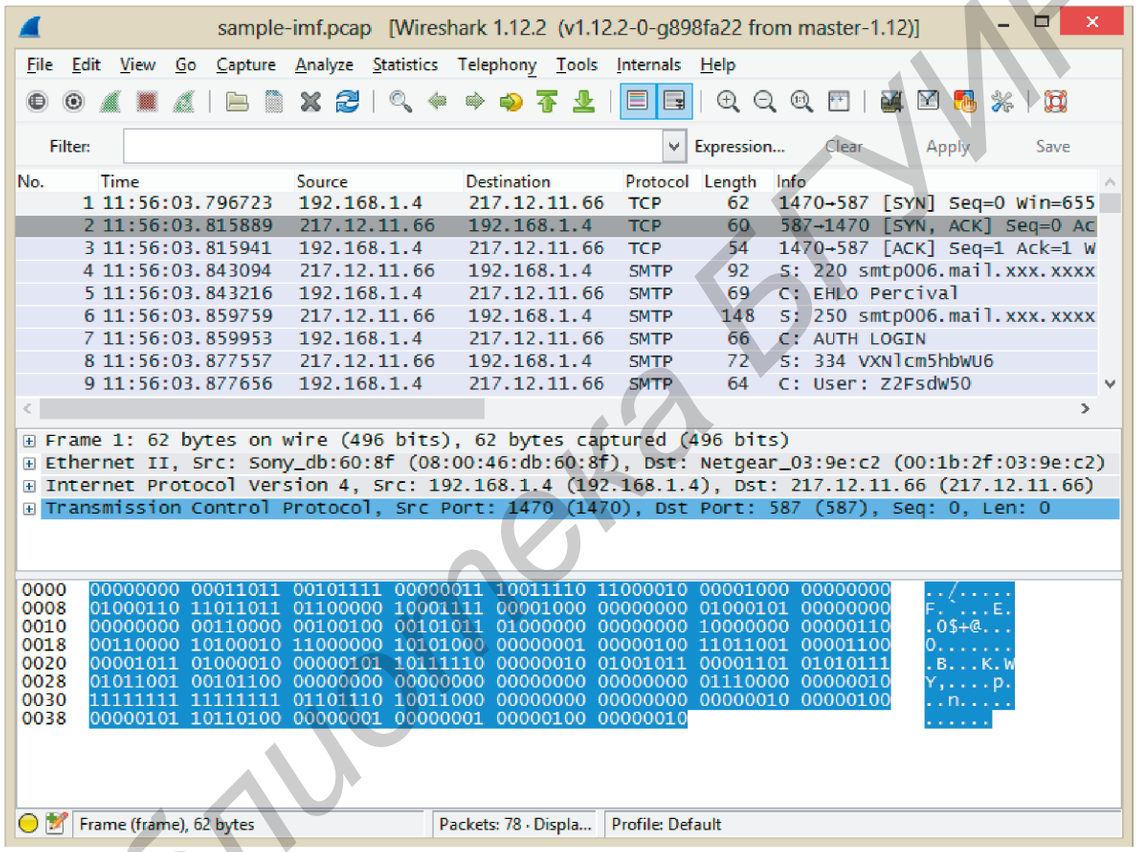
– *Ettercap* (персональная разработка, платформа *Linux*, *freeware*);

– *Microsoft Message Analyzer* (*freeware*).

Wireshark

Wireshark (ранее *Ethereal*) – анализатор трафика для сетей 802.3 *Ethernet*, 802.11 *WLAN* и др. Приложение позволяет просматривать проходящий по сети трафик в режиме реального времени, переводя сетевую карту в режим *promiscuous mode*. *Wireshark* позволяет декодировать *PDU* сетевых протоколов, что позволяет ему отображать значение каждого поля заголовка протокола любого уровня. Также *Wireshark* умеет работать с форматами файлов других снифферов и имеет свободную лицензию, что делает его наиболее привлекательным.

Графический интерфейс *Wireshark* в работе представлен на рисунке 6.1.



1 – меню; 2 – панель инструментов; 3 – панель фильтров; 4 – список принятых пакетов; 5 – панель дерева протоколов, инкапсулированных в кадр; 6 – битовое представление пакета

Рисунок 6.1 – Основное окно приложения *Wireshark*

Традиционно **Меню** (1) приложения *Wireshark* включает в себя все инструменты и команды, которые доступны пользователю во время работы. **Панель инструментов** (2) дублирует те инструменты и команды меню, которые чаще всего могут быть использованы пользователем. **Панель фильтров** (3) представляет собой инструмент, позволяющий гибко управлять отображением интересующих *PDU* в **Списке принятых пакетов** (4). **Панель дерева протоколов, инкапсулированных в кадр**, (5) и **Битовое представление пакета** (6) предназначены для анализа информации пользователем.[1]

[1] Объём лабораторного практикума ограничен и не позволяет привести подробное описание компонентов и возможностей *Wireshark*, поэтому дополнительную информацию по приложению студентам необходимо найти самостоятельно на официальном сайте приложения.

Захват трафика *Wireshark*

Для захвата трафика на сетевом интерфейсе в меню *Wireshark* используется группа команд *Capture*. Важным нюансом является тот факт, что для захвата трафика в ОС *Windows* 7 и выше *Wireshark* должен быть запущен с правами администратора (для установки собственного драйвера сетевого адаптера).

Для реализации процедуры захвата необходимо выполнить следующие действия:

1 **Выбрать интересующие сетевые интерфейсы узла** с помощью команды *Capture → Interfaces...*, *Ctrl+I* или нажатием соответствующей кнопки на панели инструментов. При вызове команды откроется окно (рисунок 6.2) со списком доступных приложению интерфейсов. В этом окне отображается системное имя интерфейса, описание, сетевой адрес, количество входящих/исходящих пакетов и кнопка отображения детальной информации об интерфейсе.

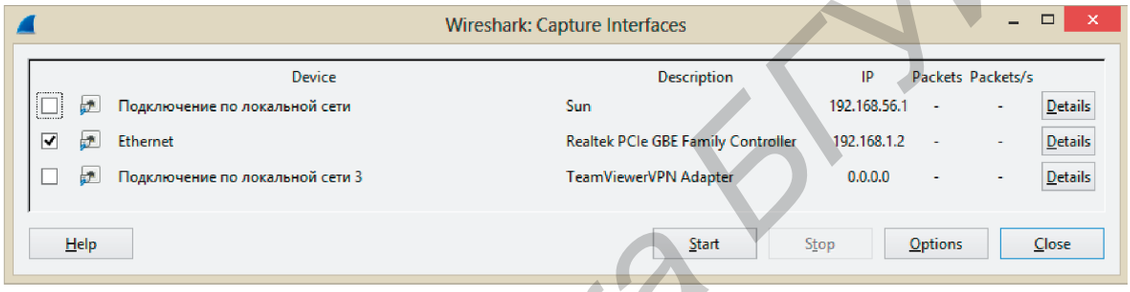


Рисунок 6.2 – Окно выбора интерфейсов для захвата трафика

2 **Определить настройки захвата трафика** с помощью команды *Capture → Options...*, *Ctrl+K* или нажатием соответствующей кнопки на панели инструментов. При вызове команды откроется окно (рисунок 6.3) со списком настроек. Все настройки просты и понятны. При возникновении сложностей можно обратиться ко встроенной системе помощи, нажав кнопку *Help*.

3 **Начать захват трафика** с помощью команды *Capture → Start*, *Ctrl+E* или нажатием соответствующей кнопки на панели инструментов или в окне настроек захвата трафика.

4 **Остановить захват трафика** при необходимости с помощью команды *Capture → Stop*, *Ctrl+E* или нажатием соответствующей кнопки на панели инструментов.

Особенности захвата трафика:

– если не указать файл для сохранения захваченного трафика, то он будет храниться в оперативной памяти, что не является эффективным решением с точки зрения накапливаемого объёма информации;

– если применить фильтр трафика, это может существенно сократить расходы памяти, а также снизить нагрузку на жёсткий диск в процессе его захвата;

– при установленных параметрах *Resolve MAC addresses*, *Resolve network-layer name* и *Resolve transport-layer name* приложение будет по возможности отображать зарегистрированного производителя для *MAC*-адресов, сетевое или доменное имя и название программы, за которой зарегистрирован соответствующий порт.

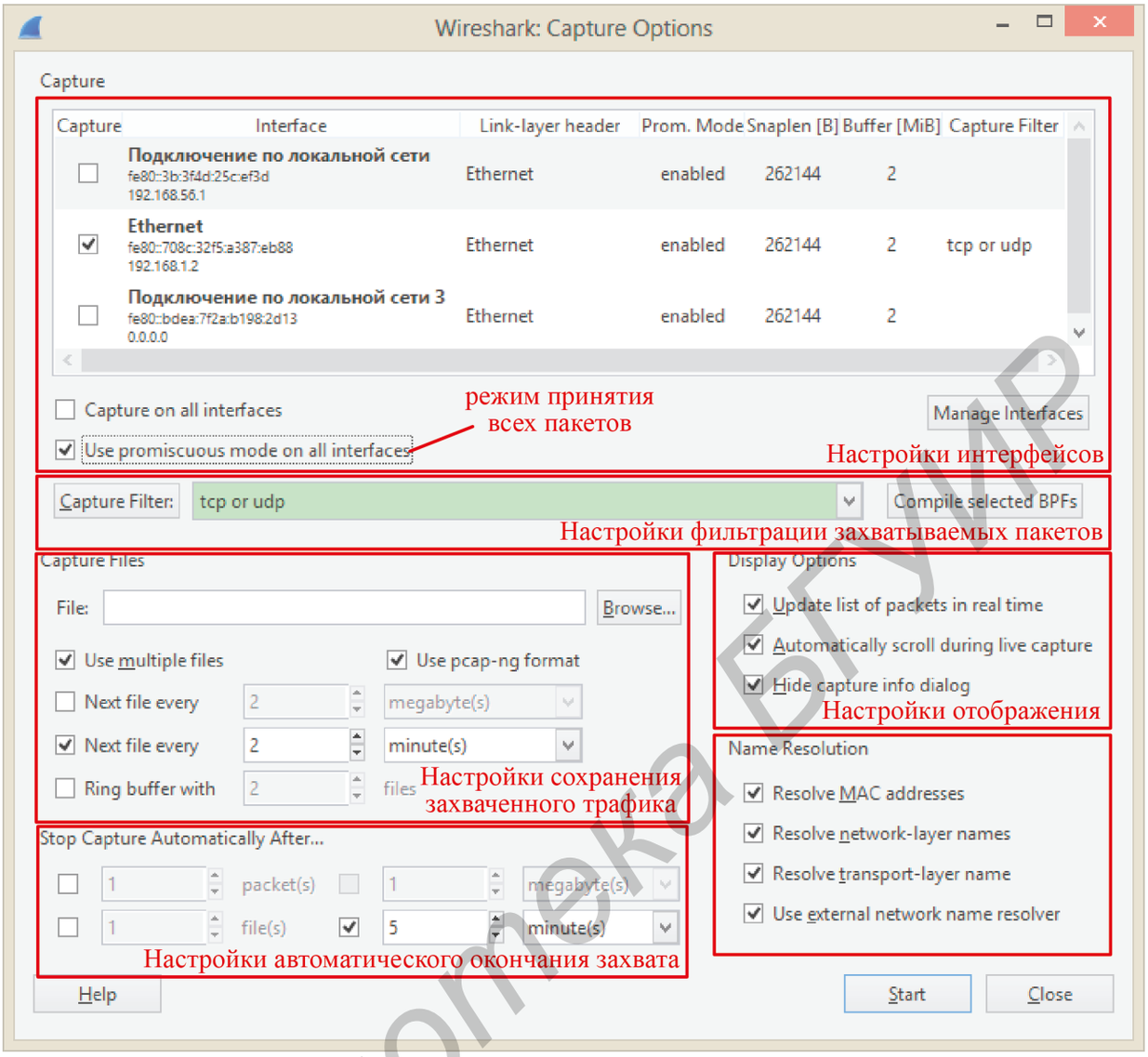


Рисунок 6.3 – Окно настроек захвата трафика

Фильтры *Wireshark*

Фильтры *Wireshark* бывают двух видов – фильтры захвата (*Capture Filters*) и фильтры отображения (*Display Filters*).

Фильтры захвата служат для отбрасывания не интересующих пользователя пакетов на этапе захвата трафика. При их применении часть трафика безвозвратно теряется и восстановить его нет возможности.

Фильтр захвата представляет собой строку, сформированную из правильно построенных выражений (синтаксических конструкций), объединённых логическими операторами, шаблон которой представлен на рисунке 6.4. Каждое такое выражение может состоять из следующих компонентов: оператора НЕ, протокола, направления трафика, типа объекта и его значения. Каждый компонент синтаксической конструкции не является обязательным, а значит, отсутствие фильтра (отсутствие всех компонентов) также воспринимается приложением как своего рода нулевой фильтр.

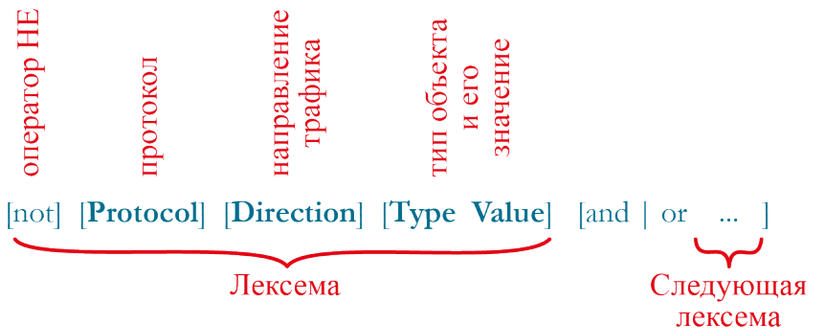


Рисунок 6.4 – Шаблон построения фильтров захвата

Значения компонентов синтаксических конструкций приведены в таблице 6.1.

Таблица 6.1 – Значения компонентов синтаксических конструкций при формировании фильтров захвата

Компонент	Описание	Значения
Protocol	Фильтрация по определённому сетевому протоколу	arp, icmp, ip, rarp, tcp, udp и т. д.
Direction	Направление передачи *PDU*: от узла или к узлу	src, dst
Type	Фильтрация по различным типам адреса: физическому, сетевому, номеру порта и т. п.	ether, host, net, port (portrange)

Примечание – полный список значений компонента **Protocol** можно посмотреть в источнике [2].

Логические операторы, применяющиеся для формирования сложного (составного) фильтра захвата, могут быть записаны в двух формах:

– логический оператор И: **&&** или **and**;

– логический оператор ИЛИ: **||** или **or**;

– логический оператор НЕ: **!** или **not**.

Примеры фильтров захвата:

– захват трафика, связанного с определённым узлом: host 172.18.5.4;

– захват трафика от определённого сегмента: src net 192.168.0.0/24;

– захват только *DNS*-сообщений (порт 53): port 53;

– захват трафика с определённого узла, исключая *HTTP*-сообщения и *SMTP*-сообщения: host www.example.com and not (port 80 or port 25);

– захват *TCP*-сегментов, связанных с определённым диапазоном портов: tcp portrange 1501-1549.

Также при захвате есть возможность осуществлять фильтрацию по длине пакета, протоколу и типу вещания. Правила записи таких лексем можно посмотреть во встроенной системе помощи или на сайте разработчиков приложения в разделе *Документация* [2].

Фильтры отображения служат для того, чтобы скрыть уже принятые пакеты, не интересующие пользователя. При их применении пакеты не теряются, а отобразить скрытую информацию можно отменив фильтрацию.

Фильтр отображения также как и фильтр захвата представляет собой строку, сформированную из правильно построенных выражений (синтаксических конструкций), объединённых логическими операторами. Однако здесь выражения могут представлять сложные логические выражения, например

```
ip.src == 192.168.0.10  and  ip.dst == 192.168.0.0 /16
```

Этот фильтр отобразит весь трафик, отправленный узлом 192.168.0.10 во внутреннюю сеть.

Синтаксические конструкции представляют собой поля (названия протоколов и их параметры; таблица 6.3) и интересующие значения этих полей, связанные операторами сравнения (например ip.src == 192.168.0.10). При этом значение поля может быть и не указано. В этом случае отображается трафик, к которому применимо это поле, но с любым его значением. Все используемые операторы сравнения приведены в таблице 6.2.

Таблица 6.2 – Операции сравнения при построении фильтров отображения

Операция	Текстовая форма	Значение	Пример использования
==	eq	... равно ...	ip.dst == 192.168.3.10
!=	ne	... не равно ...	udp.dst != 53
<	gt	... меньше, чем ...	ip.ttl < 10
>	lt	... больше, чем ...	tcp.stream > 90
<=	ge	... меньше или равно ...	frame.len ge 0x21
>=	le	... больше или равно ...	tcp.window_size >= 64
	matches	регулярные выражения	frame matches «[Pp][Aa][Ss]»
	contains	содержит	dns.resp.name contains google

Все поля строго типизированы (имеют определённый формат значения). Существующие в *Wireshark* форматы значений:

– беззнаковое 8-, 16-, 24- или 32-битовое целое число (*unsigned integer*);
– 8-, 16-, 24- или 32-битовое целое число со знаком (*signed integer*);
– логическое значение (*boolean*);
– 6-байтовый *Ethernet*-адрес (*Ethernet address*);
– последовательность байт произвольной длины (*byte string*);
– 4-байтовый *IPv4*-адрес (*IPv4-address*);
– 16-байтовый *IPv6*-адрес (*IPv6 address*);
– 4-байтовый номер *IPX* сети (*IPX network number*);
– последовательность символов произвольной длины (*string*);
– 8-байтовое число двойной точности с плавающей точкой (*double-precision floating point number*).

Таблица 6.3 – Поля для формирования фильтров отображения

Ethernet		
eth.addr	eth.len	eth.src
eth.dst	eth.lg	eth.trailer
eth.ig	eth.padding	eth.type

IEEE 802.1Q Virtual LAN		
vlan.cfi	vlan.id	vlan.priority
vlan.etype	vlan.len	vlan.trailer

IPv4	
ip.addr	ip.fragments
ip.checksum	ip.hdr_len
ip.checksum_bad	ip.host
ip.checksum_good	ip.id
ip.dsfield	ip.len
ip.dst	ip.proto
ip.dst_host	ip.src
ip.flags	ip.src_host
ip.flags.df	ip.tos
ip.flags.mf	ip.tos.cost
ip.flags.rb	ip.tos.delay
ip.fragment	ip.tos.reliability
ip.fragment.count	ip.tos.throughput
ip.fragment.error	ip.ttl
ip.fragment.multipletails	ip.version

IPv6	
ipv6.addr	ipv6.host
ipv6.class	ipv6.mipv6_length
ipv6.dst	ipv6.mipv6_type
ipv6.dst_host	ipv6.nxt
ipv6.dst_opt	ipv6.opt.pad1
ipv6.flow	ipv6.opt.padn
ipv6.fragment	ipv6.plen
ipv6.fragment.error	ipv6.reassembled_in
ipv6.fragment.more	ipv6.routing_hdr
ipv6.fragment.offset	ipv6.routing_hdr.addr
ipv6.fragment.overlap	ipv6.routing_hdr.left
ipv6.fragments	ipv6.routing_hdr.type
ipv6.fragment.id	ipv6.src
ipv6.hlim	ipv6.src_host
ipv6.hop_opt	ipv6.version

ARP	
arp.dst.hw_mac	arp.proto.size
arp.dst.proto_ipv4	arp.proto.type
arp.hw.size	arp.src.hw_mac
arp.hw.type	arp.src.proto_ipv4
arp.opcode	

TCP	
tcp.ack	tcp.options.qs
tcp.checksum	tcp.options.sack
tcp.checksum_bad	tcp.options.sack_le
tcp.checksum_good	tcp.options.sack_perm
tcp.continuation_to	tcp.options.sack_re
tcp.dstport	tcp.options.time_stamp
tcp.flags	tcp.options.wscale
tcp.flags.ack	tcp.options.wscale_val
tcp.flags.cwr	tcp.pdu.last_frame
tcp.flags.ecn	tcp.pdu.size
tcp.flags.fin	tcp.pdu.time
tcp.flags.push	tcp.port
tcp.flags.reset	tcp.reassembled_in
tcp.flags.syn	tcp.segment
tcp.flags.urg	tcp.segment.error
tcp.hdr_len	tcp.segment.overlap
tcp.len	tcp.segments
tcp.nxtseq	tcp.seq
tcp.options	tcp.srcport
tcp.options.cc	tcp.time_delta
tcp.options.ccecho	tcp.time_relative
tcp.options.ccnew	tcp.urgent_pointer
tcp.options.echo	tcp.window_size
tcp.options.echo_reply	tcp.options.mss_val
tcp.options.md5	
tcp.options.mss	

UDP	
udp.checksum	udp.length
udp.checksum_bad	udp.port
udp.checksum_good	udp.srcport
udp.dstport	

HTTP		FTP
http.accept	http.proxy_authorization	ftp.active.cip
http.accept_encoding	http.proxy_connect_host	ftp.active.nat
http.accept_language	http.proxy_connect_port	ftp.active.port
http.authbasic	http.referer	ftp.eprt.af
http.authcitrix	http.request	ftp.eprt.args_invalid
http.authcitrix.domain	http.request.full_uri	ftp.eprt.ip
http.authcitrix.password	http.request.line	ftp.eprt.ipv6
http.authcitrix.session	http.request.method	ftp.eprt.port
http.authcitrix.user	http.request.uri	ftp.epsv.args_invalid
http.authorization	http.request.version	ftp.epsv.ip
http.cache_control	http.request_in	ftp.epsv.ipv6
http.chat	http.request_number	ftp.epsv.port
http.chunk_boundary	http.response	ftp.passive.ip
http.chunk_size	http.response.code	ftp.passive.nat
http.chunkd_and_length	http.response.line	ftp.passive.port
http.chunked_trailer_part	http.response.phrase	ftp.request
http.connection	http.response_in	ftp.request.arg
http.content_encoding	http.response_number	ftp.request.command
http.content_length	http.sec_websocket_accept	ftp.response
http.content_length_header	http.sec_websocket_extensions	ftp.response.arg
http.content_type	http.sec_websocket_key	ftp.response.code
http.cookie	http.sec_websocket_protocol	
http.cookie_pair	http.sec_websocket_version	**ICMP**
http.date	http.server	icmp.addr_entry_size
http.host	http.set_cookie	icmp.address_mask
http.last_modified	http.ssl_port	icmp.checksum
http.leading_crlf	http.subdissector_failed	icmp.code
http.location	http.time	icmp.data_time
http.next_request_in	http.transfer_encoding	icmp.ext
http.next_response_in	http.unknown_header	icmp.ext.checksum
http.notification	http.upgrade	icmp.ext.data
http.prev_request_in	http.user_agent	icmp.ext.length
http.prev_response_in	http.www_authenticate	icmp.ext.res
http.proxy_authenticate	http.x_forwarded_for	icmp.ext.version

BitTorrent		
bittorrent.azureus_msg	bittorrent.extended	bittorrent.peer_id
bittorrent.bdict	bittorrent.info_hash	bittorrent.piece.begin
bittorrent.bdict.entry	bittorrent.length	bittorrent.piece.data
bittorrent.bint	bittorrent.msg	bittorrent.piece.index
bittorrent.blist	bittorrent.msg.length	bittorrent.piece.length
bittorrent.bstr	bittorrent.msg.prio	bittorrent.port
bittorrent.bstr.length	bittorrent.msg.type	bittorrent.protocol.name

Целые числа и адреса могут быть представлены как в десятичном, так и в шестнадцатеричном формате:

frame.pkt_len > 10 эквивалентно frame.pkt_len > 0xA
ip.src == 192.168.1.1 эквивалентно ip.src == 0xC0. 0xA8.0x1.0x1

В случае логических значений true эквивалентно 1, а false эквивалентно 0. В аппаратных адресах числа могут разделяться символами двоеточия (:), точкой (.) и дефисом (-):

eth.src == aa-aa-aa-aa-aa-aa эквивалентно eth.src == aa:aa:aa:aa:aa:aa

В качестве адресов узлов можно вместо числовых адресов использовать символьные адреса, но при этом необходима доступная служба разрешения имён:

ip.dst eq www.habrahabr.ru and ip.src == 192.168.1.1

Wireshark поддерживает огромное количество полей и в лабораторной работе представлена только незначительная их часть. Весь перечень поддерживающихся полей можно посмотреть на сайте разработчиков в разделе *Документация* [4].

Анализ трафика в *Wireshark*

Анализ трафика в *Wireshark* является нетривиальным процессом и в основном базируется на группах команд *Analyze* и *Statistics* меню *Wireshark*. Алгоритм анализа сильно зависит от поставленной задачи, а полученные при этом результаты могут представляться не только в общем, но и графическом виде.

В качестве примера проанализируем процесс обмена информацией между тестовым сервером **http://v4.speedtest.reliableservers.com** и клиентским узлом. С точки зрения пользователя обмен информацией будет заключаться в загрузке файла **10Mbtest.bin**, имеющего размер 10 Мбайт (рисунок 6.5).

Рисунок 6.5 – Загружаемый тестовый файл

Неудивительно, что кроме пакетов, доставляющих файл, на сетевой адаптер также приходит множество другой информации. Поэтому, для того чтобы эта побочная информация не мешала, необходимо применить фильтры отображения.

Установим удобный нам формат представления времени командами *View → Time Display Format → Seconds Since Beginning of Capture* и *View →Time Display Format → Microseconds*. Зная имя файла, можно найти соответствующий *HTTP*-запрос и определить номер *TCP*-соединения для фильтрации (рисунок 6.6).

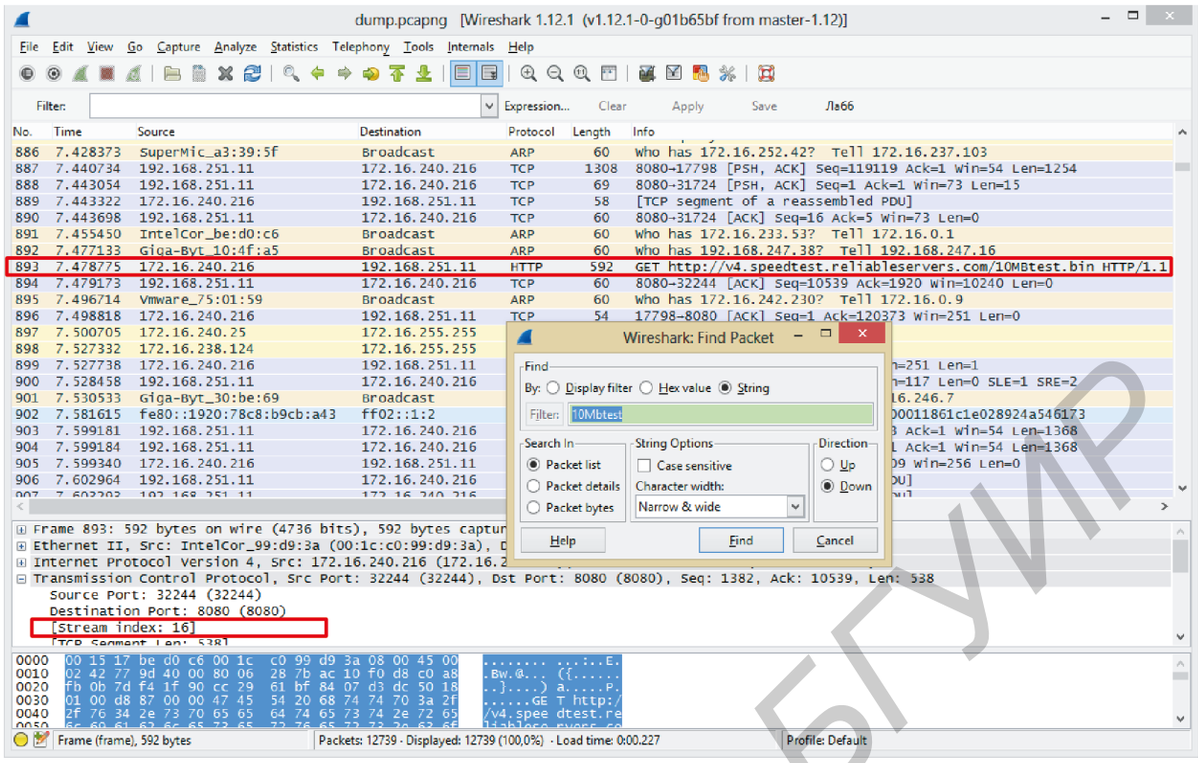


Рисунок 6.6 – Поиск номера *TCP*-соединения

Для отображения только нужной информации необходимо применить фильтр по номеру *TCP*-соединения и номеру кадра. Необходимый фильтр будет иметь вид **(tcp.stream == 16) and (frame.number > 600)**.

Инструмент просмотра информации *TCP*-соединения (команда *Analyze* → *Follow TCP Stream*). В результате выполнения этой команды отображается окно (рисунок 6.7), в котором представлен весь «диалог» между узлами в рамках указанного *TCP*-соединения. Для анализа информации можно выбирать наиболее удобную форму её представления.

Как видно из рисунка 6.7, запрос к серверу базировался на протоколе *HTTP*, из заголовка которого уже можно почерпнуть достаточно информации для анализа.

Инструмент просмотра информации о сессиях узлов (команда *Statistics* → *Conversations*). В результате выполнения команды отображается окно (рисунок 6.8) с перечнем всех захваченных сессий узлов. Здесь можно также получить детальную информацию о *TCP*-соединениях и графики приёма-передачи информации.

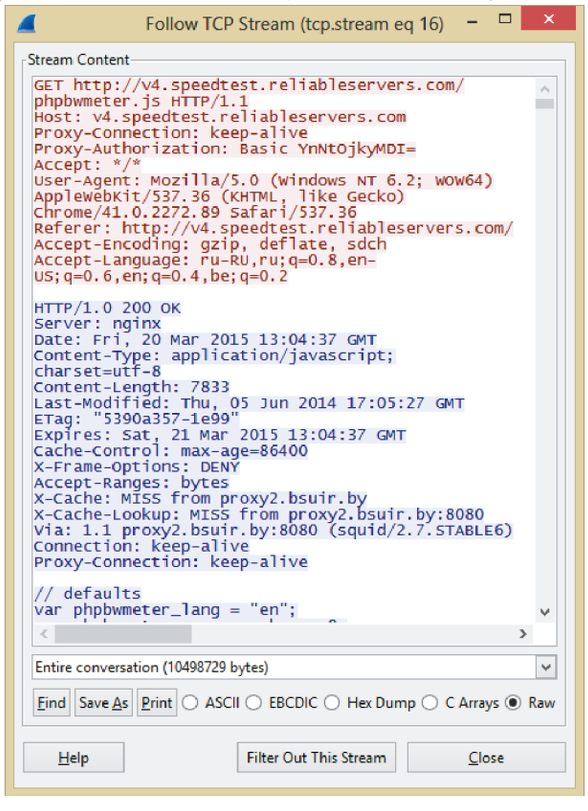


Рисунок 6.7 – Окно *Follow TCP Stream*

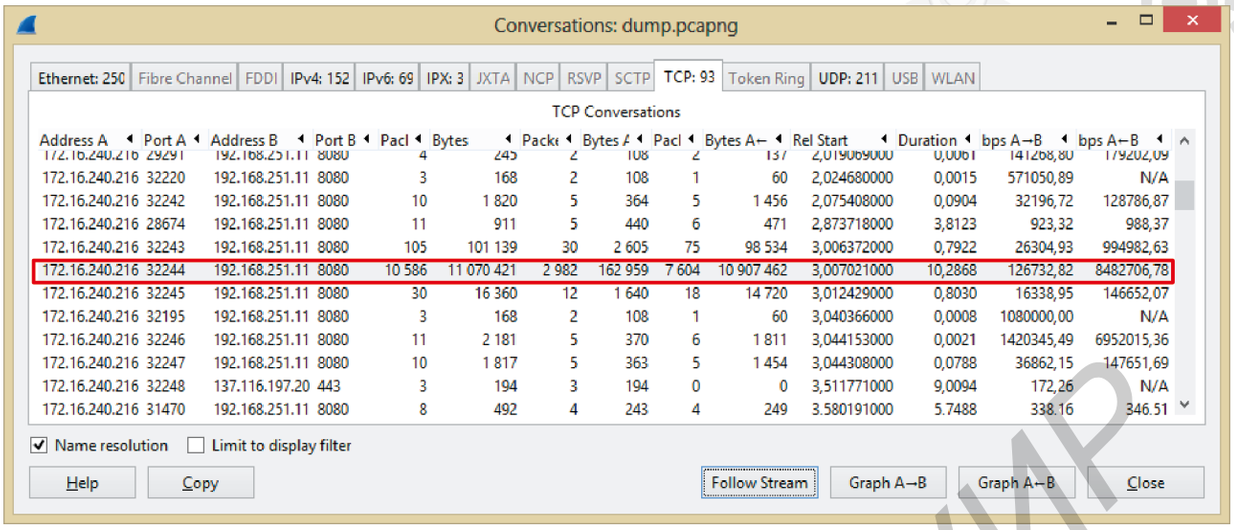


Рисунок 6.8 – Окно *Conversations*

На рисунке 6.8 красным прямоугольником выделена строчка, соответствующая *TCP*-соединению, в рамках которого был загружен файл **10Mbtest.bin**. Здесь можно узнать, что от клиентского узла серверу было передано 2 982 пакета (162 959 байт), а от сервера клиенту – 7 604 пакета (10 907 462 байт). Общее время соединения составило 10,2868 с, а средняя скорость загрузки равна 8482706,78 байт/с.

Построение графиков потоков трафика осуществляется командой *Statistics → IO Graph*. Графики, построенные с помощью этого инструмента, отображают количество пакетов, байтов или битов, захваченных в единицу времени и удовлетворяющих установленным условиям (фильтрам). Окно *IO Graph* представлено на рисунке 6.9. В нём построен график, соответствующий времени скачивания указанного ранее файла (чёрная линия), при этом красная линия соответствует пакетам, которые содержат этот файл. Как видно из рисунка 6.9, в некоторые моменты времени были получены пакеты, не относящиеся к актуальному *TCP*-соединению (чёрный график выше красного).

Инструмент *IO Graph* позволяет строить намного более сложные графики (при значении параметра *Y Axis/Unit*, равном *Advanced...*, и указании аналитического описания интересуемой зависимости). Также присутствует возможность менять стиль графиков, масштабировать их и сохранять результаты в виде файлов изображений.

Инструмент анализа *TCP*-соединения (команда *Statistics → TCP StreamGraph*). Представляет собой набор следующих графиков для анализа: *Time/Sequence Steven's-style*, *Time/Sequence tcptrace-style*, *Throughput Graph*, *Round-trip Time* и *Window Scaling*.

Time/Sequence Graph Steven's-style выглядит как наклонная кривая, состоящая из точек (рисунок 6.10). Координаты каждой точки графика – это значение *Sequence number TCP* сегмента (номер первого байта данных сегмента в общем потоке, ось *Y*) и время его захвата в секундах (ось *X*).

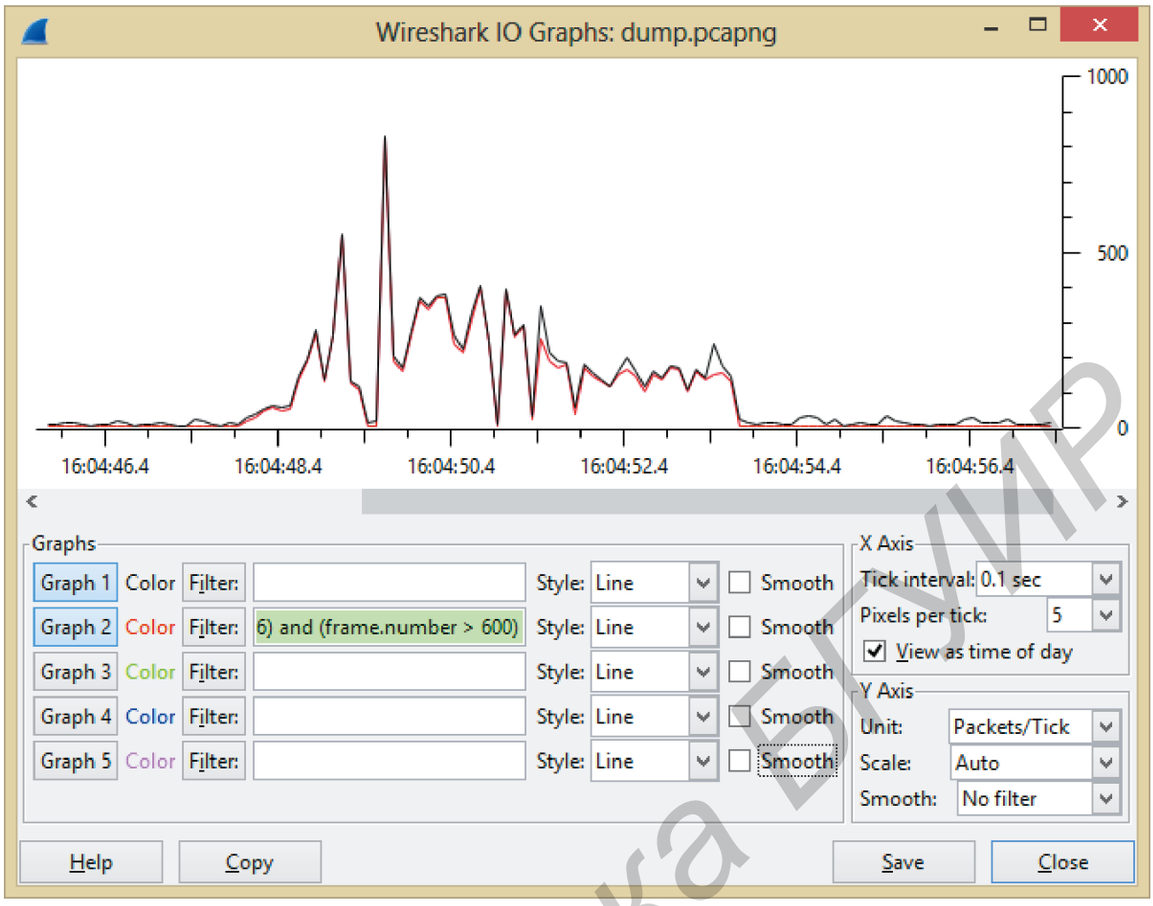


Рисунок 6.9 – Окно *IO Graph*

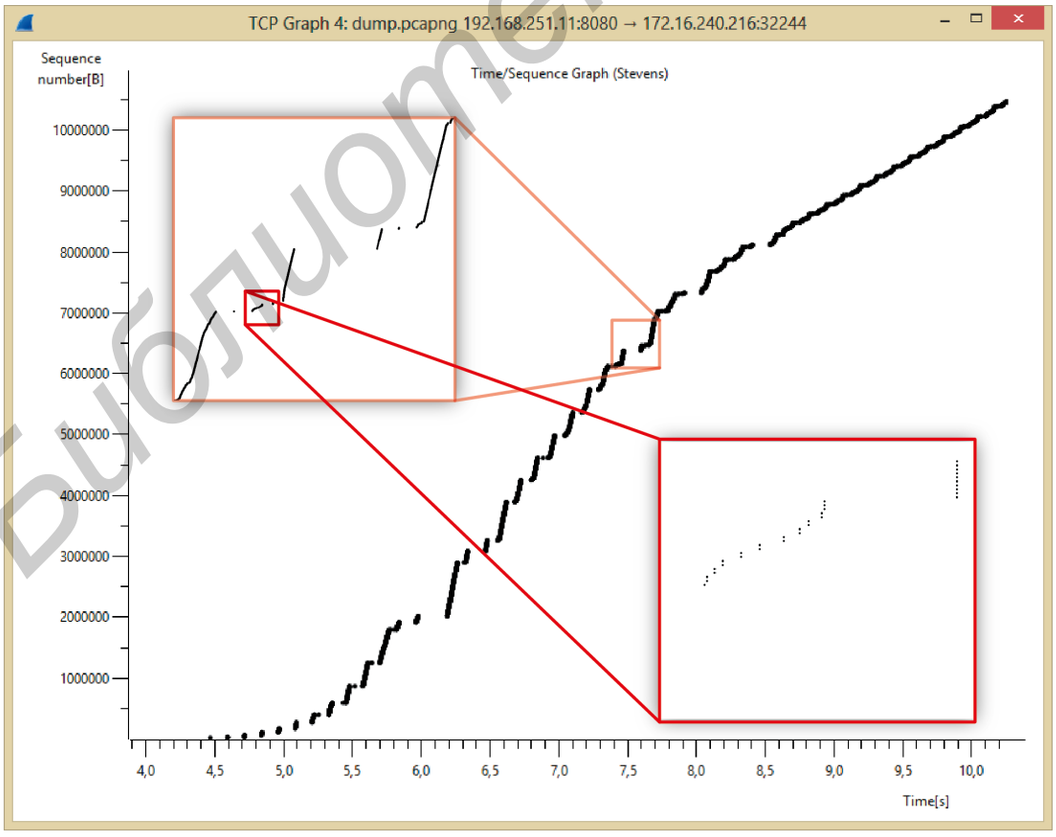


Рисунок 6.10 – График *Time/Sequence Graph Steven's-style*

На графике учитываются только сегменты с данными одного *TCP*-соединения, перемещавшиеся в определенном направлении (от сервера к клиенту или наоборот).

На любом участке можно рассчитать скорость передачи данных (*Sequence number* делённая на *Time*, получаем байт/с). Как следствие, по изменениям наклона участков кривой можно судить об изменениях скорости передачи данных.

В идеальных условиях график выглядит как диагональная линия с большим углом наклона, однако на практике это не всегда так. По аномалиям на кривой графика можно выявить задержки в передаче данных, потери сегментов и их повторные отправки (*retransmission*).

Так как график на рисунке 6.10 отображает загрузку файла, то можно сделать выводы о неравномерности скорости и отсутствии повторной пересылки пакетов.

Time/Sequence Graph tcptrace-style напоминает предыдущий график и предназначен для более полного анализа возможных проблем (рисунок 6.11). В нём также выводятся значения *Sequence number* сегментов потока данных на временной шкале. Кроме того, в него добавился ещё один атрибут сегмента – его размер. Поэтому сегменты отображаются уже не точками, а вертикальными отрезками с засечками на концах, как английская буква *I*. Основание отрезка – это *Sequence number*, а длина – размер сегмента в байтах.

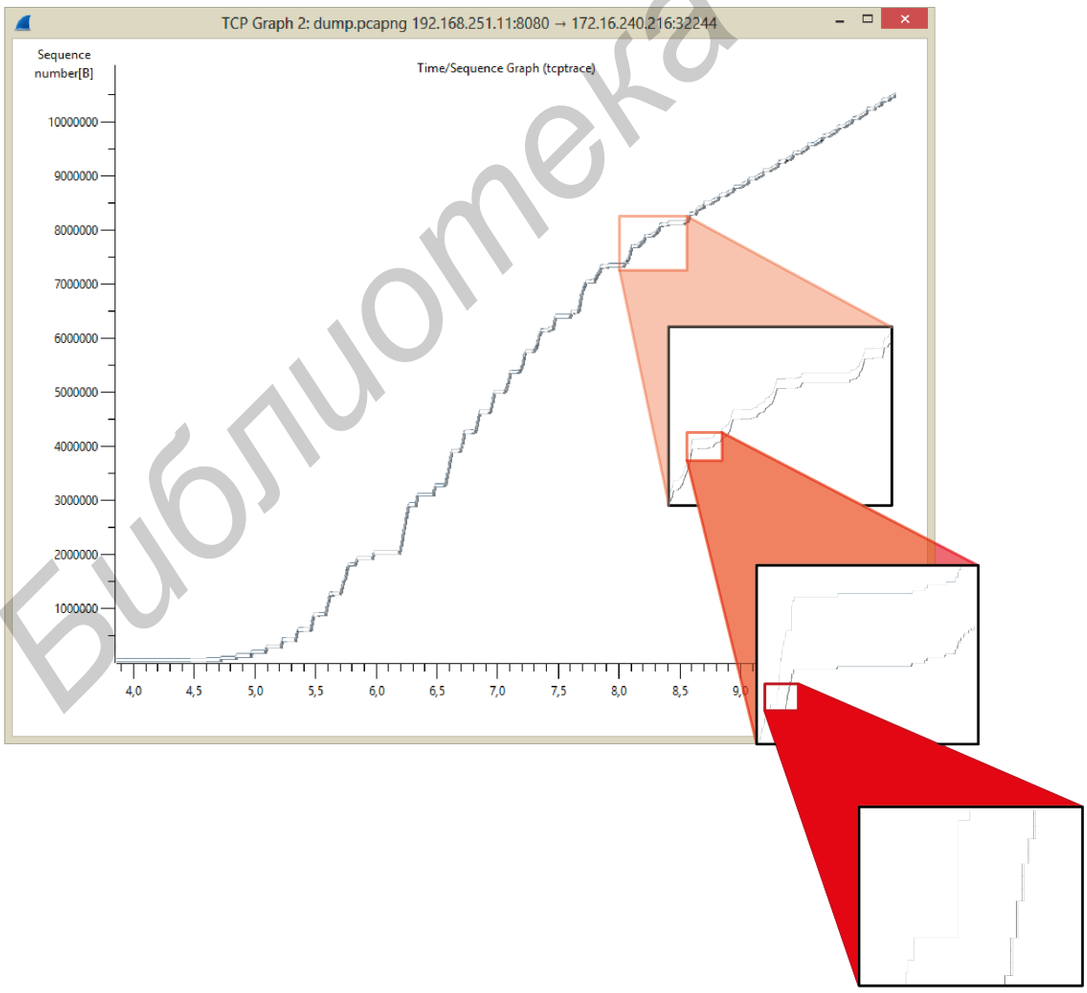


Рисунок 6.11 – График *Time/Sequence Graph tcptrace-style*

Также на графике выводится информация из обратного потока подтверждающих сегментов, ширина окна, *Acknowledgment number* и *Selective Acknowledgments*. Значения *Acknowledgment number* отображаются ступенчатой кривой, проходящей ниже сегментов данных. Каждая вершина ступени – это момент времени прихода подтверждения об общем количестве непрерывно принятых байтов получателем.

Аналогично «ступенчато» отображается размер окна принимающей стороны. Кривая проходит выше потока данных. Вершина ступени – это сумма значений *Acknowledgment number* и ширины окна подтверждающего сегмента.

В общем виде график представляет собой «коридор» из двух ступенчатых кривых, внутри которого перемещаются сегменты с данными. Сужение «коридора» говорит об уменьшении размера окна приёма, расширение – об обратном.

Throughput Graph выглядит как множество точек, иногда расположенных весьма хаотично (рисунок 6.12). Координаты каждой точки – это расчётная скорость перемещения сегмента в потоке данных (ось *Y*) и время его захвата (ось *X*).

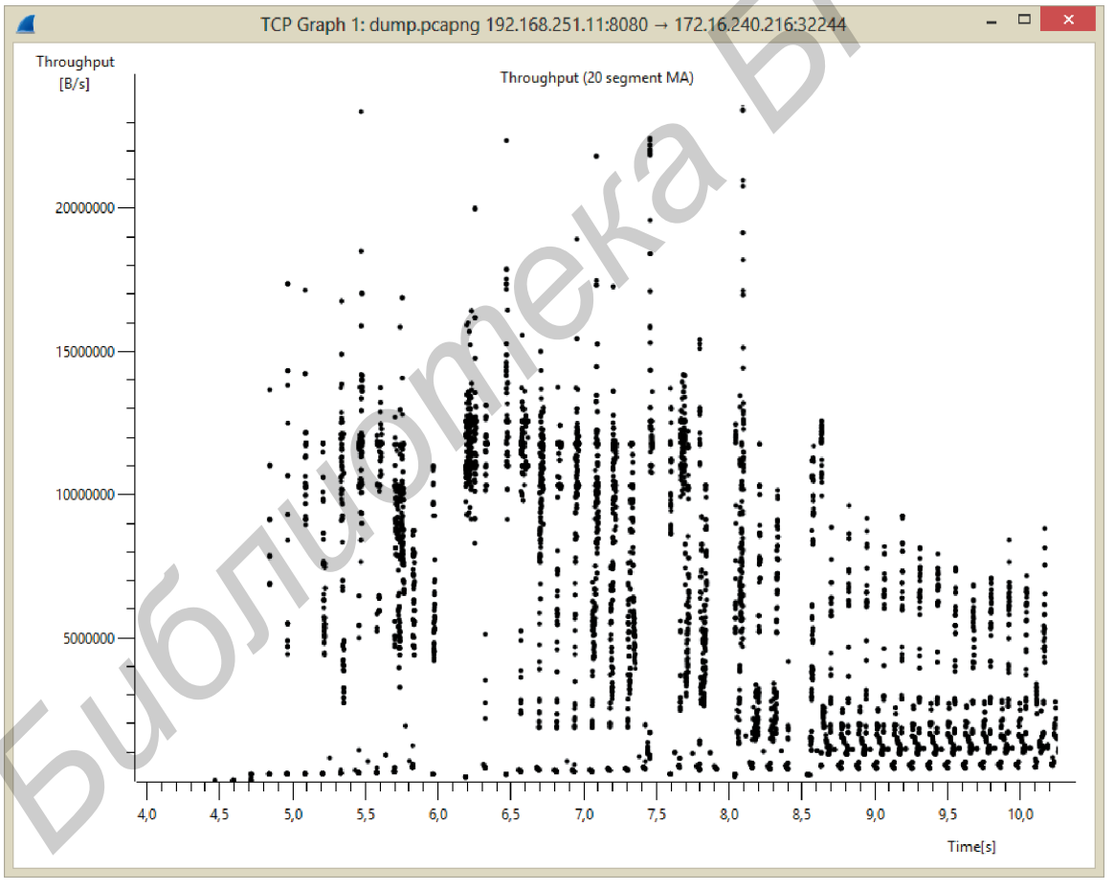


Рисунок 6.12 – График *Throughput Graph*

Для сглаживания колебаний на графике фиксируются не реальные, а усреднённые значения скорости. Используется усредняющая функция скользящего среднего (*Moving Average*, *MA*) на 20 значений за предыдущий период. То есть средняя скорость *n*-го сегмента равна сумме длин всех сегментов от *n* до *n* – 20, делённая на разницу времени между их захватом.

Round Trip Time (***RTT***) – это время, прошедшее между отправкой сегмента с данными и получением подтверждения о его успешной доставке. *Round Trip Time Graph* показывает *RTT* (ось *Y*) по каждому сегменту из потока данных (рисунок 6.13). Идентификатором сегмента выступает его *Sequence number* (ось *X*).

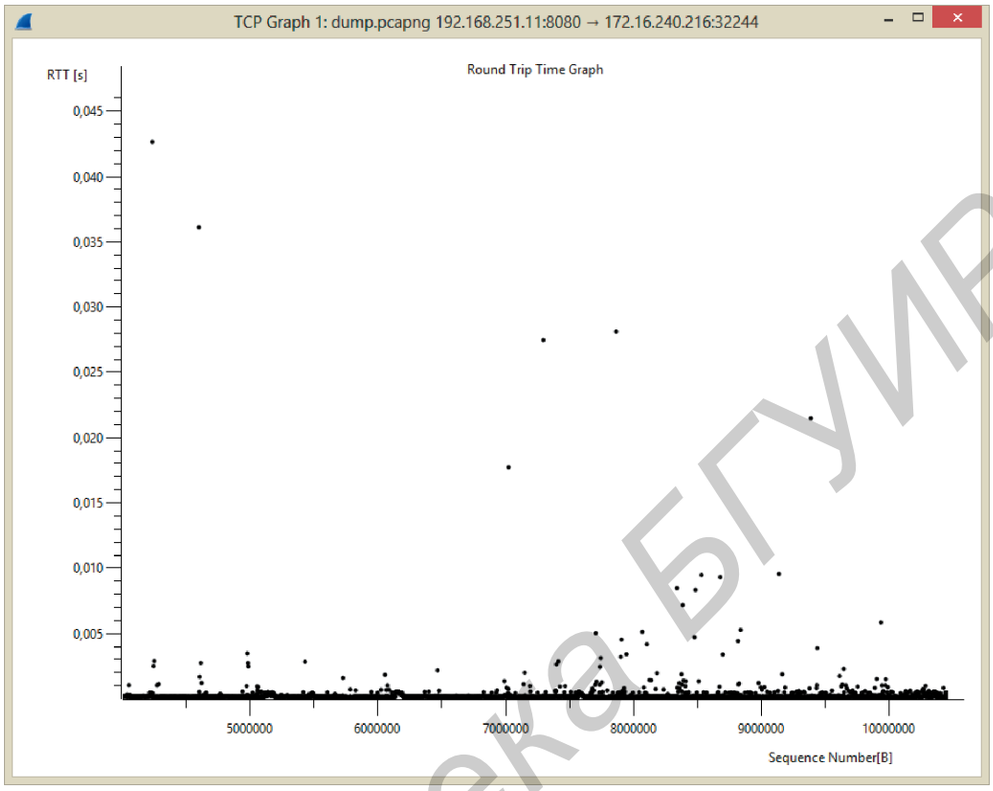


Рисунок 6.13 – График *Round Trip Time Graph*

При нормальных условиях большая часть точек концентрируется в нижней части графика.

Все графики связаны между собой следующим отношением:

$$Throughput = Window\ size\ /\ RTT$$

Координаты каждой точки *Window Scaling Graph* – это размер окна (ось *Y*) сегмента на момент времени его захвата (ось *X*). В *Window Scaling Graph* присутствует информация о проблемных случаях сокращения размера окна до критичных размеров.

Также полезным инструментом для анализа *TCP*-соединения является *Statistics/Flow Graph...*, однако в рамках данной лабораторной работы он не рассматривается.

6.3 Практическая часть лабораторной работы

Для выполнения практической части лабораторной работы необходимо:

1 Запустить анализатор сетевого трафика *Wireshark*, изучить его графический интерфейс и функциональные возможности. Запуск *Wireshark* необходимо производить с правами администратора.

2 Подготовить *Wireshark* к захвату сетевого трафика на интерфейсе, подключённом к сети, имеющей доступ в сеть *Internet*.

3 Осуществить захват трафика во время начала сеанса работы с указанным преподавателем узлом.

4 Выполнить анализ захваченного трафика, который включает в себя:

4.1 Анализ процесса обмена данными с *DNS*-сервером. Для этого нужно рассмотреть сообщение-запрос к *DNS*-серверу и сообщение-ответ от него.

4.2 Анализ процесса установления *TCP*-соединения с *HTTP*-сервером сайта (принцип «тройного рукопожатия»). Пояснить значения полей заголовка *TCP*-сегментов при этом процессе.

4.3 Анализ диалога узла с *HTTP*-сервером и процесса получения *HTML*-страницы. Для этого нужно просмотреть *HTTP*-сообщения, отправленные на сервер и полученные от него.

4.4 Получение графиков *Statistics/IO Graph*, *Time/Sequence Steven's-style*, *Time/Sequence tcptrace-style*, *Throughput Graph*, *Round-trip Time* и *Window Scaling*.

5 Осуществить захват трафика, при котором начать скачивание файла с сервера, указанного преподавателем. Длительность захвата установить в анализаторе вручную.

6 С помощью инструмента *Statistics*/*Endpoints* и правильно применённых фильтров определить объём трафика, переданного и принятого на рассматриваемом интерфейсе.

7 С помощью инструментов *Statistic*/*IO Graphs* и *Statistics/TCP StreamGraph*, а также правильно применённых фильтров построить графики, позволяющие проанализировать процесс приёма файла по сети.

8 Оформить отчёт о проделанной работе.

6.4 Содержание отчёта

Отчёт о выполненной работе должен содержать:

1 Титульный лист. Цель лабораторной работы.

2 Описание подключения интерфейса к локальной сети БГУИР и (или) настройки подключения к сети *Internet*.

3 Список *PDU*, захваченных анализатором *Wireshark* на сетевом интерфейсе при выполнении пп. 3...4 практической части лабораторной работы с ручными пометками тех *PDU*, анализ которых осуществлялся (не более двух страниц, в случае большого объёма трафика разрешается отфильтровать ненужные *PDU* средствами *Wireshark*).

4 Представленные в виде таблиц сообщения *DNS*-сервера, касающиеся рассматриваемого сайта. Предлагаемый вид таблицы приведён в таблице 6.4.

5 Представленные в виде таблиц *TCP*-сегменты, поясняющие процесс установления *TCP*-соединения между узлом сети и сервером (не более одной страницы). Оформление таблиц выполнять по аналогии с таблицей 6.4.

Таблица 6.4 – Структура заголовка *DNS*-сообщения

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Transaction ID															
QR	*OPCODE*				*AA*	*TC*	*RD*	*RA*	*Z*			*RCODE*			
Questions															
Answers RRs															
Authority RRs															
Additional RRs															

6 Представленные в виде таблиц заголовки *PDU*, которые поясняют процесс обмена информацией между узлом и *HTTP*-сервером и процесс получения *HTML*-страницы (не более двух страниц).

7 Графики *Statistics/IO Graph*, *Time/Sequence Steven's-style*, *Time/Sequence tcptrace-style*, *Throughput Graph*, *Round-trip Time* и *Window Scaling*.

8 Вывод по работе, в котором необходимо указать наиболее сложный этап лабораторной работы.

6.5 Контрольные вопросы

1 Что такое сниффер?

2 Почему использование сниффера в целях диагностики сети в сетях с коммутаторами затруднительно? Каким образом можно решить эту проблему?

3 На каком уровне (или каких уровнях) модели *OSI* сниффер *Wireshark* позволяет анализировать трафик?

4 В чём отличие фильтров захвата от фильтров отображения в снифферe *Wireshark*?

5 Какие инструменты анализа работы сети присутствуют в снифферe *Wireshark*, кроме перечисленных в теоретической части данной лабораторной работы?

6.6 Литература

1 Sanders, Chris. Practical packet analysis : using Wireshark to solve real-world network problems / Chris Sanders. – San Francisco : No Starch Press, Inc., 2007. – 255 p.

2 Wireshark. Capture Filters [Электронный ресурс]. – 2014. – Режим доступа : http://wiki.wireshark.org/CaptureFilters.

3 Wireshark – приручение акулы [Электронный ресурс]. – 2013. – Режим доступа : http://habrahabr.ru/ company/pentestit/blog/204274/.

4 Wireshark. Display Filters [Электронный ресурс]. – 2014. – Режим доступа : https://wiki.wireshark.org/DisplayFilters.

5 Фильтры отображения для сетевых анализаторов (Wireshark, Paketyzer) [Электронный ресурс]. – 2014. – Режим доступа : http://habrahabr.ru/post/211292/.

ЛАБОРАТОРНАЯ РАБОТА №7
IPv4-АДРЕСАЦИЯ И СТАТИЧЕСКАЯ МАРШРУТИЗАЦИЯ С ИСПОЛЬЗОВАНИЕМ CISCO 1841 INTEGRATED SERVICE ROUTER

7.1 Цель работы

Изучить основные технические и функциональные характеристики маршрутизаторов на примере *Cisco 1841 Integrated Service Router* (далее – *Cisco 1841*), а также получить навыки настройки *Cisco 1841* для реализации взаимодействия подсетей в среде моделирования *Cisco Packet Tracer*.

7.2 Краткие теоретические сведения

IPv4-адресация

Каждый узел, обменивающийся данными в сети, имеет **сетевой адрес** (***network address***, иногда его называют **логическим адресом**). Сетевым он называется потому, что он обеспечивает адресацию на сетевом (третьем) уровне *OSI*.

В стеке протоколов *TCP/IP* протоколом сетевого уровня является протокол *IP*. Соответственно сетевой адрес в *IP*-сетях принято называть ***IP*-адресом** (***IP-address***), а любое устройство, которое может принимать и отправлять *IP*-пакеты, – ***IP*-узлом** (***IP-host***). Сегодня существуют две версии протокола: *IPv4* и *IPv6*. В дальнейшем, говоря про *IP*, будем иметь в виду *IPv4*.

IP-адрес представляет собой 32-битовое (4-байтовое) число и обрабатывается аппаратной частью узлов в **двоичной форме**. Для удобства была введена **десятичная нотация**, в которой байты (октеты) записываются как десятичные числа, разделённые точками. Например, 192.168.7.129 – это десятичная запись двоичного адреса 11000000 10101000 00000111 10000001. Каждый сетевой интерфейс должен использовать уникальный *IP*-адрес в рамках одного устройства и в рамках одного сегмента сети. Это сделано с целью исключения неопределённости адресации.

IP-адрес состоит из двух частей: **адреса сети** (***net address***) и **адреса узла** (***host address***). Для разделения этих частей используется **маска сети** (***net mask***) – 32-битовое число, представляющее последовательность двоичных единиц для указания сетевой части адреса и двоичных нулей для указания узловой части адреса. Например, маска сети 255.255.255.0 (в двоичной нотации – 11111111 11111111 11111111 00000000) определяет, что адрес сети содержит 24 бита, а адрес узла – 8 бит. Для рассматриваемого ранее адреса это означает, что 192.168.7 – сетевая часть (адрес сети принято записывать 192.168.7.0), а 129 – адрес узла в этой сети.

В качестве *IP*-адреса узла нельзя использовать адрес сети (адрес, содержащий двоичные нули в узловой части адреса), а также **широковещательный адрес сети** (***network broadcast***), содержащий двоичные единицы в узловой части адреса. Эти *IP*-адреса входят в адресное пространство каждого сегмента сети, однако не могут быть назначены ни одному из её узлов. Таким образом, количество устройств в сети будет на два меньше, чем доступных *IP*-адресов.

Стандарт *RFC* 791 определяет три класса сетей для адресов, которые используются отдельными узлами. Такие адреса называют **одноадресатными** (***unicast address***). Эти три класса называются *A*, *B* и *C*, их характеристики отображены в таблице 7.1. В спецификации *TCP/IP* определяются также адреса класса *D* (**многоадресатные**) и класса *E* (экспериментальные).

Таблица 7.1 – Характеристики классов сетей *A*, *B* и *C*

Характеристика	Класс A	Класс B	Класс C
Диапазон значений первого октета	1...126	128...191	192...223
Адресное пространство	1.0.0.0 ... 126.0.0.0	128.0.0.0 ... 191.255.0.0	192.0.0.0 ... 223.255.255.0
Стандартная маска сети	255.0.0.0	255.255.0.0	255.255.255.0
Количество сетей в классе	$2^7 - 2$ (126)	2^{14} (16 384)	2^{21} (2 097 152)
Количество узлов в сети	$2^{24} - 2$ (16 777 214)	$2^{16} - 2$ (65 534)	$2^8 - 2$ (254)
Размер адреса сети (в байтах)	1	2	3
Размер адреса узла (в байтах)	3	2	1

Для адресации узлов в частных сетях (*private networks*), изолированных от *Internet* или имеющих доступ к *Internet* через ограниченное количество выделенных адресов, используются специально выделенные диапазоны *IP*-адресов, указанные в таблице 7.2.

Таблица 7.2 – Диапазоны адресов частных сетей (*address blocks for private internets*)

Класс сети	Диапазон адресов	Количество сетей
Класс *A*	10.0.0.0 ... 10.255.255.255	1
Класс *B*	172.16.0.0 ... 172.31.255.255	16
Класс *C*	192.168.0.0 ... 192.168.255.255	256

При построении сети предприятия выбор нужного класса производят с учётом количества имеющихся узлов сети и возможного их увеличения в будущем. После определения класса необходимо назначить уникальный сетевой адрес физическому интерфейсу каждого узла. Для рассмотренной ранее сети класса *C* 192.168.7.0 это будут следующие адреса: 192.168.7.1 (1-й узел), 192.168.7.2 (2-й узел), 192.168.7.3 (3-й узел) и т. д.

На практике реальное количество узлов в сети существенно отличается от значений, указанных в таблице 7.1. Это обусловлено следующими причинами:

– отсутствием необходимости организации настолько больших сетей;

– физическим ограничением длины кабеля и количества узлов;

– технологическим ограничением количества узлов (например, в сети стандарта *10Base-T* – не более 1024);

– опасностью возникновения «**широковещательных штормов**»;

– снижением **пропускной способности** сети.

Рассмотренная ситуация, когда при назначении адресов *IP*-адрес рассматривается состоящим из двух частей, называется **классовой адресацией** (***classful addressing***). Классовая адресация означает, что адрес имеет сетевую часть (которая определяет класс сети *A*, *B* или *C*) и узловую часть.

Для более эффективного использования адресного пространства был разработан метод **сегментации классов** (***subnetting***). В этом случае сеть классов *A*, *B* или *C* разбивается на несколько меньших групп *IP*-адресов, называемых **подсетями** (***subnets***, сокращение от «подразделённая сеть») или **сегментами** (***segments***). Суть метода состоит в разбиении узловой части *IP*-адреса на два фрагмента: **адрес подсети** (***subnet address***) и **адрес узла** (***host address***), при этом размер сетевой части адреса не сокращается (рисунок 7.1).

В основе метода сегментации классов лежит использование **маски переменной длины** (***variable length subnet mask***, ***VLSM***).

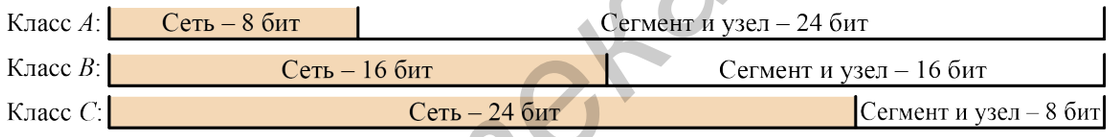


Рисунок 7.1 – Формат адреса при использовании сегментов

Альтернативным подходом к *IP*-адресации является **бесклассовая адресация** (***classless addressing***), основанная на концепции ***CIDR*** (***classless inter-domain routing***). В основе такого подхода является отказ от деления исходного адресного пространства на классы и объединение сетевой части *IP*-адреса и части, определяющей сегмент. Вместо трёх частей каждый адрес содержит две части: адрес сети/сегмента (иногда называющаяся **префиксом**) и адрес узла.

В случае разделения сети на сегменты маска будет определяться количеством узлов сегмента. Примером всех возможных масок для деления сети 192.168.7.0 класса *C* на сегменты является таблица 7.3.

Существует ещё один удобный метод записи маски, называемый **префиксной записью** (***prefix notation***), а иногда ***CIDR*-записью**. Такой вариант записи намного короче. В префиксной записи маски необходимо указать количество двоичных единиц после символа прямой косой черты « / ». Например, для маски 255.255.255.0, двоичный эквивалент которой 11111111 11111111 11111111 00000000, соответствующая префиксная запись будет выглядеть как « /24 ». Таким образом, для рассмотренного примера (см. таблицу 7.3, строку 1) можно записать «192.168.7.0 /25» вместо «192.168.7.0 с маской 255.255.255.128».

Таблица 7.3 – Возможные маски для деления сети 192.168.7.0

Биты 4-го октета	Маска	Кол-во сегментов	Кол-во узлов	Адрес сегмента/узла (последнее число)
10000000	255.255.255.128	2	126	0, 128
11000000	255.255.255.192	4	62	0, 64, 128, 192
11100000	255.255.255.224	8	30	0, 32, ... , 192, 224
11110000	255.255.255.240	16	14	0, 16, ... , 224, 240
11111000	255.255.255.248	32	6	0, 8, ... , 240, 248
11111100	255.255.255.252	64	2	0, 4, ... , 248, 252

Примечание – В обозначении **11100000**, красные биты **1** соответствуют адресу сегмента, а синие биты **0** – адресу хоста.

Расчёт адресации любой сети начинают с определения проектных условий: количества и объёма сегментов, оценки количества узлов предприятия и планов развития сети. Рассмотрим расчёт сети на нескольких примерах.

Пример 7.1 – Расчёт сегментов

Нам необходимо организовать сеть 192.168.7.0, содержащую 5 групп не более чем по 20 узлов в каждой. В этом случае рационально разбить её на 8 сегментов с максимальным количеством узлов 30 в каждом сегменте (см. таблицу 7.3). При таком разделении количество битов части сегмента равно 3 (количество сегментов: $2^3 = 8$), а количество битов части узла – 5 (количество узлов: $2^5 – 2 = 30$). Маска сети в этом случае будет равна 255.255.255.224 или /27 в префиксной записи.

Адреса рассчитанных сегментов: 192.168.7.0 (сегмент 1), 192.168.7.32 (сегмент 2), 192.168.7.64 (сегмент 3), 192.168.7.96 (сегмент 4), 192.168.7.128 (сегмент 5), 192.168.7.160 (сегмент 6), 192.168.7.192 (сегмент 7), 192.168.7.224 (сегмент 8).

Для реализации пяти сегментов можно выбрать любые пять адресов. Оставшиеся три адреса сегментов можно использовать для дальнейшего расширения сети предприятия. Затем необходимо определить адреса узлов в каждом сегменте. Например, для сегмента №2 это следующие адреса: 192.168.7.33 (узел 1), 192.168.7.34 (узел 2), 192.168.7.35 (узел 3), 192.168.7.36 (узел 4), ..., 192.168.7.62 (узел 30).

Пример 7.2 – Расчёт сети

Количество узлов сети предприятия равно 100. На предприятии имеется 5 структурных подразделений, максимальное количество компьютеров в которых равно 22. Программа развития предприятия предполагает увеличение количества компьютеров в ближайшем будущем до 150 и создание ещё одного отдела. Требуется выполнить адресацию сети с учётом этих данных.

Проектирование сети выполняется поэтапно.

Первый этап – выбор адреса сети. Адрес выбираем из пространства частных сетей. При количестве узлов не более 254 отнесём сеть к классу *C*. Выберем адрес 192.168.1.0. Маска сети в этом случае равна 255.255.255.0 или /24.

Второй этап – определение количества сегментов. В соответствии со структурой предприятия количество сегментов не превысит 8. Для нумерации 8 сегментов потребуется

расширить маску сети на 3 разряда, тогда последний октет маски для каждой подсети в двоичном представлении будет равен 11100000. Десятичное представление этого октета равно 224. Таким образом, маска подсети равна 255.255.255.224 или /27.

Третий этап – проверка соответствия оставшегося количества разрядов в узловой части адреса каждой подсети и количества узлов в сегменте. Во внимание принимается максимальное количество узлов сегмента, равное 22. Для задания адреса узла потребуется 5 двоичных разрядов. Так как узловая часть адреса сети содержит 5 нулей, а значит количество узлов равно 30, то подтвердим значение маски 255.255.255.224.

Четвёртый этап – определение адресов сегментов. Адрес сегмента №1 равен 192.168.1.0 /27, адрес сегмента №2 – 192.168.1.32 /27, адрес сегмента №3 – 192.168.1.64 /27 и т. д. Адрес последнего сегмента (№8) равен 192.168.1.224 /27. Для назначения сегментов пяти подразделениям предприятия можно выбрать любые пять адресов из полученных восьми.

Пятый этап – определение адресов узлов для каждого из сегментов. Пусть выбраны сегменты №2...6. Адресное пространство для сегмента №2 имеет вид 192.168.1.32 /27 ... 192.168.1.62 /27. Адрес 192.168.0.32 /27 является адресом сегмента, 192.168.1.63 /27 ... адресом широковещательной рассылки в этом сегменте. Таким образом, допустимыми адресами для узлов являются 192.168.0.33 /27 ... 192.168.0.62 /27. Адреса узлов для сегмента №3 равны 192.168.1.65 /27 ... 192.168.1.94 /27. Таким же образом определяем адреса узлов во всех сегментах. Если количество узлов в сегменте меньше 30, то выбираем любые адреса из рассчитанных диапазонов.

Пример 7.3 – **Расчёт сети**

Количество узлов сети предприятия равно 500. На предприятии существует 12 структурных подразделений, максимальное количество компьютеров в которых равно 40. Программа развития предприятия предполагает увеличение количества компьютеров в ближайшем будущем до 600, возможно создание ещё двух отделов. Требуется спроектировать сеть с учётом этих данных.

Проектирование сети выполним поэтапно, но в другой последовательности.

Первый этап – определение количества битов, необходимых для адресации узлов. Так как максимальное количество узлов в сегменте должно быть не менее 40, то количество битов должно быть не менее 6 (это позволит назначить адрес 62 узлам в каждом сегменте, см. таблицу 7.3).

Второй этап – определение количества битов, необходимых для адресации сегментов. Для 14 сегментов эта часть адреса должна содержать не менее 4 бит (это позволит задать адрес 16 сегментам).

Третий этап – определение маски сети. Маска сети должна содержать в узловой части нули, количество которых будет достаточным для нумерации всех сегментов и всех узлов в каждом сегменте: 4 бит + 6 бит = 10 бит. Таким образом, маска сети в двоичном представлении равна **11111111 11111111 11111**100 00000000, в десятичном – 255.255.252.0, а в префиксной форме – /22.

Четвёртый этап – определение адреса сети и адресов сегментов. Адрес сети выбираем из диапазона частных адресов. Адреса из диапазона 172.16.0.0 ... 172.31.255.255 (класс *B*) рекомендованы для сетей с количеством узлов, большим чем 254. Возьмём адрес сети, равный 172.20.0.0. В двоичном представлении это **10101100 00010100 00000**00 00000000 (выделена сетевая часть адреса). Тогда адреса подсетей будут равны:

– подсеть №0 – 172.20.0.0 /22 (двоичное 10101100 000101000 00000**00 00**000000);
– подсеть №1 – 172.20.0.64 /22 (двоичное 10101100 000101000 00000**00 01**000000);
– подсеть №2 – 172.20.0.128 /22 (двоичное 10101100 000101000 00000**00 10**000000);

– подсеть №3 – 172.20.0.192 /22 (двоичное 10101100 000101000 00000**00 11**000000);
– подсеть №4 – 172.20.1.0 /22 (двоичное 10101100 000101000 00000**01 00**000000);
– подсеть №5 – 172.20.1.64 /22 (двоичное 10101100 000101000 00000**01 01**000000);
– подсеть №6 – 172.20.1.128 /22 (двоичное 10101100 000101000 00000**01 10**000000);
– подсеть №7 – 172.20.1.192 /22 (двоичное 10101100 000101000 00000**01 11**000000);
– подсеть №8 – 172.20.2.0 /22 (двоичное 10101100 000101000 00000**10 00**000000);
– подсеть №9 – 172.20.2.64 /22 (двоичное 10101100 000101000 00000**10 01**000000);
– подсеть №10 – 172.20.2.128 /22 (двоичное 10101100 000101000 00000**10 10**000000);
– подсеть №11 – 172.20.2.192 /22 (двоичное 10101100 000101000 00000**10 11**000000);
– подсеть №12 – 172.20.3.0 /22 (двоичное 10101100 000101000 00000**11 00**000000);
– подсеть №13 – 172.20.3.64 /22 (двоичное 10101100 000101000 00000**11 01**000000);
– подсеть №14 – 172.20.3.128 /22 (двоичное 10101100 000101000 00000**11 10**000000);
– подсеть №15 – 172.20.3.192 /22 (двоичное 10101100 000101000 00000**11 11**000000).

В двоичном представлении выделены биты, определяющие номер сегмента.

Пятый этап – определение адресов узлов в каждом сегменте. Для примера определим адреса узлов для сегмента №13:

– хост №1 – 172.20.3.65 /22 (двоичное 10101100 000101000 00000011 01**000001**);
– хост №2 – 172.20.3.66 /22 (двоичное 10101100 000101000 00000011 01**000010**);
– хост №3 – 172.20.3.67 /22 (двоичное 10101100 000101000 00000011 01**000011**);
– хост №4 – 172.20.3.68 /22 (двоичное 10101100 000101000 00000011 01**000100**);

…

– хост №61 – 172.20.3.125 /22 (двоичное 10101100 000101000 00000011 01**111101**);
– хост №62 – 172.20.3.126 /22 (двоичное 10101100 000101000 00000011 01**111110**).

В двоичном представлении выделены биты, определяющие номер узла.

Адрес 172.20.3.127 /22 (двоичное 10101100 000101000 00000011 01**111111**) является адресом широковещательной рассылки для этого сегмента.

Шлюзы

Отдельные сегменты объединяются в сеть с помощью шлюзов. **Шлюзом** (***gateway***) называют аппаратно-программную систему, содержащую не менее двух сетевых интерфейсов (не обязательно аппаратных) и соединяющую разнородные участки сети. Им может служить как маршрутизатор, так и любой из компьютеров в сети, имеющий больше одного сетевого интерфейса. В случае использования компьютера в качестве шлюза, на нём необходимо настроить соответствующие службы. Чаще всего в качестве шлюза используется именно маршрутизатор.

Пусть есть два сегмента, называемые *A* и *B*, и их необходимо соединить через шлюз. В этом случае один из интерфейсов шлюза подключается к сегменту *A* и ему назначается адрес из диапазона адресов сегмента *A*, а другой – к сегменту *B* и ему назначается адрес из диапазона адресов сегмента *B*. В результате шлюз является одновременно узлом в двух сегментах сети. В настройках интерфейсов всех узлов каждого сегмента необходимо указать *IP*-адрес своего шлюза при настройке протокола *TCP*/*IP*, что позволит определить промежуточное устройство, позволяющее передать данные в другой сегмент сети (рисунок 7.2).

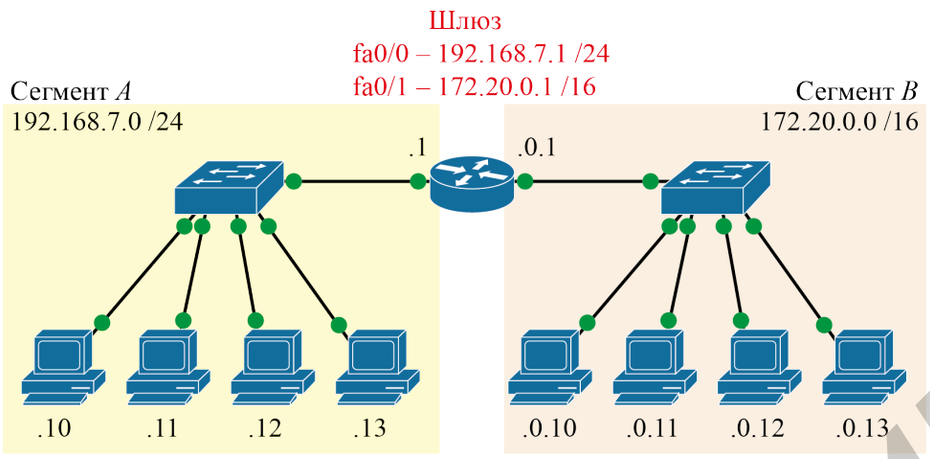


Рисунок 7.2 – Объединение двух сегментов сети с помощью шлюза

Из рисунка 7.2 видно, что логически нельзя относить к сегменту узел или устройство в целом. Однозначная принадлежность к тому или иному сегменту определена лишь у сетевого интерфейса. Один интерфейс шлюза (маршрутизатора) fa0/0 относится к сегменту *A* и имеет адрес 192.168.7.1 /24 класса *C*, а другой интерфейс шлюза fa0/1 относится к сегменту *B* и имеет адрес 172.20.0.1 /16 класса *B*. При этом в настройках интерфейса остальных узлов сегмента *A* в поле «Основной шлюз» (шлюз по умолчанию) указывается *IP*-адрес fa0/0, а сегмента *B* – fa0/1.

Основной шлюз (***default gateway***) – шлюз, на который отправляются *IP*-пакеты, для которых невозможно определить маршрут.

Статическая маршрутизация

Маршрутизация (***routing***) – процесс определения **маршрута** (***route***) передачи *IP*-пакетов в компьютерных сетях. Маршруты могут задаваться административно (**статический маршрут**, ***static route***) либо вычисляться с помощью алгоритмов, базируясь на информации о топологии и состоянии сети, полученной с помощью протоколов маршрутизации (**динамические маршруты**, ***dynamic route***). В данной лабораторной работе рассматриваются статические маршруты.

Статическая маршрутизация (***static routing***) – вид маршрутизации, при котором маршруты указываются администратором в явном виде при конфигурации шлюзов. Весь процесс при этом проходит без участия дополнительных протоколов.

При наличии только одного шлюза, объединяющего два и более сегмента (см. рисунок 7.2), настройки маршрутизации не требуется, т. к. шлюз «знает» обо всех подключённых сегментах и самостоятельно решает задачу выбора интерфейса для пересылки *IP*-пакетов.

При наличии нескольких сегментов и нескольких шлюзов (рисунок 7.3) возникает необходимость, помимо указания основного шлюза, указывать также статические маршруты. Например, шлюз *R*2 при пересылке *IP*-пакета из сегмента *A* в сегмент *C* отправит его на шлюз *R*1, т. к. ничего «не знает» о сегменте *C*, а *R*1 является для него основным шлюзом.

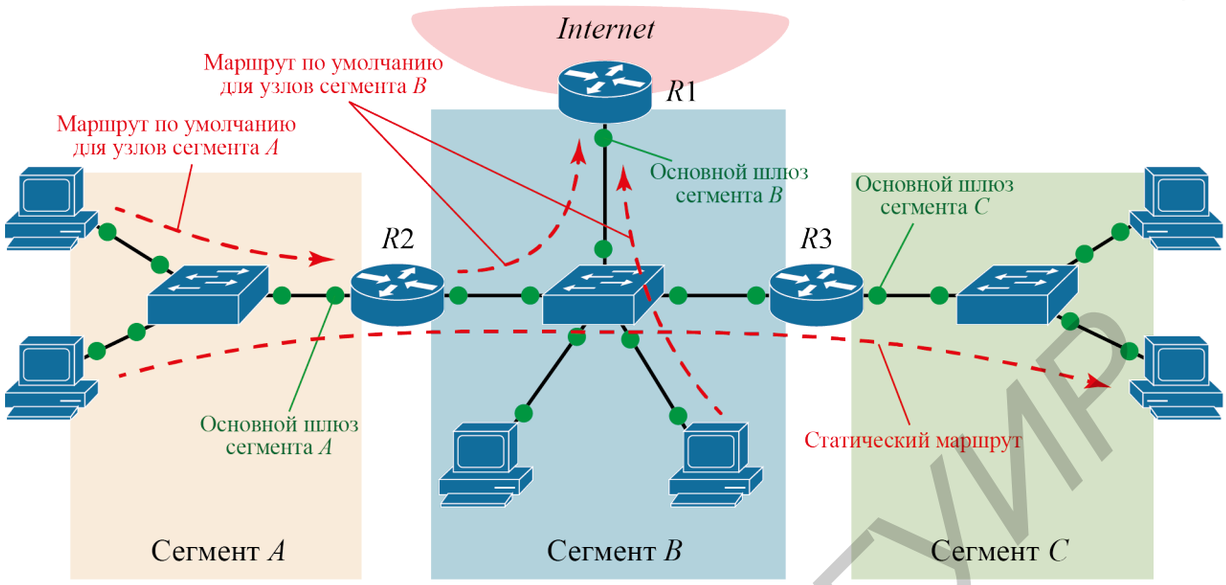


Рисунок 7.3 – Объединение нескольких сегментов сети с помощью шлюзов (маршрутизаторов)

Для обеспечения доставки пакетов от сегмента *A* к сегменту *C* на шлюзе *R2* должен быть добавлен статический маршрут, который укажет ему, куда передавать пакеты. При добавлении статического маршрута указываются: **адрес** и **маска** неизвестной шлюзу сети (сегмента), в которую должны доставляться пакеты; **адрес интерфейса следующего узла** в проектируемом администратором маршруте (или выходной интерфейс для текущего узла).

Достоинства статической маршрутизации:

– отсутствие дополнительных **накладных расходов** в сети (из-за отсутствия протоколов маршрутизации);

– мгновенная готовность (при динамической маршрутизации для построения маршрутов требуется время);

– низкая нагрузка на аппаратное обеспечение шлюзов;

– предсказуемость поведения узлов в каждый момент времени.

Недостатки статической маршрутизации:

– плохое масштабирование (при количестве шлюзов больше 3...4 процесс настройки становится трудоёмким; добавление $(n + 1)$-й сети потребует сделать $2 \times (n + 1)$ записей о маршрутах);

– низкая устойчивость к повреждениям линий связи;

– отсутствие динамического **балансирования нагрузки** в сети;

– необходимость в ведении документации о маршрутах и поддержание её в актуальном состоянии.

В реальных условиях статическая маршрутизация используется в условиях наличия основного шлюза и 1...2 сетями (сегментами).

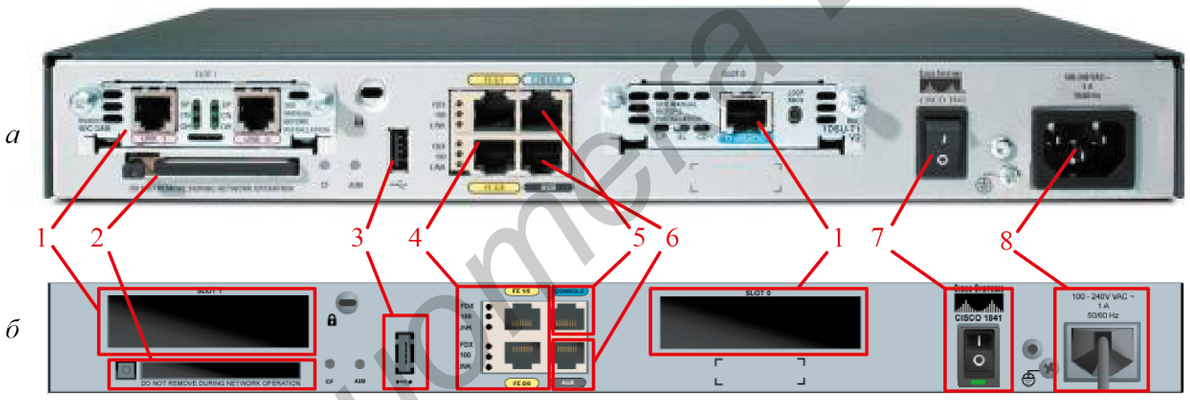
Маршрутизатор *Cisco 1841* и добавление статических маршрутов

Маршрутизатор (***router***) – сетевое устройство, объединяющее в одну сеть сегменты сети меньшего размера или сети с различной архитектурой, а также принимающее решение о маршруте следования пакетов.

На сегодняшний день в мире существует несколько десятков производителей, предлагающих свои модели маршрутизаторов, однако лидером на мировом рынке является компания *Cisco Systems*, *Inc*. Известно, что организовав сетевую академию, *Cisco* продиктовала принципы, которыми стали руководствоваться и другие производители. Изучение статической маршрутизации основано на использовании маршрутизатора начального уровня *Cisco 1841*.

Рисунок 7.4 – *Cisco 1841*

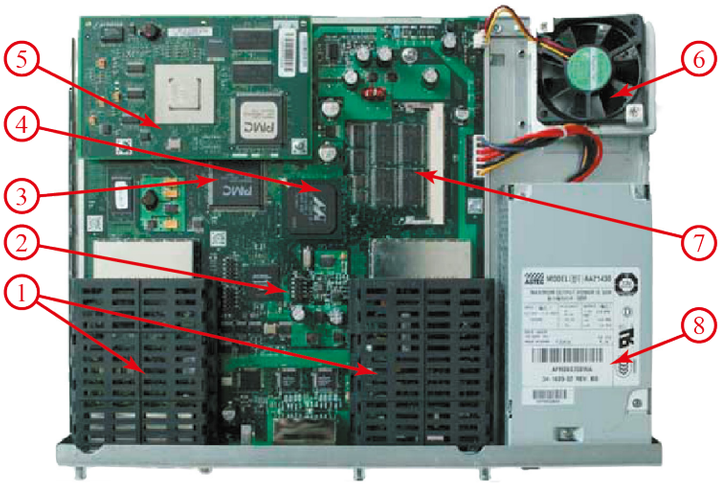
Внешний вид *Cisco 1841* представлен на рисунке 7.4. Физические интерфейсы *Cisco 1841* представлены на рисунке 7.5. Рассматриваемый маршрутизатор является модульным, что позволяет оснащать его дополнительными интерфейсами через слоты 1 (см. рисунок 7.5).



а – реального устройства; *б* – из *Packet Tracer*;
1 – порты для подключения модулей расширения *WIC*; 2 – разъём для карты памяти;
3 – порт *USB* 1.1; 4 – *RJ-45* порты *FastEthernet*; 5 – *Console*-порт *RJ-45*;
6 – *Auxillary*-порт *RJ-45*; 7 – кнопка включения питания; 8 – разъём питания
Рисунок 7.5 – Задняя панель *Cisco 1841*

Маршрутизатор *Cisco 1841* работает под управлением *Cisco IOS* (*Internetwork Operating System*) и имеет следующие внутренние компоненты: материнскую плату (*motherboard*), процессор (*CPU*), *flash*-память (используется для хранения образа *IOS*), оперативную память (*SDRAM*), постоянную память двух типов (первый тип – *NVRAM* для хранения настроек *IOS*; второй тип – *ROM* для хранения других данных), модули расширения, блок питания и т. д. Некоторые компоненты показаны на рисунке 7.6.

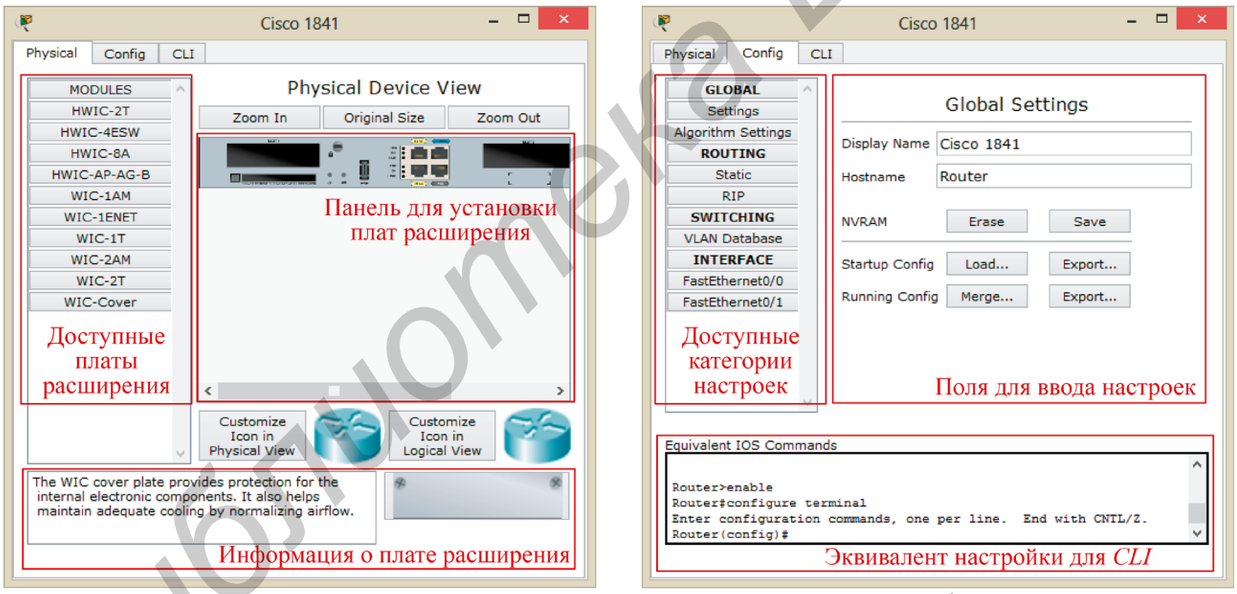
Packet Tracer позволяет настраивать как аппаратное, так и программное обеспечение *Cisco 1841*. Сделать это можно с помощью окна управления устройством, вкладок *Physical* (рисунок 7.7, *а*) и *Config* (рисунок 7.7, *б*).



1 – защитный экран для плат расширения;
2 – материнская плата; 3 – *CPU*; 4 – *NVRAM*;
5 – *AIM*; 6 – система охлаждения; 7 – *SDRAM*; 8 – БП
Рисунок 7.6 – Аппаратные компоненты *Cisco 1841*

Настройку маршрутизаторов в *Packet Tracer* также можно производить через интерфейс командной строки *IOS* (вкладка *CLI – Command Line Interface*), однако такой способ настройки не рассматривается в лабораторной работе.

Настройка аппаратного обеспечения маршрутизатора заключается в выборе и установке модулей расширения *WIC* и *HWIC*. Настройка программного обеспечения в данной лабораторной работе сводится к настройке таблицы статических маршрутов (*Routing → Static*) и сетевых адресов интерфейсов (*Interface*).



а *б*

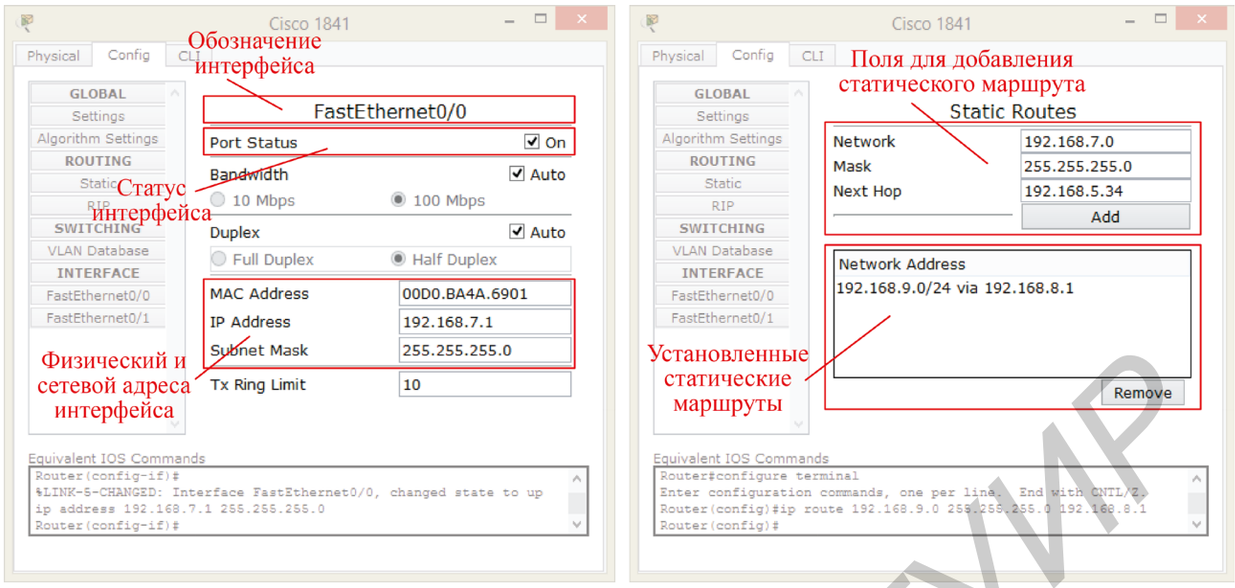
а – настройка аппаратного обеспечения; *б* – настройка программного обеспечения
Рисунок 7.7 – Настройка *Cisco 1841* с помощью графического интерфейса

Окно настройки интерфейса с пояснениями приведено на рисунке 7.8.

Статус интерфейса – аппаратное включение/выключение порта (по умолчанию интерфейс маршрутизатора отключён).

Поля для добавления статического маршрута: *network* – сеть, пункт назначения *IP*-пакетов; *mask* – маска; *next hop* – адрес интерфейса шлюза, который является следующим звеном в цепочке проектируемого маршрута.

Для добавления основного шлюза для маршрутизатора настраивается статический маршрут к сети с адресом 0.0.0.0 /0.



а *б*

а – настройка интерфейса; *б* – настройка статических маршрутов

Рисунок 7.8 – Настройка *Cisco 1841* с помощью графического интерфейса

Пример 7.4 – Настройка статической маршрутизации

Необходимо реализовать статическую маршрутизацию в сети, изображённой на рисунке 7.9. В ней определено четыре сегмента: 192.168.3.0 /24 (жёлтый), 172.20.10.0 /24 (красный), 101.14.8.0 /22 (синий) и 194.35.1.128 /27 (зелёный).

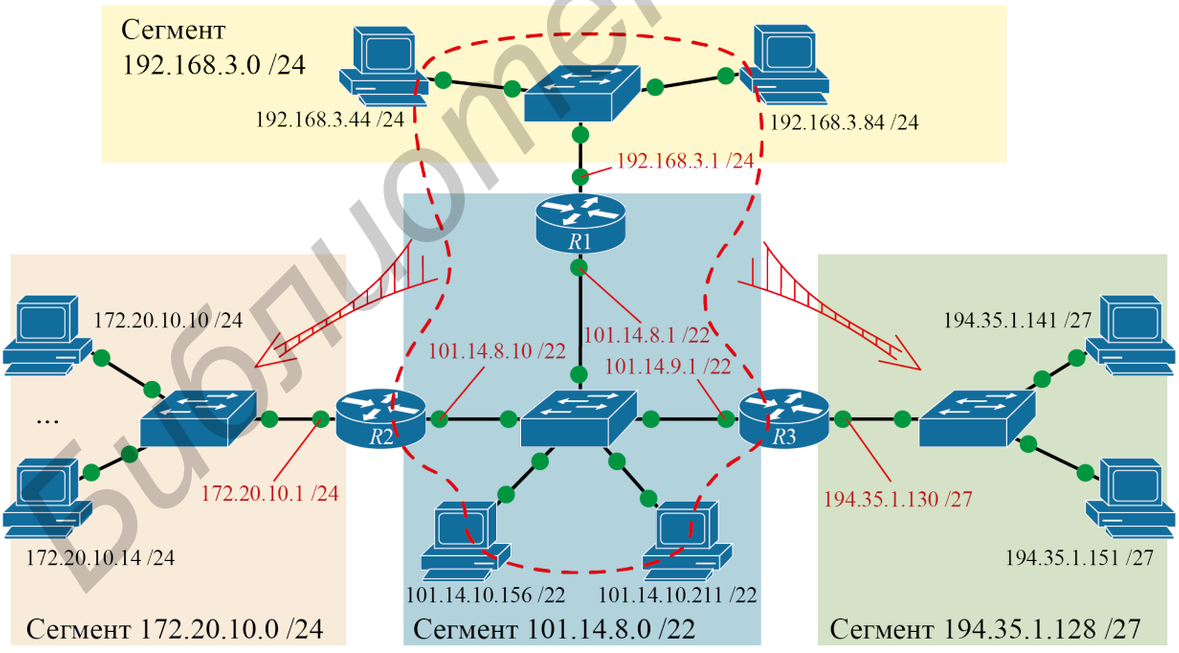


Рисунок 7.9 – Рассматриваемая в примере структура сети

Для доставки пакетов из жёлтого сегмента к остальным узлам нужно определить область видимости шлюза *R*1. Эта область показана на рисунке 7.9 красной штриховой линией (соответствует сегментам, подключённым к интерфейсам шлюза). Добавление статических маршрутов для этой области не требуется.

Для связи жёлтого и красного сегментов маршрут должен проходить через центральный синий сегмент. Однако маршрутизатор *R*1 «не знает», на какой интерфейс в синем сегменте нужно отправить пакеты, чтобы они достигли красного сегмента. Для решения задачи в маршрутизатор *R*1 необходимо внести статический маршрут, который можно сформулировать описательно: «**пакеты для сети 172.20.10.0 /24 необходимо отправить на интерфейс 101.14.8.10**». Добавляя маршрут посредством графического интерфейса (см. рисунок 7.8, *б*), в окно вносятся следующие данные: *network* – 172.20.0.0; *mask* – 255.255.255.0; *next hop* – 101.14.8.10. Маршрутизатор *R*2 «видит» красный сегмент, поэтому дальнейшей настройки передачи не требуется.

Для того чтобы ответ из красного сегмента вернулся в жёлтый сегмент, необходимо рассмотреть обратную задачу видимости шлюза *R*2. По аналогии шлюз *R*2 «не видит» жёлтый сегмент и не знает интерфейса, через который можно доставить пакеты в жёлтый сегмент. Для решения задачи на *R*2 необходимо добавить следующий статический маршрут: *network* – 192.168.3.0; *mask* – 255.255.255.0; *next hop* – 101.14.8.1.

Для полного функционирования сети необходимо также добавить следующие статические маршруты: «жёлтый – зелёный» и «красный – зелёный».

*R*1: *network* – 194.35.1.128; *mask* – 255.255.255.224; *next hop* – 101.14.9.1
*R*2: *network* – 194.35.1.128; *mask* – 255.255.255.224; *next hop* – 101.14.9.1
*R*3: *network* – 192.168.3.0; *mask* – 255.255.255.0; *next hop* – 101.14.8.1
*R*3: *network* – 172.20.10.0; *mask* – 255.255.255.0; *next hop* – 101.14.8.10

Итого для решения поставленной задачи было добавлено шесть статических маршрутов.

7.3 Практическая часть лабораторной работы

Для выполнения практической части лабораторной работы необходимо:

1 Ознакомиться с теоретическими сведениями, касающимися *IPv4*-адресации и статической маршрутизации.

2 Выбрать адрес сети исходя из проектных условий (согласно варианту, таблица 7.4).

3 Определить маску сети (исходя из количества сегментов и количества узлов в каждом сегменте), адреса сегментов, адреса узлов в каждом сегменте (согласно варианту, таблица 7.4), адреса широковещательных рассылок для каждого сегмента.

4 Построить сеть, рассмотренную на рисунке 7.2, с использованием полученных при расчёте адресов. В качестве маршрутизирующего оборудования использовать *Cisco 1841* или *Cisco 2811*.

5 Подключить с использованием маршрутизатора к полученной сети сегмент 10.0.255.0 /24 и добавить необходимые статические маршруты для полного функционирования сети (по аналогии с примером 7.4).

6 Подключить к полученной сети с использованием маршрутизатора узел 86.57.255.137 /30 и добавить необходимые статические маршруты для полного функционирования сети.

7 Выделить разноцветной заливкой (*Tools* → *Drawing Palette...*) сегменты полученной сети, а также указать на рабочем поле с помощью инструмента *Place Note* адрес каждого задействованного интерфейса.

Таблица 7.4 – Варианты заданий

Вариант	Количество узлов	Количество сегментов	Количество узлов в сегменте	Отображаемые сегменты	Отображаемые узлы сегментов
1	120	6	20	1, 2, 5	1 – 5
2	200	4	50	1, 3, 4	4 – 9
3	100	10	10	1, 5, 9	5 – 10
4	500	10	50	2, 7, 8	20 –25
5	240	7	40	1, 5, 7	7 – 9, 28, 29
6	120	5	25	2, 3, 4	13 – 16, 21, 23
7	150	6	25	2, 3, 4	17 – 21
8	200	8	25	3, 7, 8	21 – 25
9	200	7	30	1, 2, 7	10 – 12, 28, 29
10	200	10	20	3, 7, 8	2 – 4, 12, 14
11	200	10	40	4, 5, 9	1, 11, 21, 31, 40
12	500	10	50	4, 5, 8	2, 12, 22, 32, 42
13	500	5	100	2, 3, 4	90 – 94
14	1000	10	100	2, 3, 9	21, 22, 92 – 94
15	1000	100	10	15, 25, 91	2 – 6
16	2000	10	200	3, 6, 8	2, 3, 20, 30, 150
17	2000	100	20	33, 44, 68	13, 15, 17, 19, 27
18	5000	100	50	25, 45, 85	21, 39, 41, 49, 50
19	5000	50	100	2, 19, 46	1, 50, 80, 93, 99
20	10000	100	100	11, 66, 88	11, 44, 77, 88, 98

7.4 Содержание отчёта

Отчёт о выполненной работе должен содержать:

1 Титульный лист. Цель лабораторной работы.

2 Таблицу с результатами расчёта сети (по примеру таблицы 7.5).

3 Схему спроектированной компьютерной сети из *Packet Tracer* с указанием информации, необходимой для реализации этой сети другим инженером.

4 Вывод по выполненной работе, в котором необходимо указать наиболее сложный этап лабораторной работы.

Таблица 7.5 – Форма записи результатов расчёта сети

Объект	*IP*-адрес	Описание	Характеристики
Исходная сеть	___.___.___.___ /__		Всего адресов: сегментов – __, узлов – __
Сегмент 0	___.___.___.___ /__ ___.___.___.___ /__ ... ___.___.___.___ /__ ___.___.___.___ /__	– адрес сегмента – адрес узла 1 ... – адрес узла 5 – широковещательный адрес	Всего адресов: сегментов – __, узлов – __ Задействовано узлов – __ Резерв узлов – __
...	...	...	...
Сегмент *N*	___.___.___.___ /__ ...	– адрес сегмента ...	Всего адресов: ...

7.5 Контрольные вопросы и задания

1 Определить адрес сегмента, широковещательный адрес, количество узлов в сегменте и маску в десятичной форме записи для следующих *IPv4*-адресов: 192.168.100.25 /28, 172.30.10.130 /30, 10.1.113.75 /19, 128.107.14.191 /22.

2 Почему сегодня предпочтение отдаётся бесклассовой адресации?

3 Указать маску сети для пар *IP*-адресов таким образом, чтобы они находились в одном сегменте: 86.57.253.128 и 10.124.69.193, 86.57.234.248 и 10.124.69.243, 192.168.0.23 и 10.124.69.130.

4 Какие типы памяти есть у маршрутизатора и для чего используется каждая из них?

5 Определить будут ли видеть друг друга в сети узлы, имеющие следующие настройки сетевых интерфейсов:

– 10.124.69.160 / 255.255.255.128 и 10.124.69.43 /25;

– 10.124.69.248 /29 и 10.124.69.243 / 255.255.255.240;

– 10.124.69.43 /24 и 10.124.69.130 / 255.255.240.0.

6 Что для маршрутизатора означает статический маршрут в таблице маршрутизации «0.0.0.0 /0 *via* 10.30.125.6»? Какие *IP*-пакеты он отправит по этому маршруту?

7.6 Литература

1 Одом, У. Официальное руководство по подготовке к сертификационным экзаменам CCENT/CCNA ICND1 / У. Одом ; пер. с англ. – 2-е изд. – М. : ООО «И. Д. Вильямс», 2010. – 672 с.

2 Хилл, Б. Полный справочник по Cisco / Б. Хилл ; пер. с англ. – М. : ООО «И. Д. Вильямс», 2004. – 1088 с.

3 Wegner, J. D. IP Adressing and Subnetting Including IPv6 / J. D. Wegner, R. Rockell. – Syngress, 2000. – 529 p.

ЛАБОРАТОРНАЯ РАБОТА №8
ВВЕДЕНИЕ В БЕСПРОВОДНЫЕ СЕТИ МАЛОГО ПРЕДПРИЯТИЯ

8.1 Цель работы

Изучить технологию построения беспроводных компьютерных сетей масштаба малого предприятия, а также получить навыки настройки беспроводного маршрутизатора *Linksys WRT300N*.

8.2 Краткие теоретические сведения

Беспроводные сети

Беспроводные сети (***wireless network***) – компьютерные сети, реализованные без использования кабельных систем. В качестве носителя информации в таких сетях выступают радиоволны СВЧ-диапазона.

Беспроводные локальные сети стандарта *IEEE 802.11* (***wireless local-area network***, ***WLAN***, ***Wi-Fi***) – тип беспроводных сетей, реализация физического и канального уровня (*MAC*, *LLC*) которых описана в наборе стандартов *IEEE 802.11*.

На сегодняшний день определены несколько вариантов спецификаций, которые различаются используемым диапазоном частот, методом кодирования и скоростью передачи данных. Данные о вариантах представлены в таблице 8.1.

Таблица 8.1 – Различия в реализации беспроводных сетей

Стандарт	Диапазон частот	Метод кодирования	Скорость передачи данных (Мбит/с)	Ширина канала (МГц)	Год
IEEE 802.11	ИК 850 нм	–	до 2	–	1997
IEEE 802.11	2,4 ГГц	*FHSS*	до 2	22	1997
IEEE 802.11	2,4 ГГц	*DSSS*	до 2	22	1997
IEEE 802.11a	5 ГГц	*OFDM*	до 54	20	1999
IEEE 802.11b	2,4 ГГц	*DSSS*	до 11	22	1999
IEEE 802.11g	2,4 ГГц	*OFDM*	до 54	20	2003
IEEE 802.11n	2,4 ГГц, 5 ГГц	*OFDM*	до 150 [1]	20 / 40	2009
IEEE 802.11ac	5 ГГц	*OFDM*	до 866,7 [1]	до 160	2013

Современные сетевые устройства способны поддерживать одновременно несколько реализаций стандарта и автоматически переключаться между ними. В этом случае режим работы устанавливается как *802.11bg mixed* или *802.11bgn mixed* и т. п.

[1] В сетях стандартов *IEEE 802.11n* и *IEEE 802.11ac* предусмотрено несколько вариантов деления полосы частот на каналы. Максимальная скорость передачи данных указана в случае 40 и 160 МГц соответственно.

Разделение доступа к среде в беспроводных сетях

Одна из основных проблем построения беспроводных систем – это решение задачи доступа многих пользователей к ограниченному ресурсу среды передачи. Существует несколько базовых **методов разделения доступа** (их ещё называют **методами уплотнения** или **мультиплексирования**), основанных на разделении между станциями таких параметров, как пространство, время, частота и код. Задача уплотнения – выделить каждому каналу связи пространство, время, частоту и/или код с минимумом взаимных помех и максимальным использованием характеристик передающей среды. Методы разделения доступа к среде:

1 **Уплотнение с пространственным разделением**, которое основано на разделении сигналов в пространстве. То есть каждое беспроводное устройство может вести передачу данных только в границах одной определённой территории, на которой любому другому устройству запрещено передавать свои сообщения.

2 **Уплотнение с частотным разделением** (***Frequency Division Multiplexing***, ***FDM***), которое основано на том, что каждое устройство работает на строго определенной частоте, благодаря чему несколько устройств могут вести передачу данных на одной территории.

3 **Уплотнение с временным разделением** (***Time Division Multiplexing***, ***TDM***), которое основано на том, что каждый передатчик транслирует сигнал на одной и той же частоте, но в различные промежутки времени (как правило, циклически повторяющиеся) при строгих требованиях к синхронизации процесса передачи.

4 **Уплотнение с кодовым разделением** (***Code Division Multiplexing***, ***CDM***) которое предполагает, что все передатчики передают сигналы на одной и той же частоте одновременно, но с разными кодами. Данный метод реализует сложные алгоритмы в приёмнике сигнала.

5 **Механизм мультиплексирования посредством ортогональных несущих частот** (***Orthogonal Frequency Division Multiplexing***, ***OFDM***), основанный на разделении доступного частотного диапазона на большое количество поднесущих (от нескольких сот до тысяч) частот и выделении для реализации канала связи некоторого их количества, выбранного по определённому закону. Передача ведётся одновременно по всем поднесущим частотам. Распределение поднесущих частот в ходе передачи может динамически изменяться.

Расширение спектра в беспроводных сетях

Основная идея расширения спектра состоит в том, чтобы распределить информационный сигнал по широкой полосе радиодиапазона, что в итоге позволит улучшить качество передачи и значительно усложнить подавление или перехват сигнала. Существуют следующие методы расширения спектра:

1 **Расширение спектра скачкообразной перестройкой частоты** (***Frequency Hopping Spread Spectrum***, ***FHSS***). Метод основан на постоянной смене несущей частоты в пределах широкого диапазона. Последовательность несущих частот псевдослучайна и известна только приёмнику и передатчику. Метод отличается равномерной мощностью по всему диапазону частот, сложностью прослушки и

низкой скоростью передачи данных из-за больших накладных расходов по перестройке частоты и синхронизации.

2 **Прямое последовательное расширение спектра** (***Direct Sequence Spread Spectrum***, ***DSSS***). Метод заключается в замене передаваемого бита информации некоторой последовательностью (называемой расширяющейся последовательностью), которая передаётся сразу по 11 каналам шириной 22 МГц (рисунок 8.1).

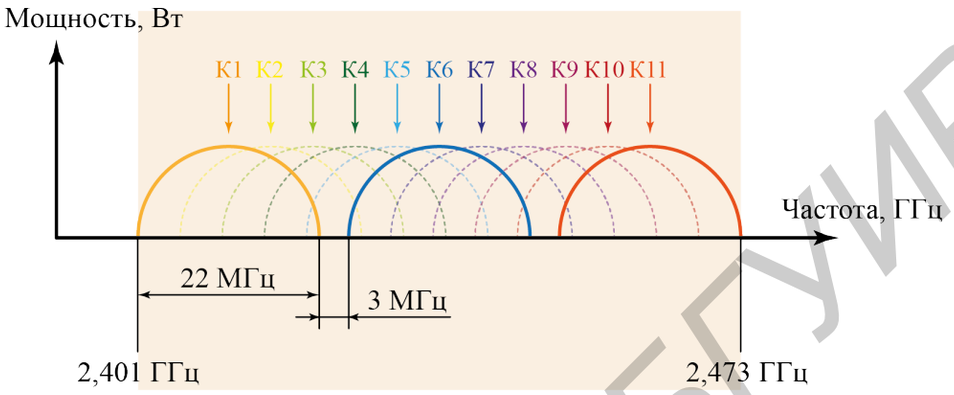


Рисунок 8.1 – Прямое последовательное расширение спектра сигнала

Данный метод позволяет усложнить прослушивание канала, снизить мощность передатчика (за счёт избыточности передаваемого кода) и выполнить мультиплексирование сигналов (за счёт использования различных заменяющих последовательностей).

Режимы организации беспроводных сетей

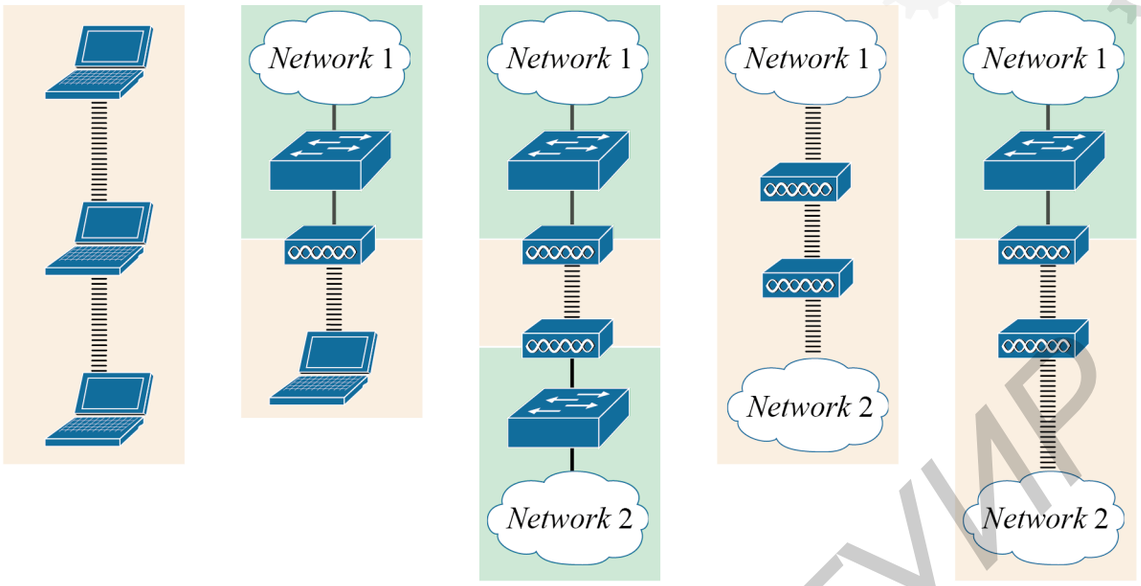
Для объединения проводной и беспроводной сред передачи данных используются беспроводные точки доступа или беспроводные маршрутизаторы. Они выступают в качестве шлюзов и позволяют организовать обмен данными между различными типами сетей.

Беспроводная точка доступа (***Wireless Access Point***, ***WAP***) – это устройство, предназначенное для обеспечения беспроводного доступа к уже существующей сети (беспроводной или проводной) или создания новой беспроводной сети. Объединение осуществляется на канальном уровне модели *OSI*.

Рассмотрим основные режимы организации беспроводных сетей:

1 Режим *Ad Hoc* (рисунок 8.2, *а*). В этом режиме узлы устанавливают связь непосредственно друг с другом. Устанавливается одноранговое взаимодействие по типу «точка-точка», и узлы взаимодействуют напрямую без применения точек доступа. При этом создаётся только одна зона обслуживания, не имеющая интерфейса для подключения к проводной сети.

Основное достоинство данного режима – простота организации: он не требует дополнительного оборудования. Режим может применяться для создания временных сетей для передачи данных на скорости не более 11 Мбит/с независимо от используемого оборудования. Дальность связи составляет не более ста метров, а скорость передачи данных быстро падает с увеличением расстояния.



а – режим *Ad Hoc*; *б* – режим инфраструктуры; *в* – режим *WDS*;
г – режим *WDS with WAP*; *д* – режим повторителя
Рисунок 8.2 – Режимы организации беспроводных сетей
(зелёным обозначены беспроводные сегменты, оранжевым – проводные)

2 Режим инфраструктуры (рисунок 8.2, *б*). В этом режиме точки доступа обеспечивают связь беспроводных узлов, а узлы не связываются непосредственно один с другим. Является основным режимом для интеграции беспроводных устройств в проводную сеть.

3 Режимы *WDS* и *WDS with WAP* (рисунок 8.2, *в*, *г*). Термин *WDS* (***Wireless Distribution System***) расшифровывается как «распределённая беспроводная система». В этом режиме точки доступа соединяются только между собой, образуя мостовое соединение. При этом каждая точка может соединяться с несколькими другими точками. Подключение узлов осуществляется только по проводной сети.

Беспроводной мост может использоваться там, где прокладка кабеля между зданиями нежелательна или невозможна. Данное решение позволяет достичь значительной экономии средств и обеспечивает простоту настройки и гибкость конфигурации при перемещении офисов.

Термин *WDS with WAP* (***WDS with Wireless Access Point***) обозначает «распределённая беспроводная система, включая точку доступа», т. е. с помощью этого режима можно организовать не только мостовую связь между точками доступа, но и одновременно подключить узлы.

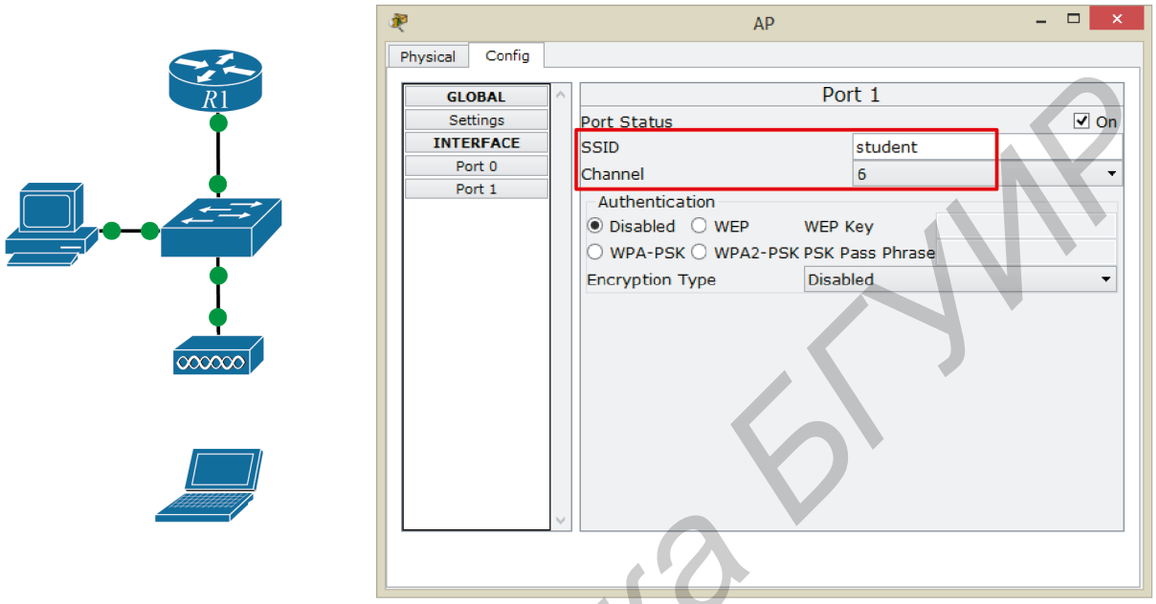
4 Режим *Repeater* (рисунок 8.2, *д*). Такой режим оправдан в ситуации, когда оказывается невозможно или неудобно соединить точку доступа с проводной инфраструктурой или с местом расположения беспроводных узлов напрямую.

Режим повторителя не включён в стандарт *IEEE 802.11*, поэтому для его реализации рекомендуется использовать однотипное оборудование (вплоть до версии прошивки) и от одного производителя. С появлением *WDS* данный режим потерял свою актуальность, потому что функционал *WDS* заменяет его.

Рассмотрим особенности подключения беспроводного узла к уже имеющейся проводной сети (192.168.0.0 /24) на примере, показанном на рисунке 8.3. Для реализации подключения необходимо настроить точку доступа. Основными параметрами любой точки доступа являются (см. рисунок 8.3, *б*):

– **идентификатор беспроводной сети** (***Self-Set Identifier***, ***SSID***);

– номер используемого канала (*channel*).



а *б*

а – топология сети; *б* – настройки точки доступа

Рисунок 8.3 – Подключение беспроводного узла к сети в режиме инфраструкуры

Устройства, координирующие работу беспроводных сетей, постоянно передают идентификатор беспроводной сети *SSID*. По умолчанию в качестве *SSID* сети используется наименование сетевого устройства либо значение по умолчанию. С целью повышения безопасности и однозначности идентификации нужной беспроводной сети целесообразно изменить *SSID*, используемый по умолчанию, на другое значение.

На стороне беспроводного узла требуется указать либо выбрать идентификатор беспроводной сети, к которой необходимо подключиться (рисунок 8.4). Подобная настройка осуществляется средствами драйвера сетевого адаптера либо операционной системы.

После выбора необходимой точки доступа и подключения ноутбук будет полноправным участником локальной сети. Убедиться в этом можно с помощью команды ping. Тем не менее настройка сетевых адресов должна быть выполнена вручную.

При использовании такого сценария настройки к проводной компьютерной сети сможет подключиться любой клиент, указав правильный *SSID*. Многие сетевые устройства поддерживают возможность запрета широковещательной трансляции идентификатора беспроводной сети, что повысит безопасность, но не исключит угрозу. При запрете на трансляцию *SSID* для подключения к беспроводной сети потребуется указать её наименование вручную.

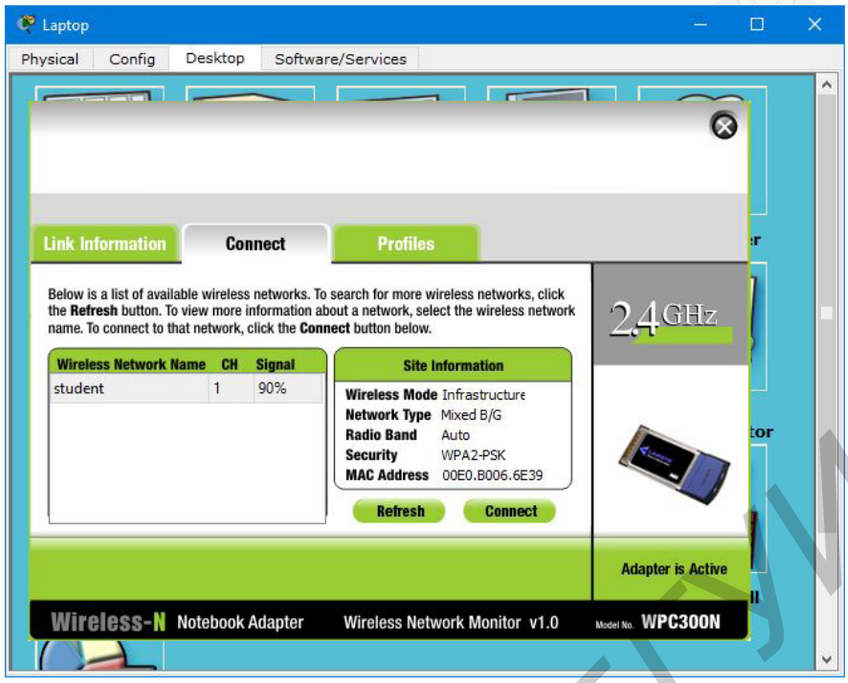


Рисунок 8.4 – Выбор беспроводной сети на оконечном оборудовании

Угрозы в беспроводных сетях

В достаточно большом количестве сетей беспроводная среда никак не контролируется. Современные беспроводные технологии предлагают ограниченный набор средств управления всей областью развёртывания сети. Это позволяет злоумышленникам, находящимся в непосредственной близости от беспроводных структур, производить целый ряд действий. Возможные угрозы:

1 Прослушивание без подключения к сети. Оборудование, используемое для подслушивания в сети, часто не сложнее того, которое используется для обычного доступа к ней. Чтобы перехватить передачу, злоумышленник должен находиться вблизи от передатчика. Прослушку такого типа практически невозможно зарегистрировать.

2 Прослушивание с подключением к сети. Часто такая атака называется атакой типа ***MITM*** (***man in the middle***, «человек посередине»). Злоумышленник подменяет идентификацию одного из сетевых ресурсов. Когда жертва атаки инициирует соединение, мошенник перехватывает его и затем завершает соединение с требуемым ресурсом, а потом пропускает все соединения с этим ресурсом через свой узел. При этом, атакующий может также посылать информацию и изменять посланную.

3 Отказ в обслуживании (***Denial of Service***, ***DOS***). В этой ситуации злоумышленник производит паразитный сигнал на рабочих частотах сети. Таким образом, во всей сети, включая сетевое оборудование и клиентские узлы, возникает такая сильная интерференция, что они не могут связываться друг с другом.

4 Глушение клиентского узла или передатчика. Глушение в сетях происходит тогда, когда преднамеренная или непреднамеренная интерференция превышает возможности отправителя или получателя в канале связи, таким образом,

выводя этот канал из строя. Глушение клиентского узла даёт возможность мошеннику подставить себя на место заглушённого клиента. Глушение передатчика предоставляет возможность подменить её атакующим передатчиком.

Криптозащита в беспроводных сетях

В беспроводных сетях применяются криптографические средства для обеспечения целостности и конфиденциальности информации.

Шифрование *WEP* (***Wired Equivalent Privacy***, секретность на уровне проводной связи) основано на алгоритме *RC4* (*Rivest's Cipher v.4*, код Ривеста), представляющем собой симметричное потоковое шифрование. Для нормального обмена пользовательскими данными ключи шифрования у узла и точки доступа должны быть идентичными.

Ядро алгоритма состоит из функции генерации ключевого потока. Эта функция генерирует последовательность битов, которая затем объединяется с открытым текстом посредством суммирования по модулю два. Дешифрация состоит из регенерации этого ключевого потока и суммирования его с шифрограммой по модулю два, восстанавливая исходный текст. Другая главная часть алгоритма – функция инициализации, которая использует ключ переменной длины для создания начального состояния генератора ключевого потока.

Шифрование *WEP* считается устаревшим в виду слабой защищённости.

Технология *WPA* (***Wi-Fi Protected Access***), призванная временно (в ожидании перехода к *802.11i*) закрыть недостатки *WEP*, обеспечивает надёжную аутентификацию пользователей с использованием стандарта *802.1x* (чаще всего это протокол *RADIUS*) и расширяемый протокол аутентификации (*EAP*). Защита включает в себя временный модуль для шифрования посредством 128-битного шифрования ключей (используется временно́й протокол целостности ключей *TKIP*). С помощью контрольной суммы сообщения (*MIC*) предотвращается изменение или форматирование пакетов данных.

Размещение точек доступа беспроводной сети

Простейшая беспроводная сеть работает только с одной точкой доступа и набором сетевых узлов. Тем не менее, когда необходимо охватить большую площадь или пространство со стенами или иными предметами, которые могут стать помехой сигналу, скорее всего количество точек доступа придётся увеличивать.

Если сеть охватывает большое открытое пространство, с первого взгляда можно разместить точки доступа на равных расстояниях друг от друга. Однако такой способ покрытия не работает при неоднородном пространстве. Оптимальный алгоритм в этом случае – начать с отдельной точки доступа на одном конце пространства и убедиться в том, что она обеспечивает приемлемую зону покрытия в пределах 50...100 м или до первого значительного отражения в перегородках. Сделать это можно при помощи компьютера со специальным тестирующим сигналом. Когда сигнал начнёт слабеть, необходимо вернуться к месту, где он стабилен, и поместить туда вторую точку.

Если вторая точка доступа не обеспечит необходимого покрытия остальной зоны, придётся добавить ещё несколько. В этом случае целью является максимум 30-процентное перекрытие в зоне между любой парой точек доступа.

Когда необходимо больше двух точек доступа в сложном пространстве, следует задуматься об использовании комбинации направленных антенн. Антенна на стене, смонтированная высоко и излучающая внутрь, может стать наилучшим вариантом для покрытия мертвой точки или расширения вашей сети в данной геометрии пространства. Также нужно обращать пристальное внимание на диаграмму направленности антенны, поскольку она может представлять собой узкий конус, а не широкую дугу (в качестве аналогии можно сравнить прожектор и лампу накаливания).

Количество пользователей сети также влияет на число необходимых точек доступа. Если к одной и той же точке доступа в одно и то же время пытается подключиться свыше 5...6 устройств, скорость передачи данных с каждого беспроводного узла падает.

Работая в режиме инфраструктуры, сеть напоминает топологию типа «звезда», в которой конечное оборудование связывается с сетью через точку доступа. Поэтому нет необходимости использовать один и тот же номер канала для всех устройств. Если можно распределить их между двумя или тремя не оказывающими помех каналами, необходимо это сделать, что улучшит быстродействие всей сети.

Внешние точки доступа

На уровень сигнала радиоканала между точкой доступа и сетевым клиентом в сети *Wi-Fi* влияют следующие факторы:

– коэффициент усиления антенны;
– излучаемая мощность;
– длина антенны;
– затухание в кабеле.

Рисунок 8.5 – Внешняя направленная антенна *Wi-Fi*

Следует помнить, что соединение *Wi-Fi* передаёт данные двунаправленно: от *WAP* к *NIC* и от *NIC* к *WAP*. Поэтому антенны и радиоустройства должны иметь способность как передавать, так и принимать радиосигнал. Характеристики антенны (рисунок 8.5) являются одинаковыми и для приёма, и для передачи, поэтому антенна, увеличивающая мощность исходящего сигнала, может также повысить чувствительность приёмника.

Внешняя антенна должна быть устойчива и к физической среде, в которой она работает. Сильный ветер может сместить антенну от цели, на которую та была изначально ориентирована, скопившийся лёд и снег могут ослабить сигнал и увеличить нагрузку на крепёжное оборудование, а солнечные лучи – ухудшить качество пластикового покрытия. Поэтому многие антенны размещены внутри обтекателей или иных корпусов, обеспечивающих дополнительную защиту.

Важно помнить, что нет особой необходимости устанавливать антенну с большим коэффициентом усиления, чем требуется на самом деле. Если возможно реализовать чистую связь при помощи антенны с низким коэффициентом усиления, сеть не станет лучше или быстрее передавать данные, оттого что точка доступа качественнее отправляет и принимает сигналы. Фактически и с более совершенной антенной качество сигнала может оставаться невысоким, поскольку она будет принимать больше шума и помех от других сетей и устройств в диапазоне 2,4 ГГц.

Стандартная всенаправленная антенна является хорошим выбором до тех пор, пока нет необходимости использовать другой тип подобного устройства. Также всегда лучше выбирать антенны от поставщика уже используемой в сети аппаратуры, чтобы предотвратить поиск несовместимости, когда оборудование откажется работать надлежащим образом.

8.3 Практическая часть лабораторной работы

Для выполнения практической части лабораторной работы необходимо:

1 Ознакомиться с теоретическими сведениями, касающимися организации беспроводных сетей масштаба малого предприятия.

2 Смоделировать в *Packet Tracer* компьютерную сеть в режиме *Ad Hoc* с настройками, указанными преподавателем. Добиться полной работоспособности сети.

3 Реализовать компьютерную сеть в режиме *Ad Hoc* на выданном преподавателем оборудовании или ноутбуках студентов и проанализировать разницу между реализацией настроек беспроводной сети в *Packet Tracer* и в полученной сети.

4 Смоделировать в *Packet Tracer* компьютерную сеть, изображённую на рисунке 8.3. Для адресации устройств использовать сеть 10.45.255.0 /24, если другая не указана преподавателем. Добиться полной работоспособности сети.

5 Реализовать компьютерную сеть, изображённую на рисунке 8.3, на выданном преподавателем оборудовании или с использованием мобильных телефонов и ноутбуков студентов и проанализировать разницу между реализацией настроек беспроводной сети в *Packet Tracer* и в полученной сети.

6 Смоделировать в *Packet Tracer* компьютерную сеть, содержащую две различные беспроводные сети, объединённые с помощью проводного или беспроводного маршрутизатора. Обеспечить взаимодействие между всеми сегментами сети, добавив необходимые статические маршруты.

7 Оформить отчёт по лабораторной работе.

8.4 Содержание отчёта

Отчёт о выполненной работе должен содержать:

1 Титульный лист. Цель лабораторной работы.

2 Изображения логических топологий, реализованных в лабораторной работе беспроводных сетей, а также информацию об адресации хостов в них.

3 Информацию о настройках беспроводных точек доступа.

4 Вывод по выполненной работе, в котором необходимо указать наиболее сложный этап лабораторной работы.

8.5 Контрольные вопросы

1 Какие существуют основные методы защиты беспроводных сетей от атак, описанных в теоретической части лабораторной работы? Каких организационных мероприятий должен придерживаться владелец беспроводной точки доступа для обеспечения приемлемого уровня безопасности?

2 Какие юридические требования и ограничения введены на организацию общественных беспроводных сетей в Республике Беларусь? Какие государственные структуры контролируют выполнение этих требований?

3 Каково назначение разделения выделенного диапазона нелицензируемых радиочастот на 11 различных каналов? В чём заключаются отличия вещания на каждом канале, кроме частоты сигнала?

4 В чём отличие метода доступа к среде передачи данных в проводных сетях *IEEE 802.3* и беспроводных сетях *IEEE 802.11*?

5 Какие числовые параметры характеризуют антенны для организации беспроводной связи стандарта *IEEE 802.11*? В каких диапазонах варьируются значения этих параметров?

8.6 Литература

1 Беспроводные сети Wi-Fi / А. В. Бобков [и др.]. – М. : ИНТУИТ, 2007. – 177 с.

2 Росс, Д. Wi-Fi. Беспроводная сеть / Д. Росс ; пер. с англ. В. А. Ветлужских. – М. : НТ Пресс, 2007. – 320 с.

3 Щербаков, А. К. Wi-Fi: Всё, что Вы хотели знать, но боялись спросить. Неофициальное пособие по глобальной системе местоопределения / А. К. Щербаков. – М. : Литературное агентство «Бук-Пресс», 2005. – 352 с.

Св. план 2015, поз. 4

Учебное издание

Шнейдеров Евгений Николаевич
Лихачевский Дмитрий Викторович
Лукашонок Дмитрий Викторович
Боровиков Сергей Максимович

КОМПЬЮТЕРНАЯ ТЕХНИКА, СИСТЕМЫ И СЕТИ. ЛАБОРАТОРНЫЙ ПРАКТИКУМ

ПОСОБИЕ

Редактор *Е. С. Чайковская*
Корректор *Е. И. Герман*
Компьютерная правка *Е. Н. Шнейдеров*
Оригинал-макет *Е. Д. Степусь*

Подписано в печать 02.05.2016. Формат 60×84 1/16. Бумага офсетная. Гарнитура «Таймс».
Отпечатано на ризографе. Усл. печ. л. 6,16. Уч.-изд. л. 6,5. Тираж 80 экз. Заказ 114.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники».
Свидетельство о государственной регистрации издателя, изготовителя, распространителя
печатных изданий №1/238 от 24.03.2014, №2/113 от 07.04.2014, №3/615 от 07.04.2014.
ЛП №02330/264 от 14.04.2014.
220013, Минск, П. Бровки, 6.