А. И. Новиков
Е. П. Дятлова

ОПЕРАЦИОННЫЕ СИСТЕМЫ, СЕТИ И ТЕЛЕКОММУНИКАЦИИ

Учебно-методическое пособие

Санкт-Петербург
2024

Министерство науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Санкт-Петербургский государственный университет промышленных технологий и дизайна»
Высшая школа технологии и энергетики

А. И. Новиков
Е. П. Дятлова

ОПЕРАЦИОННЫЕ СИСТЕМЫ, СЕТИ И ТЕЛЕКОММУНИКАЦИИ

Учебно-методическое пособие

Утверждено Редакционно-издательским советом ВШТЭ СПбГУПТД

Санкт-Петербург
2024

УДК 004.432
ББК 32.97
Н 731

Рецензенты:
кандидат технических наук, доцент, заведующий кафедрой прикладной математики и информатики Высшей школы технологии и энергетики Санкт-Петербургского государственного университета промышленных технологий и дизайна
И. В. Ремизова;
доктор технических наук, профессор, заведующий кафедрой автоматизации процессов химической промышленности Санкт-Петербургского государственного технологического института (Технического университета)
Л. А. Русинов

Новиков, А. И.
Н 731 Операционные системы, сети и телекоммуникации: учебно-методическое пособие / А. И. Новиков, Е. П. Дятлова. — СПб.: ВШТЭ СПбГУПТД, 2024. — 110 с.

Учебно-методическое пособие соответствует программам и учебным планам дисциплины «Операционные системы, сети и телекоммуникации» для студентов всех форм, обучающихся по направлению подготовки 09.03.03 «Прикладная информатика», а также может быть полезно при обучении по направлениям: 01.03.02 «Прикладная математика и информатика», 15.03.04 «Автоматизация технологических процессов и производств» и 27.03.04 «Управление в технических системах». В учебно-методическом пособии изложены основы работы вычислительных сетей. Приведены примеры настройки сети. Рассмотрены основы работы с операционными системами. Содержит разделы для самостоятельного углубленного изучения.

Пособие предназначено для подготовки бакалавров очной и заочной форм обучения. Отдельные разделы пособия могут быть полезны магистрантам и аспирантам.

УДК 004.432
ББК 32.97

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Наибольшее распространение в домашних и промышленных сетях на сегодняшний день получила технология Ethernet. Данная технология имеет несколько реализаций: непосредственно Ethernet, Fast Ethernet и Gigabit Ethernet – со скоростями 10, 100 и 1000 Мбит в секунду соответственно. Ethernet работает по коаксиальному кабелю, оптическому волокну и витой паре.

Витая пара представляет собой два провода, переплетенных вместе, что позволяет снизить уровень помех при передаче данных. Для обжима витой пары используются коннекторы RJ-45. Существуют разные способы установки коннектора RJ-45 на кабеле «витая пара».

Концентратор Hub (Хаб) и Коммутатор Switch оба являются центральными устройствами топологии «звезда» и имеют сходный внешний вид, но при этом Хаб работает только в режиме полудуплекса, а Switch может работать в режиме полного дуплекса. Данная возможность появляется благодаря использованию протоколов маршрутизации, таких как EIGRP, RIP и OSPF.

Пособие содержит разделы для самостоятельного углубленного изучения, названия которых отмечены звездочкой (*).

1. ОБЩИЕ СВЕДЕНИЯ О СЕТЯХ

1.1. Классификация сетей Ethernet

Наибольшее распространение в домашних и промышленных сетях на сегодняшний день получила технология **Ethernet**. Эта технология была разработана в 1970 г., а в 1980 г. на ее основе был создан стандарт **IEEE 802.3**. Данная технология имеет несколько реализаций: непосредственно Ethernet (рис. 1), Fast Ethernet (рис. 2), Gigabit Ethernet и др. – со скоростями 10, 100 и 1000 Мбит в секунду соответственно. Каждая последующая реализация является модернизацией предыдущей и работает на больших скоростях.

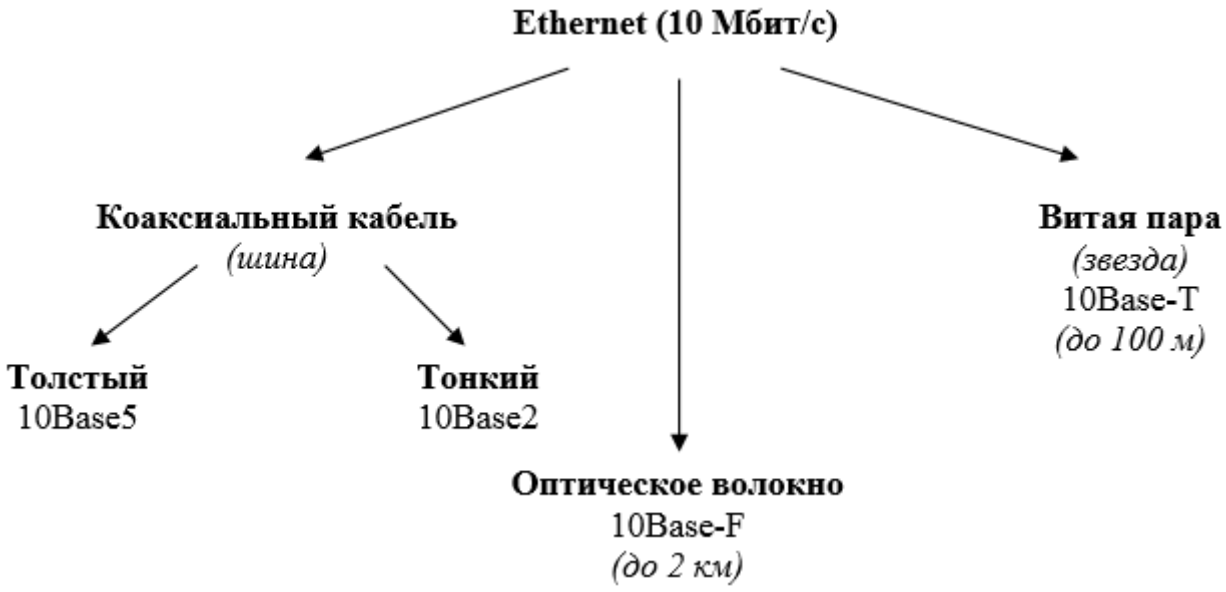


Рисунок 1 – Структура Ethernet 10 Мбит/с

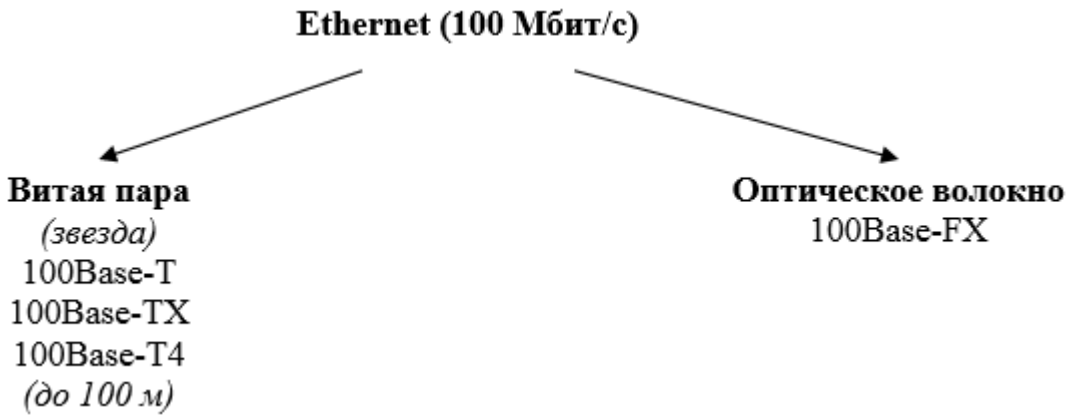


Рисунок 2 – Структура Ethernet 100 Мбит/с

Gigabit Ethernet для витой пары представлен стандартом 1000Base-T со скоростью передачи данных до 1000 Мбит/с (1 Гбит/с) и длиной одного сегмента до 100 м. Для построения сетей на основе Gigabit Ethernet должна быть использована витая пара не ниже категории **5** (Cat 5 или Cat 5e).

Существует и более быстрый стандарт 10GBase, работающий со скоростью 10 Гбит/с.

1.2. Обозначение стандартов сетей Ethernet

XBaseY,

где **X** – пропускная способность («скорость») сети (Мбит/с);

Y – максимальная длина сегмента (в сотнях метров),

или -**T** – twisted pair (витая пара),

или -**F** – fiber optic (оптическое волокно).

1.3. Коаксиальный кабель

Коаксиальный кабель (рис. 3) состоит из проводящей медной жилы и экрана, разделенных изолирующим слоем и помещенных в защитную оболочку. Коаксиальный кабель делится на толстый и тонкий в зависимости от диаметра [1]. На концах кабеля монтируются разъемы **BNC**.

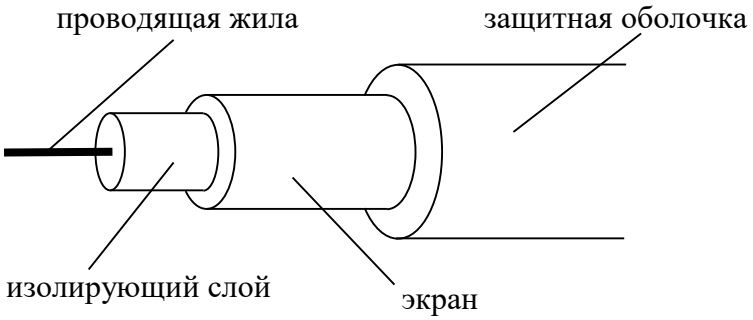


Рисунок 3 – Коаксиальный кабель

Сеть на базе коаксиального кабеля строится по типологии ШИНА (см. раздел **1.9**). Для соединения отрезков коаксиальных кабелей используются Т-коннекторы (рис. 4).

Рисунок 4 – Т-коннектор

На свободные концы коаксиального кабеля **ОБЯЗАТЕЛЬНО** устанавливаются терминаторы (рис. 5), представляющие собой гасители сигнала сопротивлением 50 Ом.

Рисунок 5 – Терминатор

1.4. Витая пара

Витая пара – это два провода, переплетенных вместе, что позволяет снизить уровень помех при передаче данных. Кабель Ethernet, который называется кабель «витая пара», представляет собой четыре (реже две) витые пары в одной оплетке. Витая пара может быть экранированной (обозначается STP) и неэкранированной (обозначается UTP). Для обжима витой пары используются коннекторы **RJ-45** (рис. 6), он же **8P8C** *(8 Position 8 Contact)*.

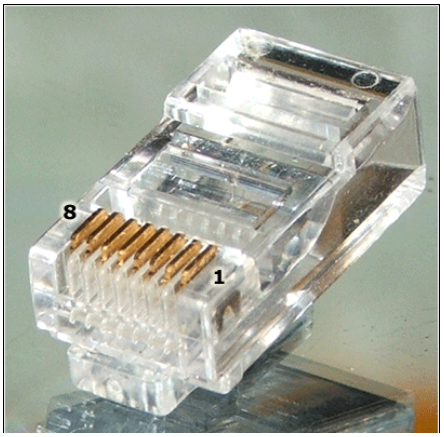


Рисунок 6 – Коннектор RJ-45

1.5. Правила обжима витой пары

Существует два основных способа установки коннектора **RJ-45** на кабеле «витая пара» [1]: EIA/TIA-568**A** (рис. 7а) и EIA/TIA-568**B** (рис. 7б).

	1	зелено-белый
	2	зеленый
	3	оранжево-белый
	4	синий
	5	сине-белый
	6	оранжевый
	7	коричнево-белый
	8	коричневый

а) EIA/TIA-568A

	1	оранжево-белый
	2	оранжевый
	3	зелено-белый
	4	синий
	5	сине-белый
	6	зеленый
	7	коричнево-белый
	8	коричневый

б) EIA/TIA-568B

Рисунок 7 – Варианты обжима витой пары

Кроме этого, правила обжима кабеля зависят от того, какие устройства необходимо соединить:

1) Компьютер – Hub

В этом случае на противоположных концах кабеля используются одинаковые способы обжима (рис. 8), либо EIA/TIA-568**A** и EIA/TIA-568**A**, либо EIA/TIA-568**B** и EIA/TIA-568**B**.

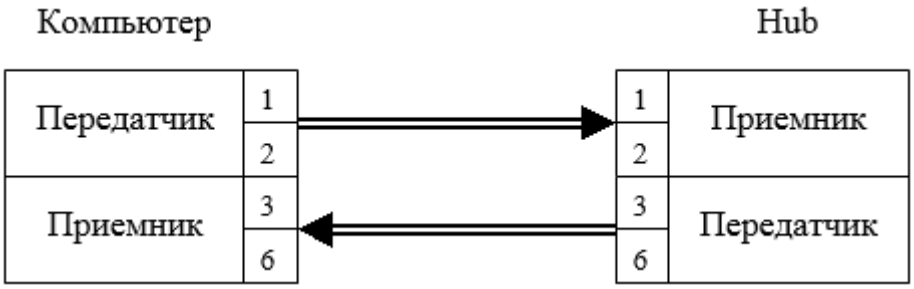


Рисунок 8 – Соединение Компьютер – Hub

2) Компьютер – Компьютер

В этом случае на противоположных концах кабеля используются различные способы обжима (рис. 9), т. е. и EIA/TIA-568**A** и EIA/TIA-568**B**, а кабель называется **перекрестным** (**кроссовым**).

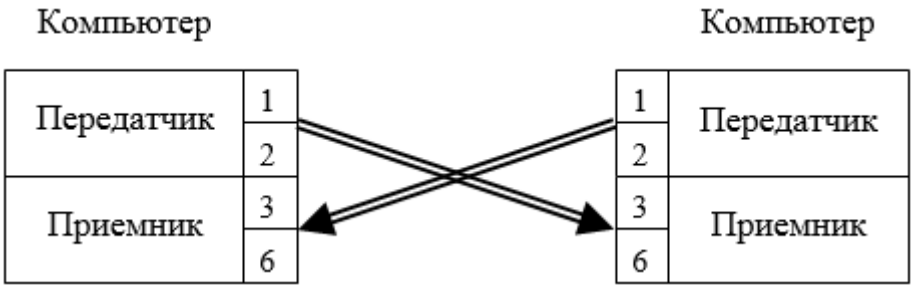


Рисунок 9 – Соединение Компьютер – Компьютер

1.6. *Технология PoE

PoE (**Power over Ethernet**, «Передача электроэнергии через Ethernet») – технология, позволяющая передавать удаленному устройству электрическую энергию вместе с данными через стандартную витую пару. Подобные технологии предназначаются для IP-камер, IP-телефонии, точек доступа беспроводных сетей, сетевых концентраторов и других устройств, к которым нежелательно или невозможно проводить отдельный электрический кабель.

В простейшем случае для сети 10/100Base-T, использующей только две из четырех пар проводов, питание передается по свободным проводникам (рис. 10).

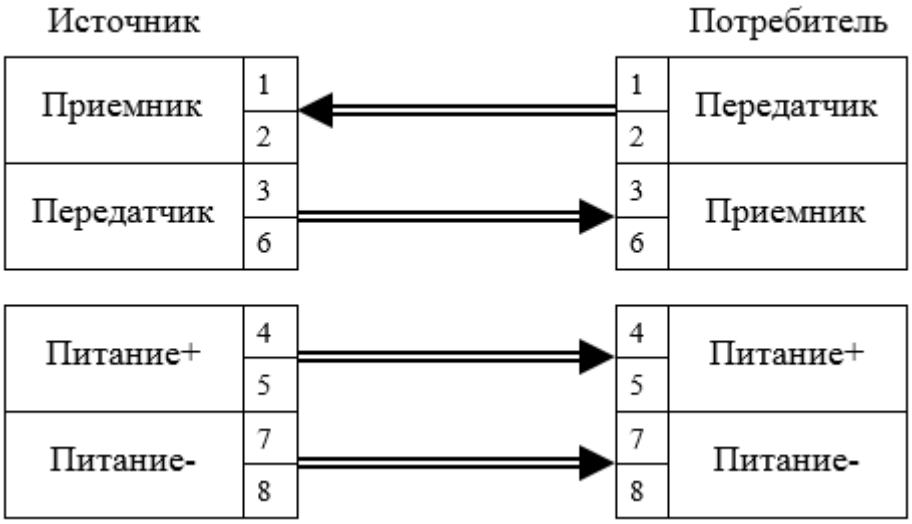


Рисунок 10 – Питание по свободным жилам

В более сложных случаях питание передается по тем же жилам, что и сигнал (данные). При этом данные могут передаваться по всем четырем парам проводников, что позволяет использовать этот способ не только для Ethernet 10/100 Мбит/с, но и для Gigabit Ethernet.

1.7. *Оптическое волокно

Оптическое волокно является более дорогим по сравнению с витой парой и коаксиальным кабелем, а также требует определенной квалификации и инструмента для его монтажа. Проходя по оптическому кабелю, свет многократно отражается (рис. 11), при этом потери сигнала малы.

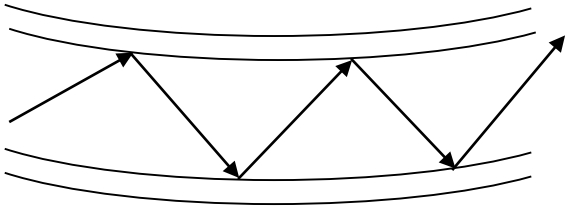


Рисунок 11 – Оптическое волокно

Как правило, используется в оптических магистралях для передачи больших объемов данных на дальние расстояния. Поэтому в большинстве случаев в оптоволоконной сети имеет место соединение «точка-точка», когда по оптоволокну соединяются лишь два устройства, а дальше данные расходятся к своим адресатам по витой паре (рис. 12). Подобная топология, так же как и для сети на основании чисто витой пары, называется «звезда».

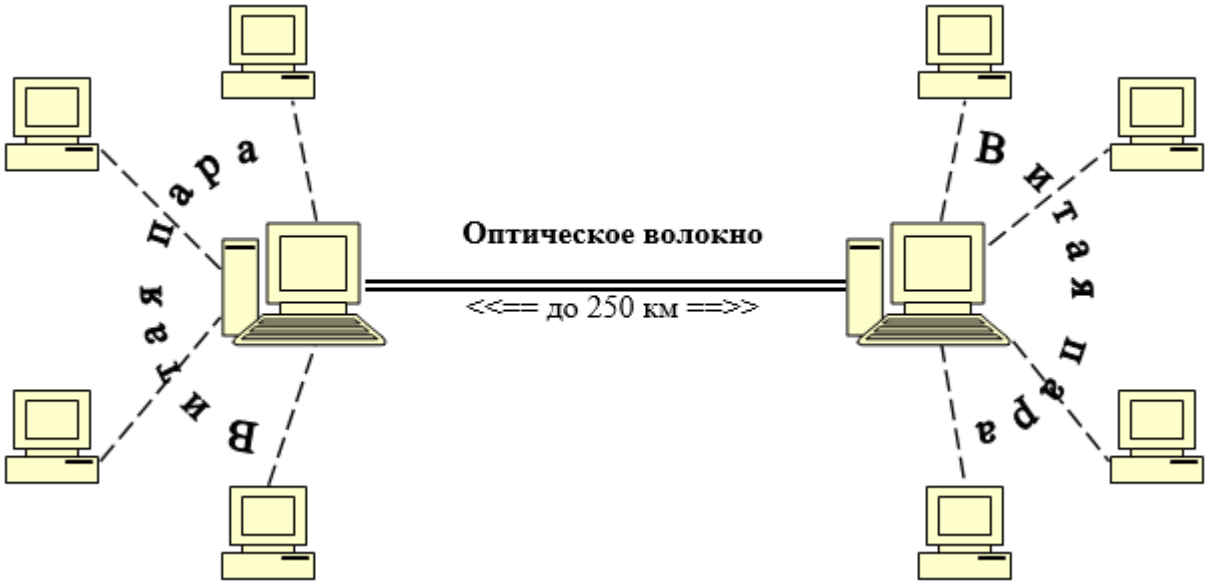


Рисунок 12 – Оптоволоконная магистраль

Существуют различные оптические разъемы (рис. 13), имеющие различные диаметры наконечников и способы крепления, такие как FC, SC, ST, LC.

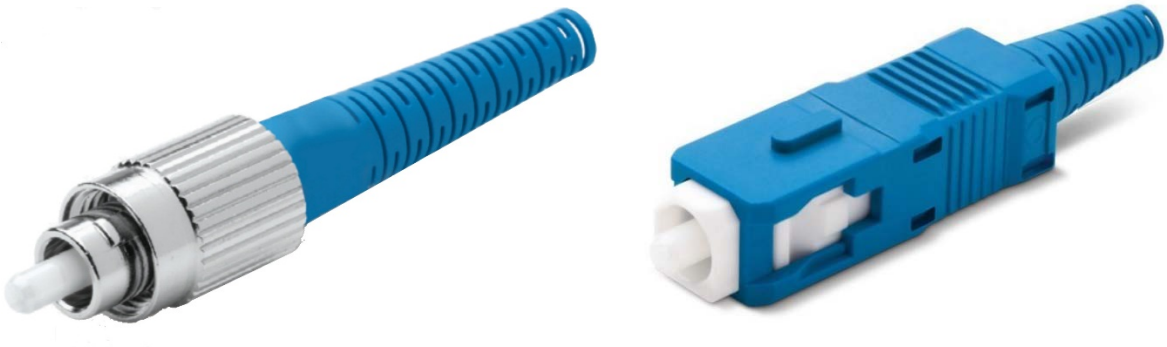


Рисунок 13 – Оптические разъемы

1.8. *Модули SFP

SFP (Small Form-factor Pluggable) – промышленный стандарт модульных компактных приемопередатчиков (трансиверов), используемых для передачи и приема данных в телекоммуникациях.

Модули **SFP** используются для присоединения платы сетевого устройства (коммутатора, маршрутизатора или подобного устройства) к оптическому волокну (рис. 14) или витой паре (рис. 15), выступающими в роли сетевого кабеля.

При этом устройства, оснащенные разъемами **SFP** (или **SFP+**), позволяют легко устанавливать различные модули в одно и то же гнездо (рис. 16).

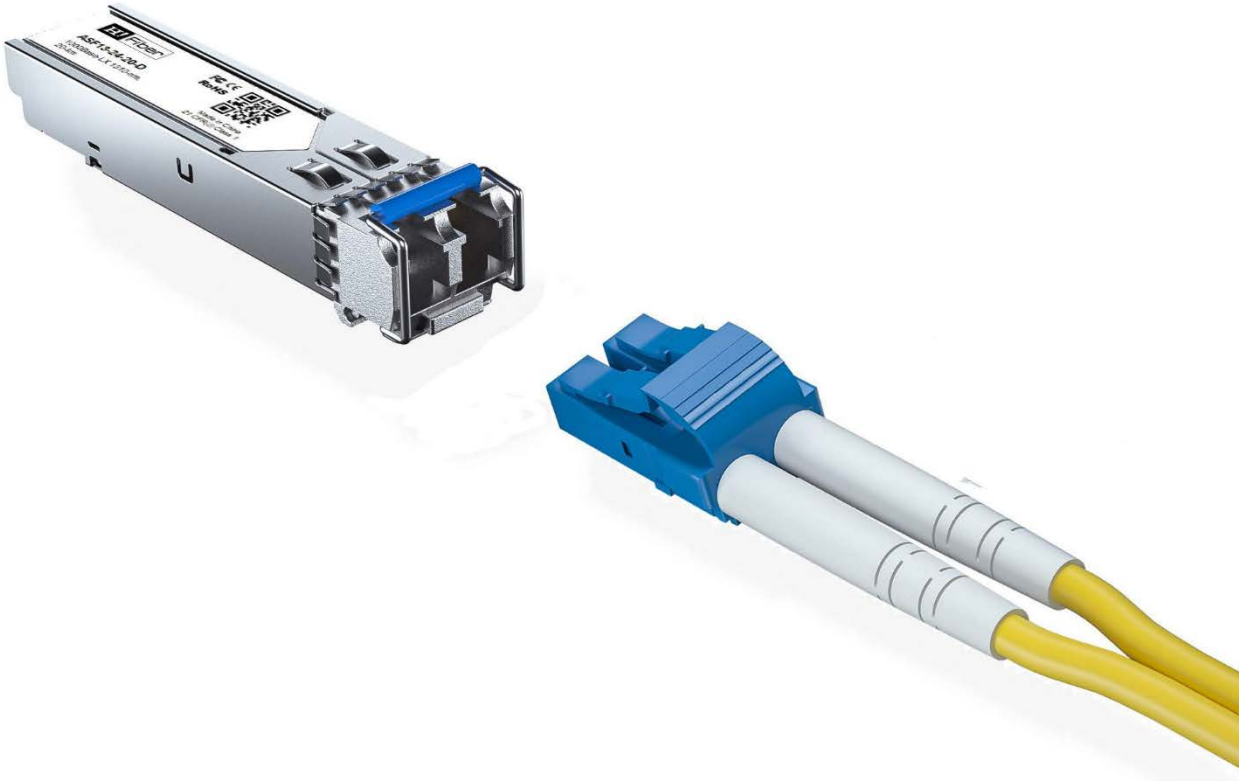


Рисунок 14 – Модуль с оптическим разъемом

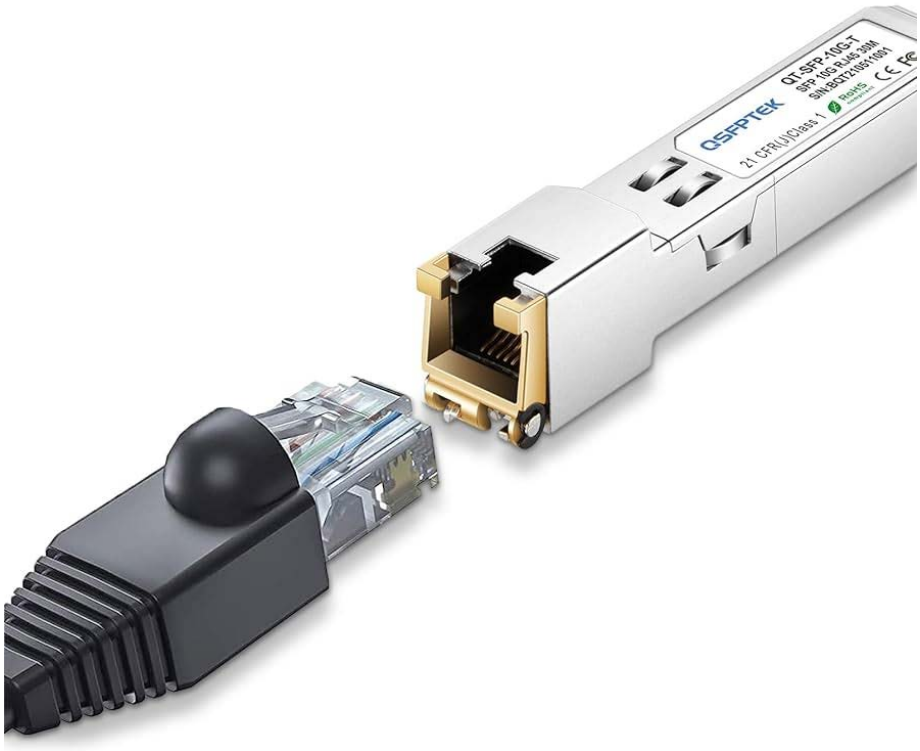


Рисунок 15 – Модуль с разъемом RJ-45

Рисунок 16 – Установка модуля

1.9. Топологии сетей

Звезда

В настоящее время «звезда» (рис. 17) является наиболее часто встречающейся топологией для массового непромышленного использования. Связано это в основном с распространением Ethernet.

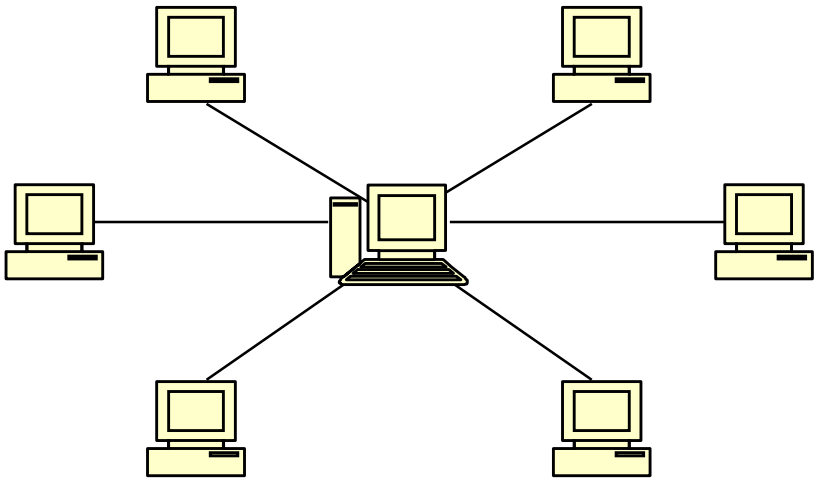


Рисунок 17 – Топология «звезда»

Компьютеры соединяются через центральное устройство. При выходе из строя одного из компьютеров сеть продолжает работать, но при выходе из строя центрального устройства связь между компьютерами пропадает.

Шина

Топология «шина» (рис. 18) используется, например, для построения сетей Ethernet 10Base5 и 10Base2, которые в настоящее время считаются устаревшими. Подсоединение компьютеров осуществляется в разрыв кабеля с помощью Т-коннектора. При отсоединении компьютера Т-коннектор удаляется, а на его место устанавливается перемычка, соединяющая два сегмента разорванного кабеля. На концах шины **ОБЯЗАТЕЛЬНО** устанавливаются терминаторы.

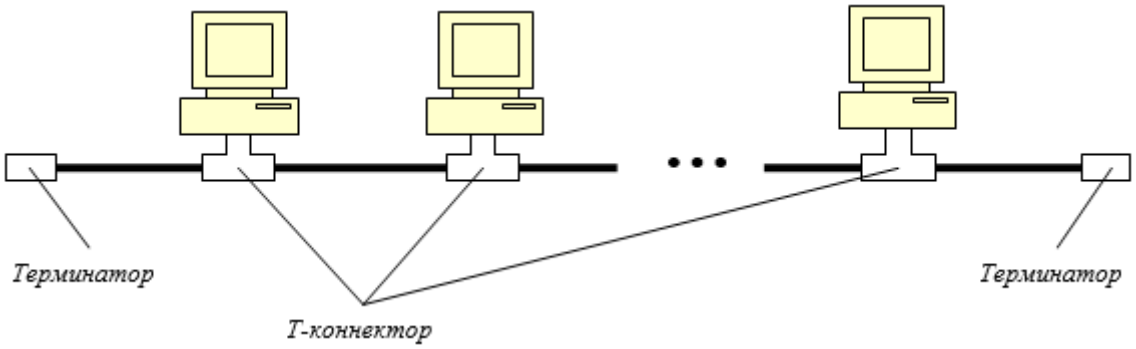


Рисунок 18 – Топология «шина»

Кольцо

«Кольцо» используется в сетях, отличных от Ethernet, например Token Ring, где по сети постоянно циркулирует короткий блок данных (маркер), получаемый каждой рабочей станцией по очереди.

Можно выделить два подтипа данной топологии: одностороннее кольцо (рис. 19) и двухстороннее кольцо (рис. 20).

В одностороннем кольце при выходе из строя одного из сегментов кабеля, соединяющего рабочие станции, выходит из строя вся сеть. Эта проблема решается передачей сообщений в обоих направлениях (см. рис. 20), и при выходе из строя одного из сегментов кабеля, соединяющего рабочие станции, сообщения могут передаваться в обратном направлении по второму кольцу. Двухстороннее кольцо является более надежным, так как даже при выходе из строя одновременно нескольких элементов сети, отдельные ее части продолжают функционировать. Но двойное кольцо является более дорогостоящим по сравнению с односторонним из-за большего количества используемого кабеля и более сложной реализации.

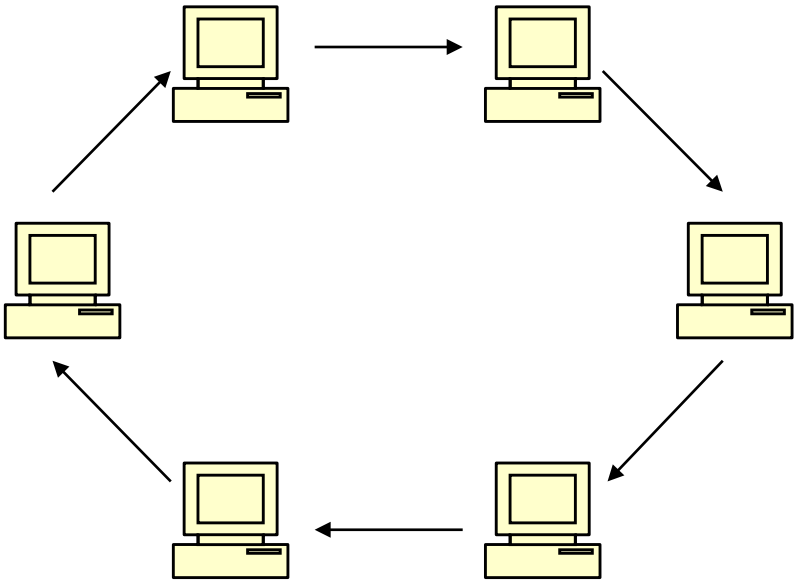


Рисунок 19 – Одностороннее кольцо

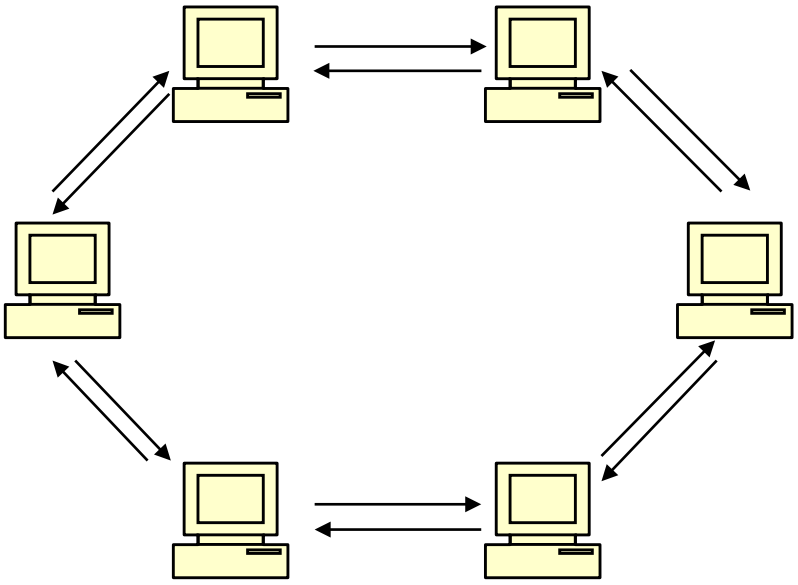


Рисунок 20 – Двухстороннее кольцо

Смешанная топология

Сеть может быть смешанной, т. е. содержать в себе признаки различных топологий (рис. 21). В некоторых литературных источниках подобную структуру сети выделяют в отдельную (четвертую) топологию.

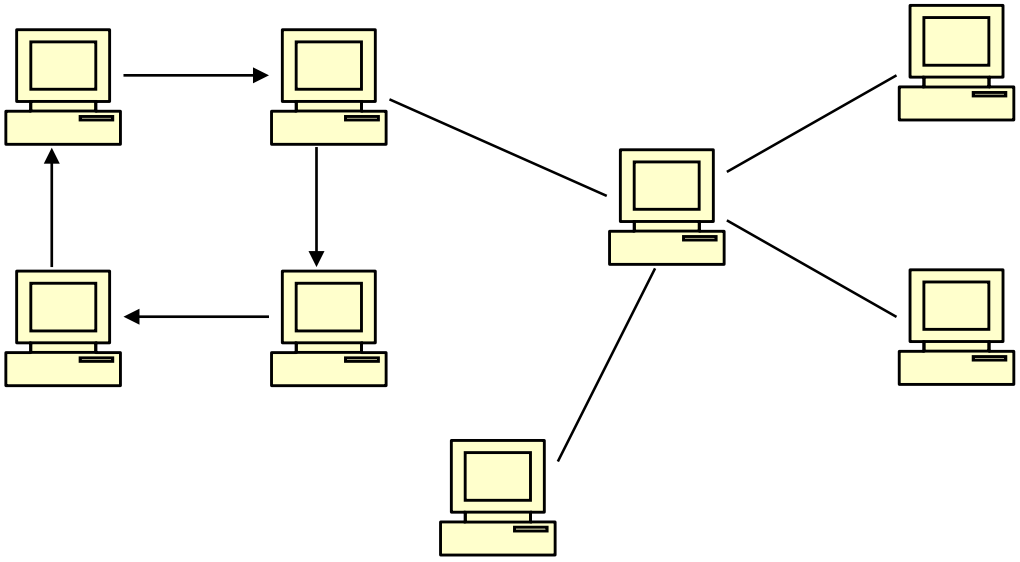


Рисунок 21 – Пример смешанной топологии

При этом узел, принадлежащий одновременно двум или более сетям, является шлюзом либо мостом.

Мост (**bridge**) – устройство, соединяющее две отдельные сети или два сегмента одной сети, использующих одинаковые протоколы.

Шлюз (**gateway**) – более сложное устройство, объединяющее разнородные сети.

1.10. Концентратор Hub

Hub («**Хаб**», «**Концентратор**») – является центральным звеном топологии «звезда» [1]. Имеет восемь разъемов **RJ-45** (или большее их число, как правило, кратное восьми). Особенностью работы хаба является то, что, получив сообщение от одного компьютера, он передает его на все компьютеры (рис. 22). Таким образом, при передаче сообщения одним из компьютеров другие передавать не могут. Хаб работает только в режиме полудуплекса (см. раздел **2.1**).

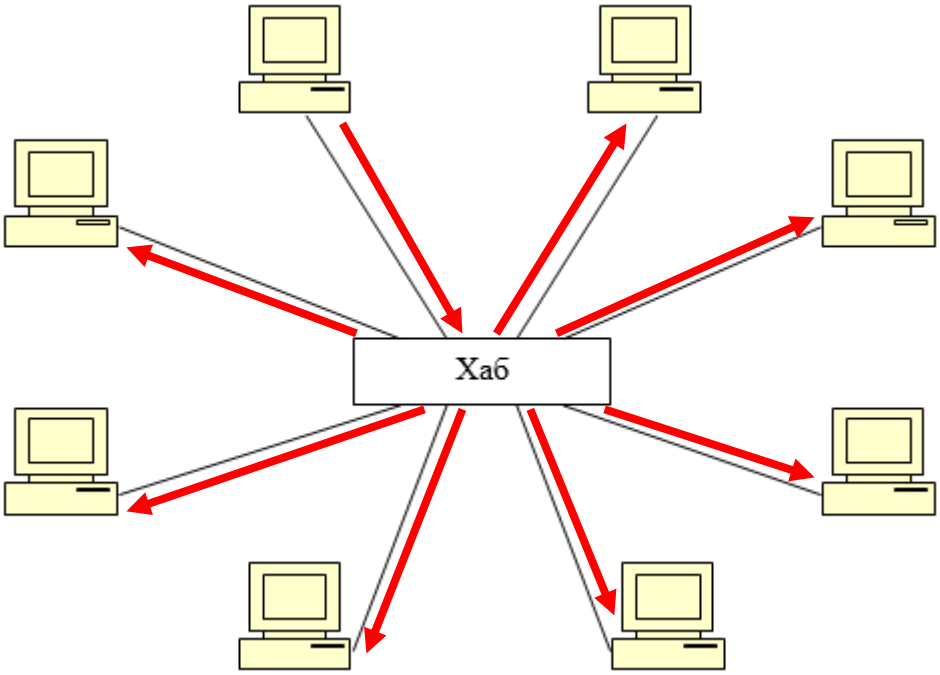


Рисунок 22 – Концентратор Hub

1.11. Коммутатор Switch

Switch («**Коммутатор**») – аналогичен Хабу, за исключением того, что сообщение посылается на конкретный компьютер. При этом другие также могут передавать (рис. 23). Switch может работать в режиме полного дуплекса (см. раздел **2.1**).

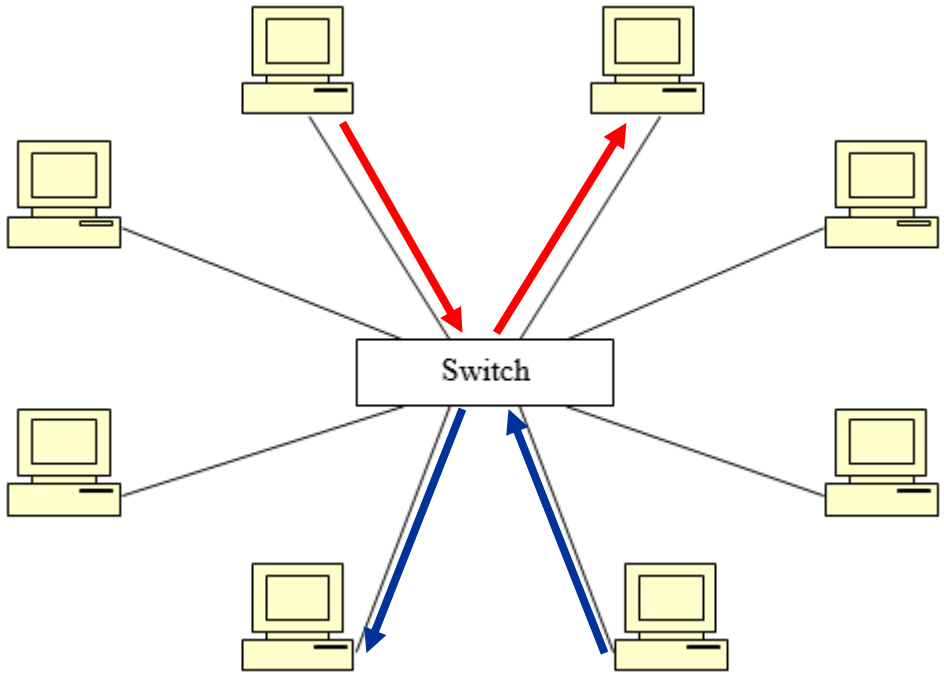


Рисунок 23 – Коммутатор Switch

1.12. Репитеры

Репитер («**Повторитель**») – используются для увеличения длины сети (выступают в роли усилителя сигнала). Длинна сегмента витой пары не может превышать 100 м, но репитеры позволяют избавиться от этого ограничения.

Репитер – это не обязательно отдельное устройство. Концентраторы и коммутаторы также выполняют эту функцию.

1.13. ЛАБОРАТОРНАЯ РАБОТА № 1
Настройка сети в ОС Windows

Настройка имени компьютера и рабочей группы

Имена компьютера и рабочей группы задаются в свойствах системы. Для вызова окна свойств системы необходимо щелкнуть правой клавишей мыши по значку «**Мой компьютер**» на рабочем столе и в появившемся меню выбрать пункт «**Свойства**» (рис. 24).

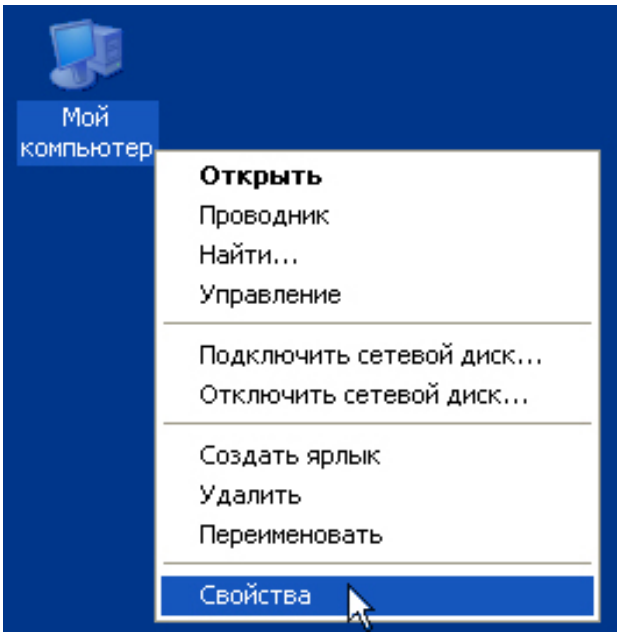


Рисунок 24 – Контекстное меню «Мой компьютер»

В окне свойств системы (рис. 25) необходимо выбрать вкладку «**Имя компьютера**» и далее нажать кнопку «**Изменить**».

Имена компьютеров в сети должны быть уникальными, а рабочая группа одинаковой.

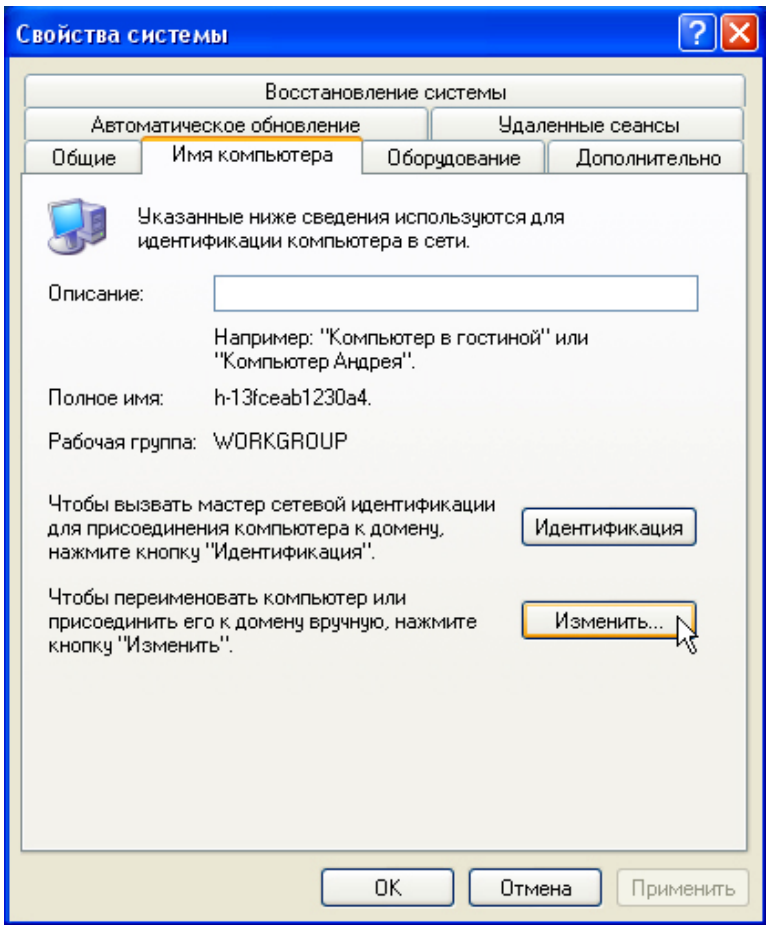


Рисунок 25 – Свойства системы

Настройка IP-адреса и маски подсети

Для ввода IP-адреса и маски подсети необходимо:

а) Щелкнуть правой клавишей мыши по значку «**Сетевое окружение**» и выбрать пункт меню «**Свойства**» (рис. 26).

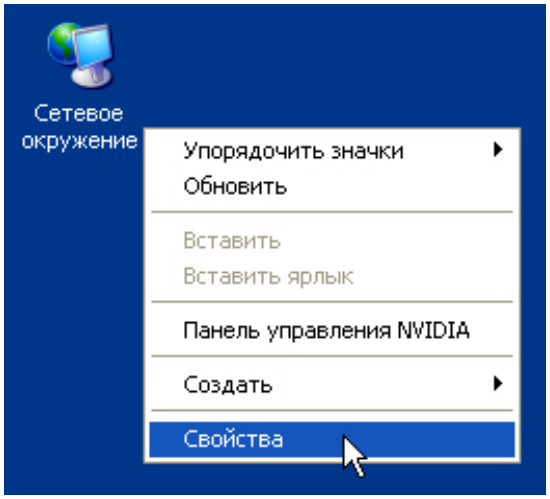


Рисунок 26 – Контекстное меню сетевого окружения

б) В некоторых версиях Windows после этого необходимо выбрать еще пункт «Изменение параметров адаптера» (рис. 27).

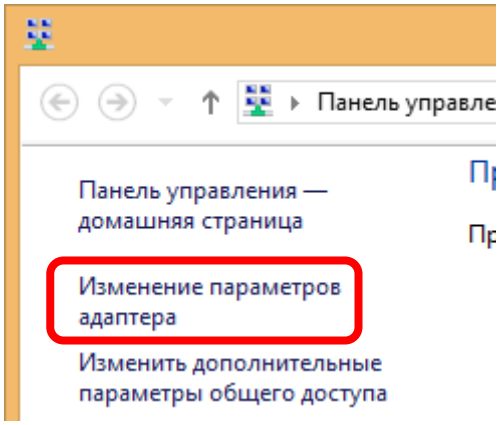


Рисунок 27 – Изменение параметров адаптера

в) В открывшемся окне «Сетевые подключения» (рис. 28) будет отображен список всех сетевых устройств, установленных на компьютере.

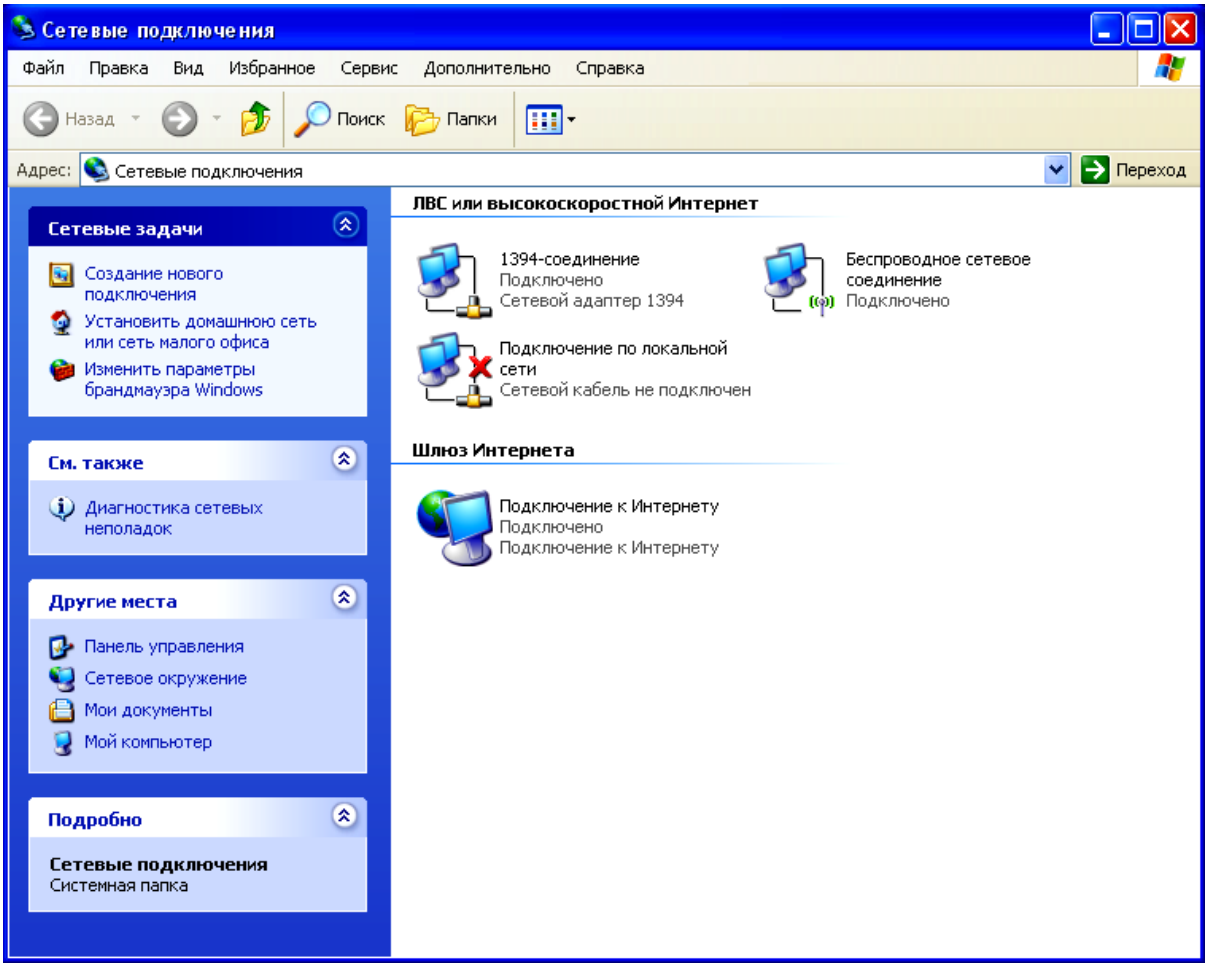


Рисунок 28 – Сетевые подключения

Текущее состояние каждого из подключений можно определить по внешнему виду его ярлыка, а также по надписи под названием подключения.

Подключения могут иметь три основных состояния:

 – Подключено (цветной значок в виде двух компьютеров);

 – Отключено (черно-белый значок в виде двух компьютеров);

 – Сетевой кабель не подключен (Сеть не найдена)
(цветной значок в виде двух компьютеров с красным крестом).

Переход из состояния «Отключено» в состояние «Подключено»/«Сеть не найдена» осуществляется выбором пункта «Подключить» контекстного меню для соответствующего соединения. Вызов меню осуществляется щелчком правой клавиши мыши по значку соответствующего соединения.

Переход из состояния «Подключено»/«Сеть не найдена» в состояние «Отключено» осуществляется выбором пункта меню «Отключить».

Переход между состояниями «Подключено» и «Сеть не найдена» осуществляется автоматически в зависимости от того, установлено ли физическое соединение (подключен ли кабель).

г) Выбрать из контекстного меню требуемого соединения пункт «Свойства». В появившемся окне (рис. 29) установить галочку «Вывести значок в области уведомления» («Отображать состояние подключения»).

Из списка протоколов выбрать протокол TCP/IP, дважды кликнув по нему мышкой.

д) В появившемся окне (рис. 30) вручную задать IP-адрес и маску подсети.

IP-адреса компьютеров в сети должны быть уникальным, а маска подсети одинаковой. Например, как представлено в табл. 1.

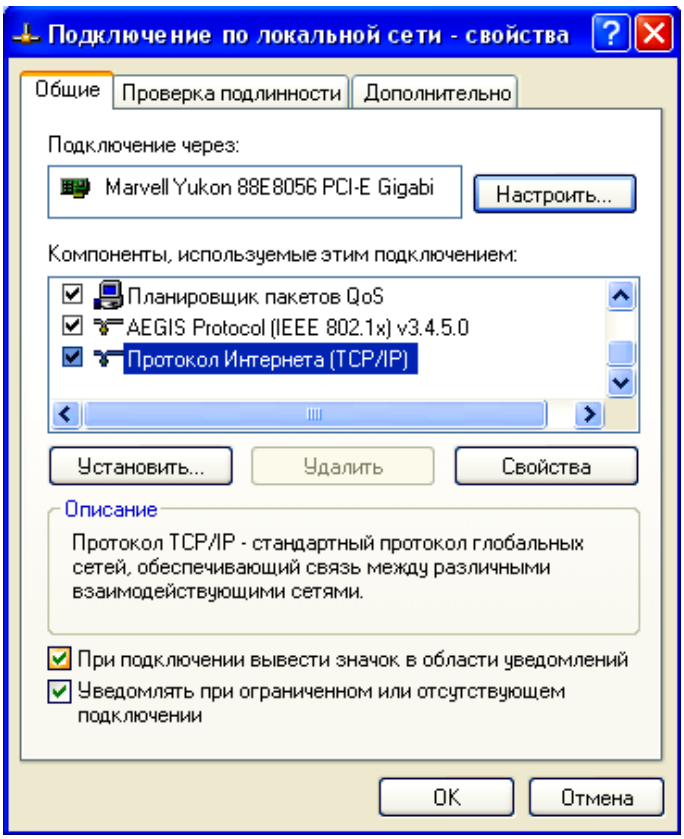


Рисунок 29 – Свойства подключения

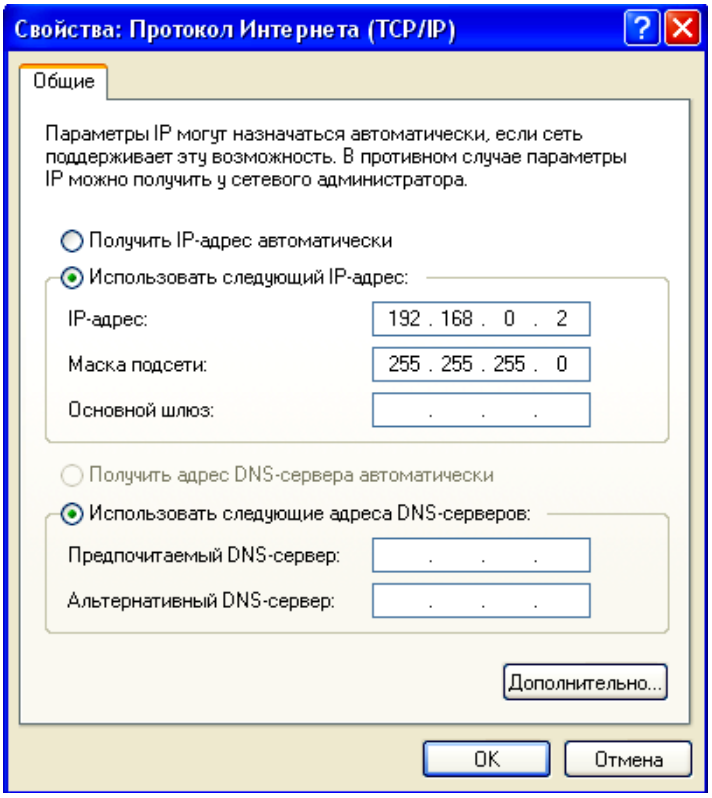


Рисунок 30 – Свойства протокола TCP/IP

Таблица 1 – Настройки сети

Имя компьютера	Comp1	Comp2	Comp3	...
Рабочая группа	WORKGROUP			
IP	192.168.0.1	192.168.0.2	192.168.0.3	...
Маска подсети	255.255.255.0			

Определение IP-адреса и MAC-адреса

Для определения текущего IP-адреса (в т. ч. когда он настроен на автоматическое получение по DHCP), необходимо открыть окно (рис. 31) «Сведения о сетевом подключении» («Детали сетевого подключения»). В этом же окне будет отображен и MAC-адрес («Физический адрес»).

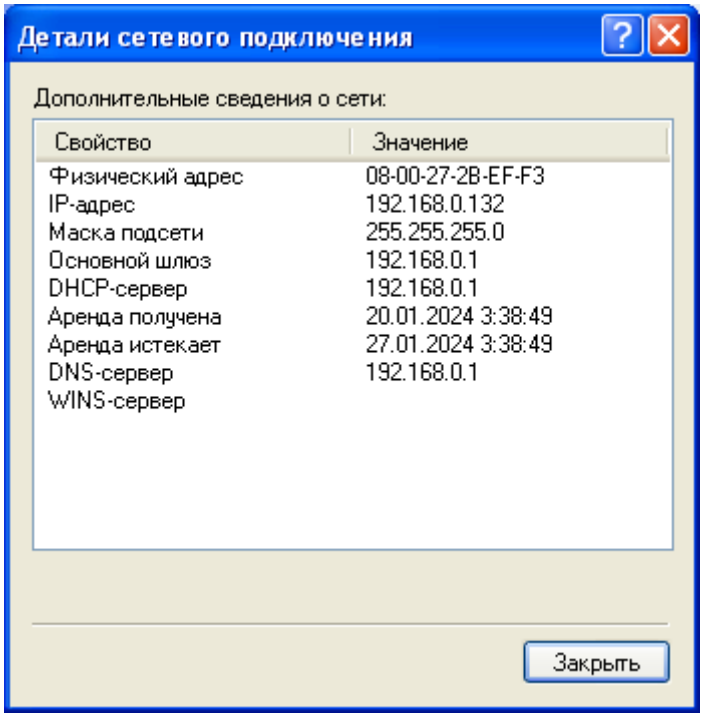


Рисунок 31 – IP-адрес и MAC-адрес

Добраться до данного окна можно через:

- значок сетевого подключения в «трее» (у часов);
- или через Свойства сетевого окружения.

Необходимо вызвать окно «Состояние» сетевого подключения (рис. 32–33), в котором требуется нажать кнопку «Сведения» («Подробности»).

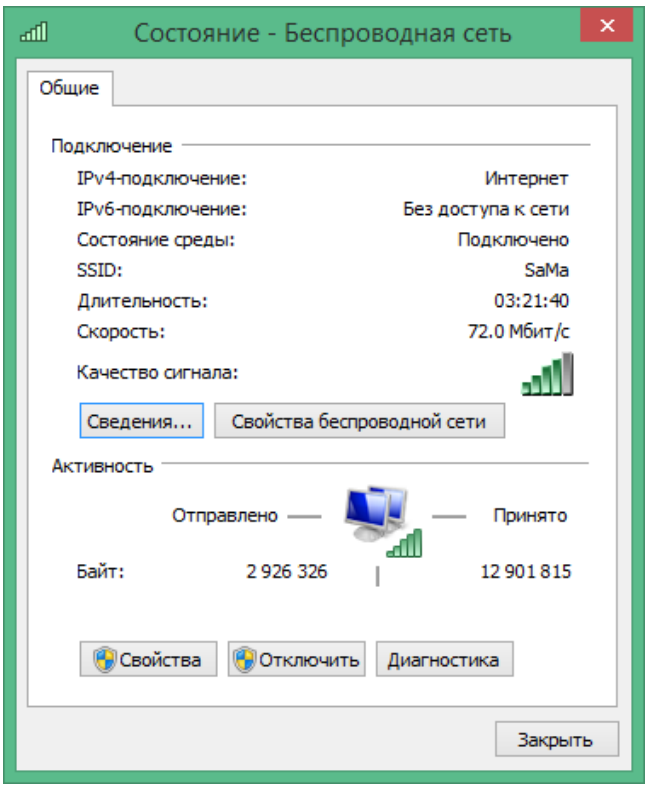


Рисунок 32 – Состояние сетевого подключения (в Windows 8.1)

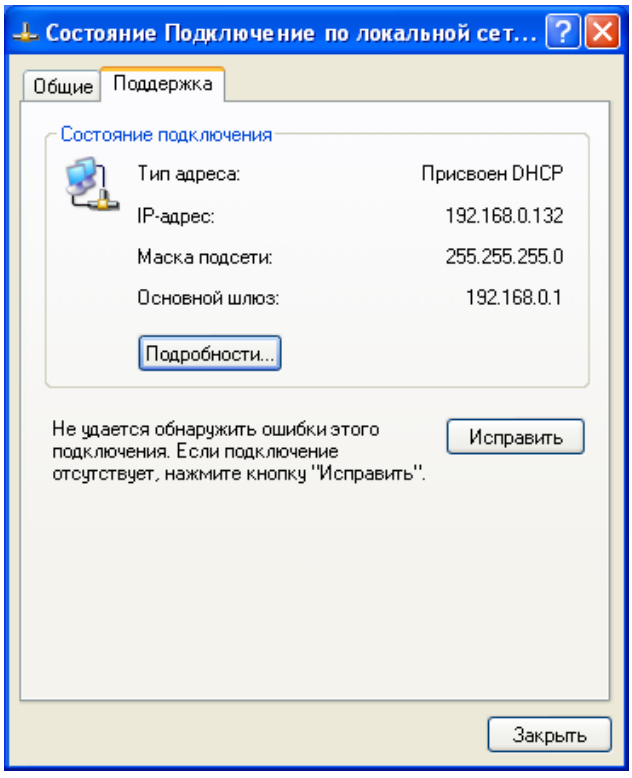


Рисунок 33 – Состояние сетевого подключения (в Windows XP)

***Настройка подключения к сети Интернет**

Для подключения к сети **Internet** необходимо также указать **шлюз** и **DNS-сервер** (рис. 34). Как правило, в малых сетях адреса **шлюза** и **DNS-сервера** совпадают, что свидетельствует о совмещении обеих функций в одном устройстве (роутере). По умолчанию это адрес **192.168.0.1**.

За более подробной информацией о настройках подключения к сети Интернет обращайтесь к вашему провайдеру.

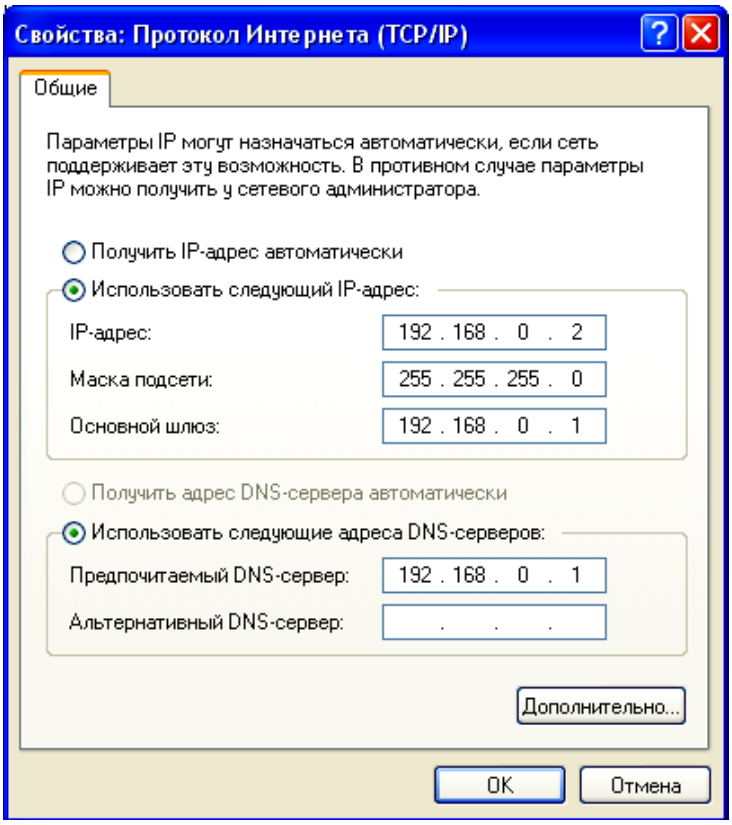


Рисунок 34 – Указание шлюза и DNS- сервера

1.14. Команда PING

Для проверки наличия в сети узла предназначена команда **PING**. Чтобы воспользоваться данной командой, необходимо выбрать пункт «**Выполнить**» из меню «Пуск» или нажать комбинацию клавиш «**Win+R**». После чего ввести необходимую команду, например (рис. 35):

PING 192.168.0.1

или

PING rambler.ru

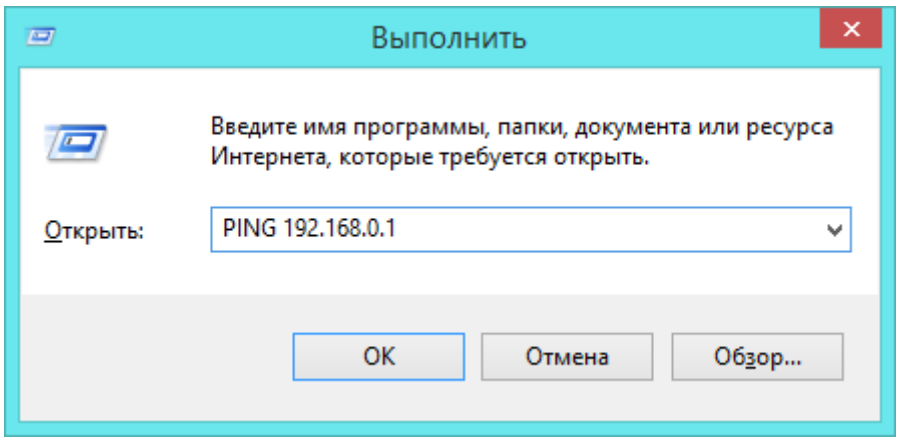


Рисунок 35 – Запуск команды Ping

В результате работы команды Ping будет отображена следующая информация (рис. 36).

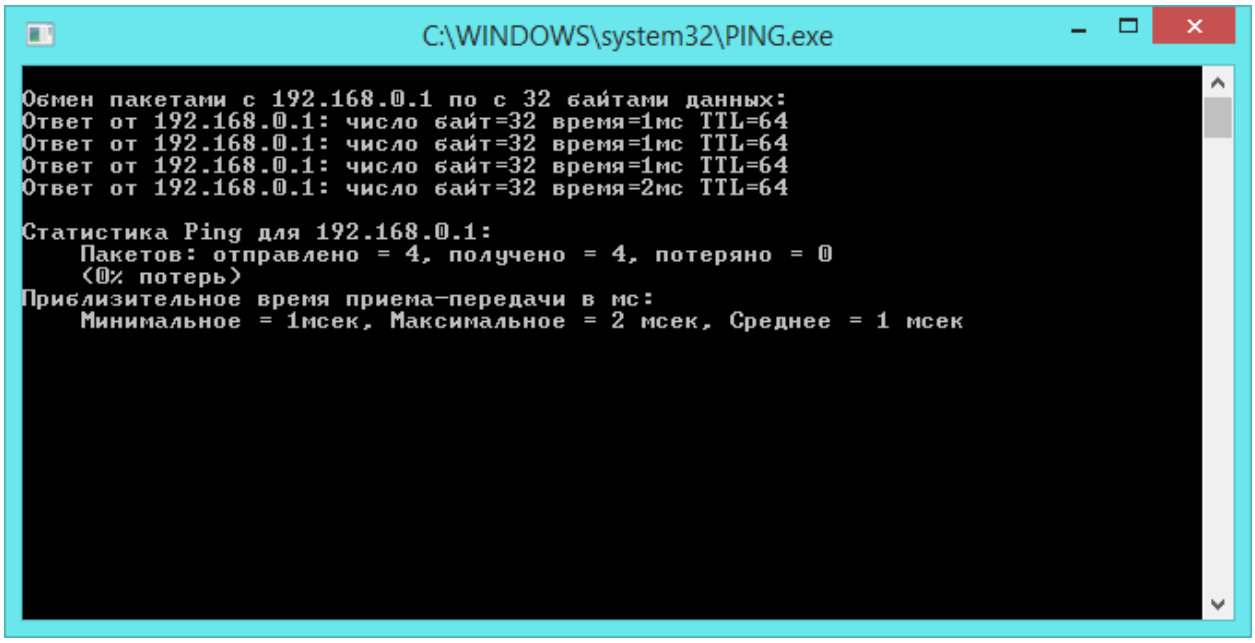


Рисунок 36 – Результат работы команды Ping

1.15. Общий доступ к папке

Общий доступ нельзя предоставить для отдельного файла. Файл, передаваемый по сети, вначале нужно положить в созданную для этого папку. В папке должен быть хотя бы один файл.

Также применяется термин «**Расшарить папку**» (от англ. «share» – «делиться»), означающий «Открыть доступ к папке».

Для предоставления общего доступа к папке необходимо зайти в ее «Свойства», после чего на вкладке «**Доступ**» (рис. 37) установить соответствующие галочки.

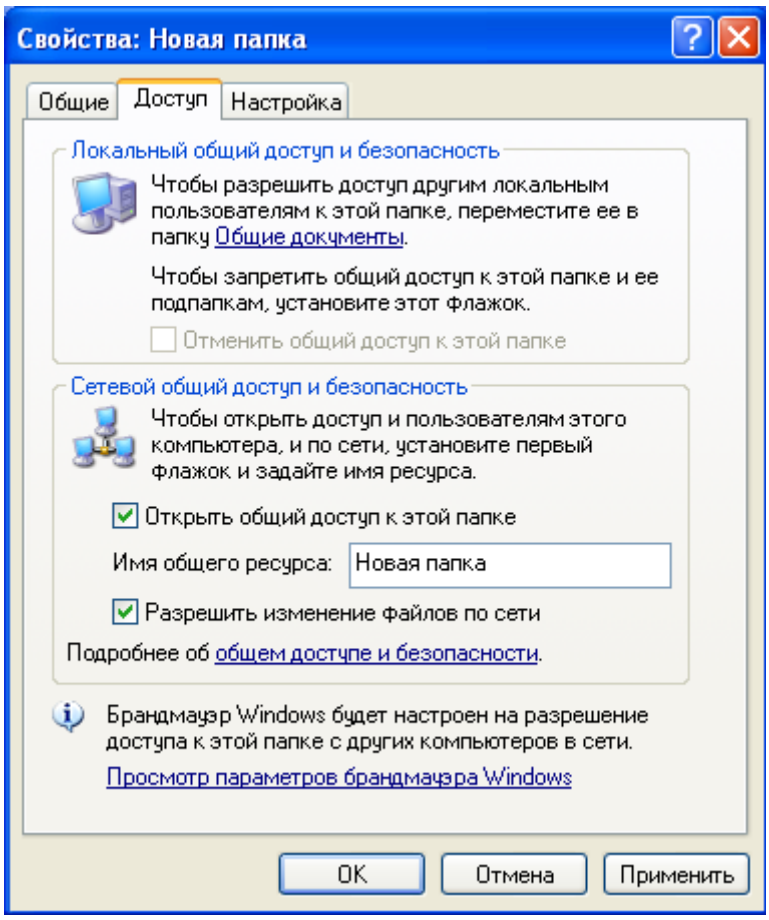


Рисунок 37 – Настройка общего доступа

После настройки иконка папки, к которой предоставлен общий доступ, изменится (рис. 38).

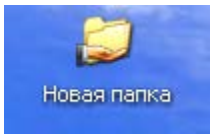


Рисунок 38 – Иконка папка с общим доступом

Для того чтобы открыть папку с другого устройства, необходимо в проводнике ввести имя компьютера (которое мы настроили ранее в разделе **1.13**), начинающийся с «\\» (рис. 39). Аналогично можно получить доступ по IP-адресу (рис. 40).

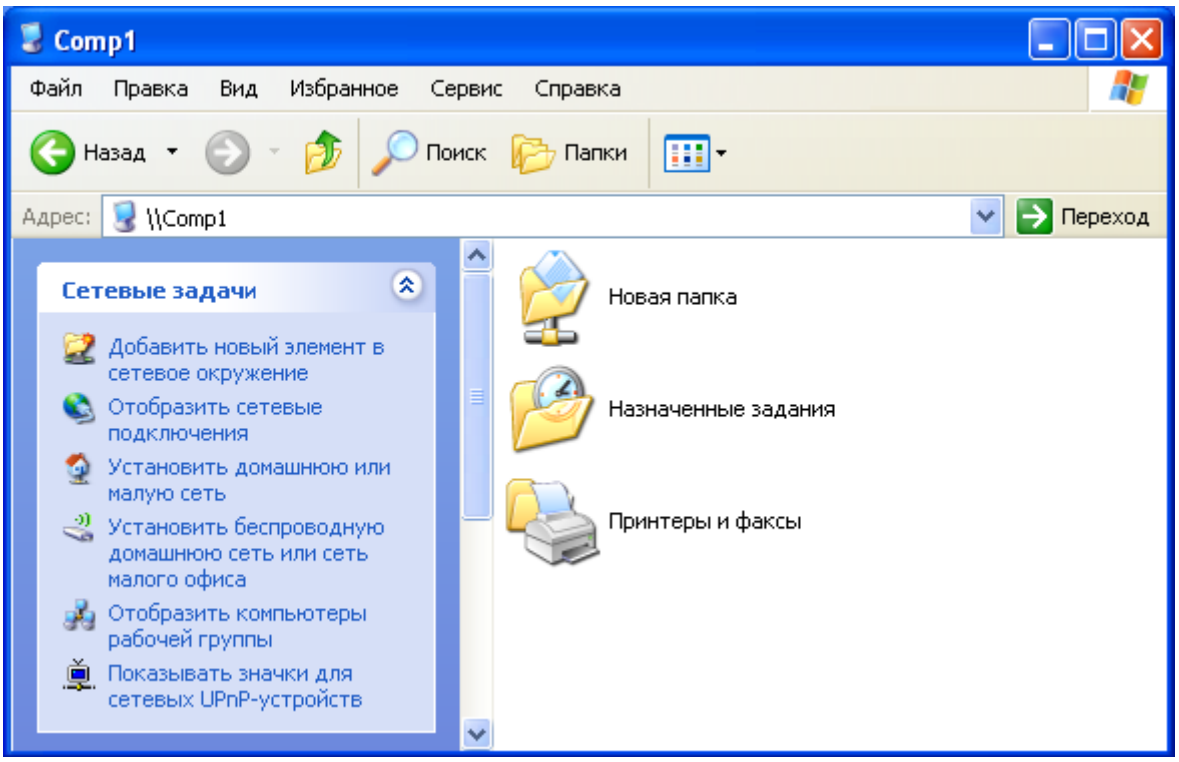


Рисунок 39 – Доступ к папке

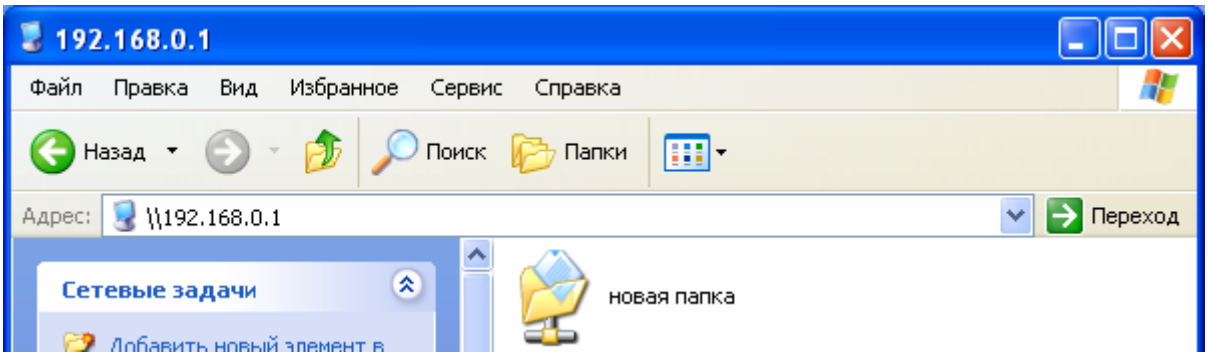


Рисунок 40 – Доступ к папке по IP-адресу

1.16. *Протокол SMB

Ранее (в разделе **1.15**) была рассмотрена Локальная сеть Windows. Данный протокол называется **SMB** (**Samba**, **CIFS**). Кроме доступа к файлам, протокол SMB позволяет получить доступ к сетевым принтерам.

SMB (Server Message Block) – сетевой протокол прикладного уровня для удаленного доступа к файлам, принтерам и другим сетевым ресурсам, а также для межпроцессного взаимодействия. Известен также под названием **CIFS** (Common Internet File System, «Единая файловая система Интернета»).

Samba – пакет программ, которые позволяют обращаться к сетевым дискам и принтерам на различных операционных системах по протоколу SMB/CIFS. Является свободным программным обеспечением, выпущен под

лицензией GPL. Работает на macOS, Linux и других Unix-подобных операционных системах.

Аналогично Локальной сети Windows для других операционных систем также существуют «родные» протоколы для доступа к файлам, такие как **NFS** (Network File System) для Unix, или **AFP** (Apple Filing Protocol) для Mac OS.

1.17. *NAS

NAS (Network Attached Storage, «Сетевое хранилище») – является сервером для хранения данных на файловом уровне. По сути представляет собой компьютер с некоторым дисковым массивом, подключенный к сети.

NAS-узел – представляет собой отдельный компьютер или специализированное устройство (рис. 41), основным предназначением которого является предоставление служб для хранения данных другим устройствам в сети. Обычно системы NAS содержат несколько жестких дисков, которые объединены в **RAID**-массивы с возможностью восстановления данных при сбое. Например, можно создать «зеркальный массив» **RAID 1**, который пишет одни и те же данные сразу на два разных диска.

Рисунок 41 – Сетевое хранилище Qnap D4

Операционная система и программы NAS обеспечивают работу хранилища данных и файловой системы, доступ к файлам, а также контроль над

функциями системы. Устройство не предназначено для выполнения обычных вычислительных задач, хотя запуск других программ на нем и возможен с технической точки зрения. Зачастую NAS-системы имеют скудный графический или консольный интерфейс (или не имеют его вовсе), а все настройки производятся через web-интерфейс.

Обычно такие хранилища позволяют работать сразу с множеством высокоуровневых прикладных протоколов, таких как SMB/CIFS, FTP, SFTP, HTTP/HTTPS, AFP, NFS, WebDAV, BitTorrent, DLNA и др. А также позволяет организовывать резервное копирование файлов или собственное «облачное хранилище» и запускать СУБД (MySQL, MariaDB).

В NAS необходимо использовать только специализированные жесткие диски, такие как WD Red или Seagate IronWolf (рис. 42). Эти диски отличаются пониженными оборотами шпинделя **5400 об/мин**, более высокой надежностью и способностью работать 24 часа в сутки.

Рисунок 42 – Жесткие диски для NAS

1.18. ПРАКТИЧЕСКАЯ РАБОТА
Доступ к общей папке с Android-устройства

К папке (доступ к которой мы открыли ранее в разделе **1.15**) можно получить доступ не только с устройств на Windows, но и из любых других операционных систем, например, из Android.

Важно, чтобы оба устройства при этом лежали в одной Локальной сети (например, были подключены к одному **Wi-Fi** роутеру).

Total Commander

Установим на Android-устройство программу Total Commander (рис. 43).

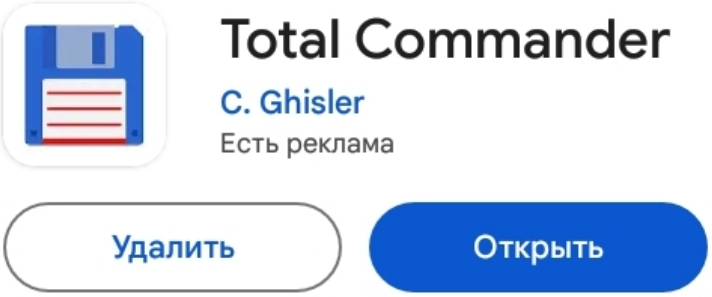


Рисунок 43 – Total Commander в Google Play маркете

Для работы с различными сетевыми протоколами в Total Commander имеются Плагин («plug-in» – «подключать»), т. е. дополнительные модули (рис. 44). Необходимо выбрать пункт «**Скачать плагины**».

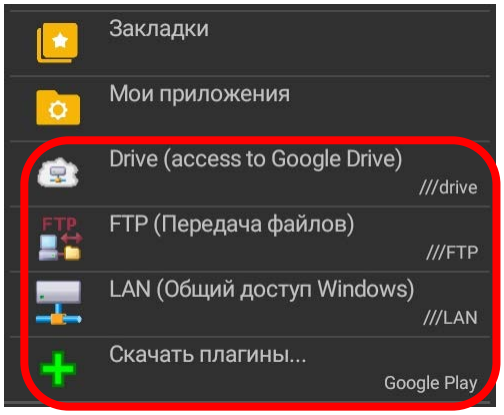


Рисунок 44 – Плагины Total Commander

Для работы с локальной сетью Windows требуется установить плагин LAN (рис. 45).

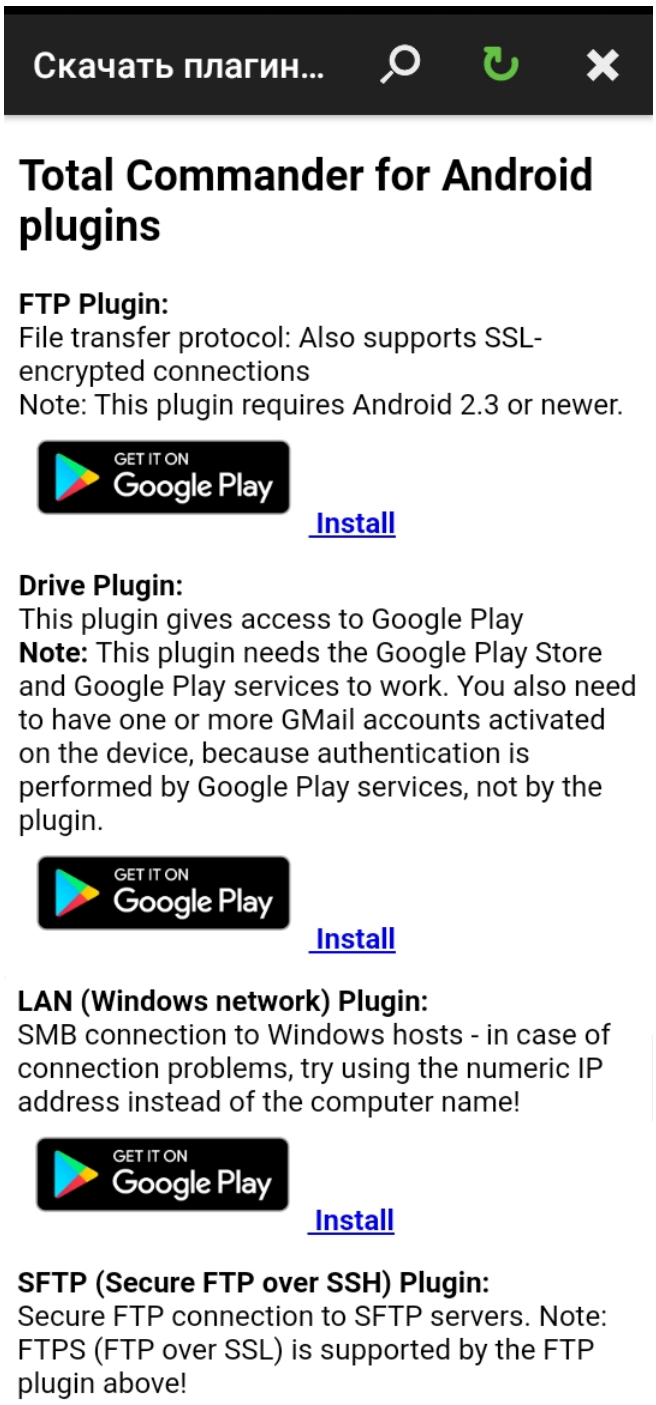


Рисунок 45 – Плагины Total Commander

Далее создаем новый сервер (рис. 46). В настройках необходимо указать IP-адрес сервера (рис. 47), т. е. компьютера, к которому производится подключение.

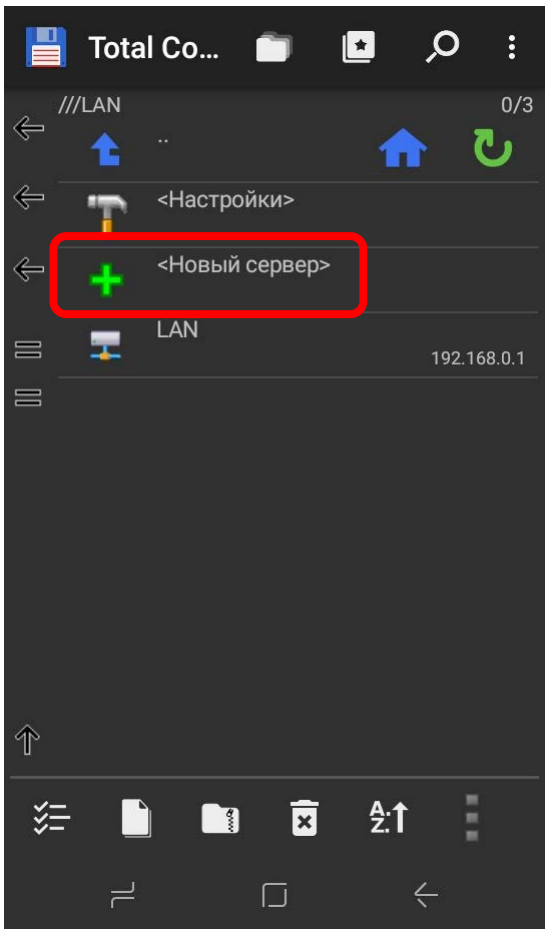


Рисунок 46 – Создание подключения

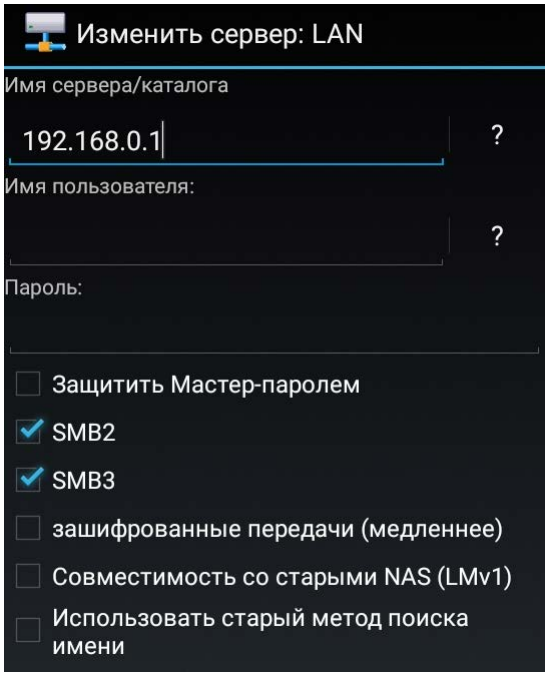


Рисунок 47 – Настройка подключения

IP-адрес и MAC-адрес в Android

Для того чтобы в Android узнать IP-адрес и MAC-адрес, необходимо зайти в Настройки/Система/«Сведения о телефоне»/**Состояние** (рис. 48).

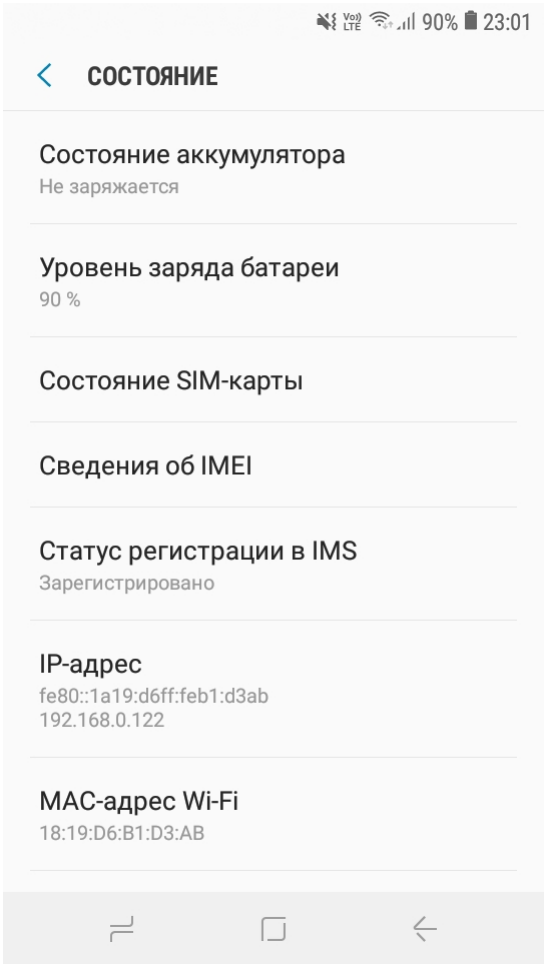


Рисунок 48 – IP-адрес и MAC-адрес в Android

2. СПОСОБЫ ПЕРЕДАЧИ ДАННЫХ

2.1. Режимы передачи данных

Существует три режима передачи данных:

- односторонний (симплексный);
- дуплексный (полнодуплексный);
- полудуплексный.

При односторонней передаче данных (рис. 49) одно устройство всегда является только передающим, а другое только принимающим. Примером одностороннего режима может служить телевидение или радио.

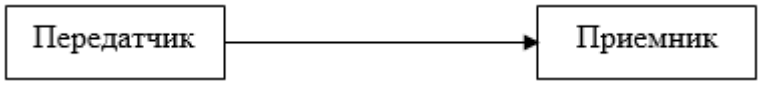


Рисунок 49 – Односторонний режим передачи данных

При полнодуплексном режиме (рис. 50) данные передаются в обоих направлениях одновременно.

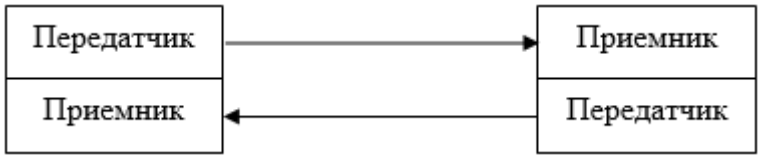


Рисунок 50 – Полнодуплексный режим передачи данных

При полудуплексном режиме (рис. 51) данные передаются в обоих направлениях, но по очереди (по одной паре проводов).

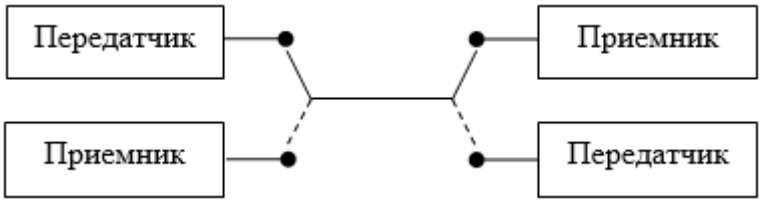


Рисунок 51 – Полудуплексный режим передачи данных

2.2. Типы модуляции

Существуют два способа передачи информации: цифровой и аналоговый [2]. **Цифровые** данные передаются по проводнику импульсно, путем смены текущего напряжения (рис. 52): «1» – есть напряжение, «0» – нет напряжения.

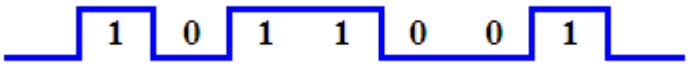


Рисунок 52 – Цифровой способ передачи

При **аналоговом** способе цифровые данные передаются путем управления параметрами сигнала несущей частоты.

Канал называется **узкополосным**, если по нему передаются данные только на одной частоте.

Канал называется **широкополосным**, если он пропускает много частот, т. е. каждый абонент работает на своей собственной частоте.

Несущая частота

Несущая частота представляет собой гармонический сигнал, описываемый уравнением:

$$X = X_m \cdot \sin(\omega \cdot t + \varphi_0),$$

где X_m – амплитуда колебаний;

ω – частота колебаний;

t – время;

φ_0 – начальная фаза колебаний.

Передать цифровую информацию по аналоговому каналу можно, управляя одним из параметров сигнала несущей частоты: амплитудой, частотой или фазой. Соответственно, существуют следующие типы модуляции:

– амплитудная модуляция;
– частотная модуляция;
– фазовая модуляция.

*Цифровая модуляция также может называться «**Манипуляция**» (амплитудная манипуляция, частотная манипуляция, фазовая манипуляция).*

Амплитудная модуляция

При амплитудной модуляции (рис. 53) «0» означает отсутствие сигнала (т. е. отсутствие колебаний несущей частоты), а «1» означает наличие сигнала (т. е. наличие колебаний несущей частоты).

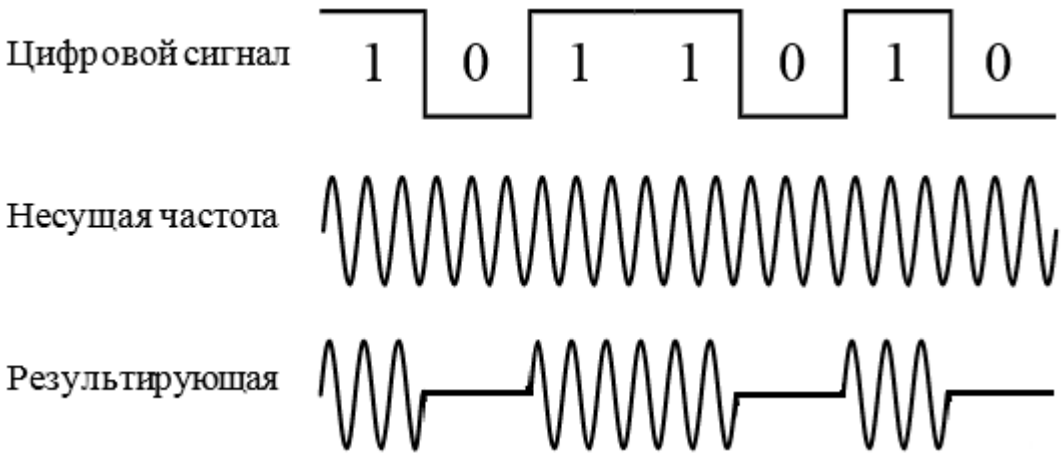


Рисунок 53 – Амплитудная модуляция

Частотная модуляция

При частотной модуляции (рис. 54) передача сигналов «0» и «1» осуществляется на разных частотах.

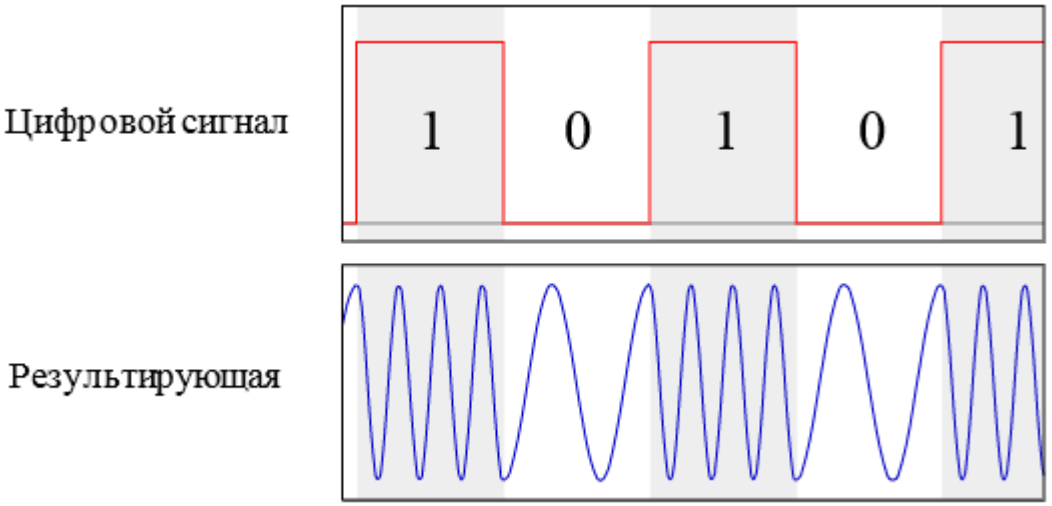


Рисунок 54 – Частотная модуляция

В этом случае используется сразу две несущие частоты, одна для «0», другая для «1».

Фазовая модуляция

При фазовой модуляции (рис. 55) в момент перехода от «0» к «1» и от «1» к «0» изменяется фаза колебаний.

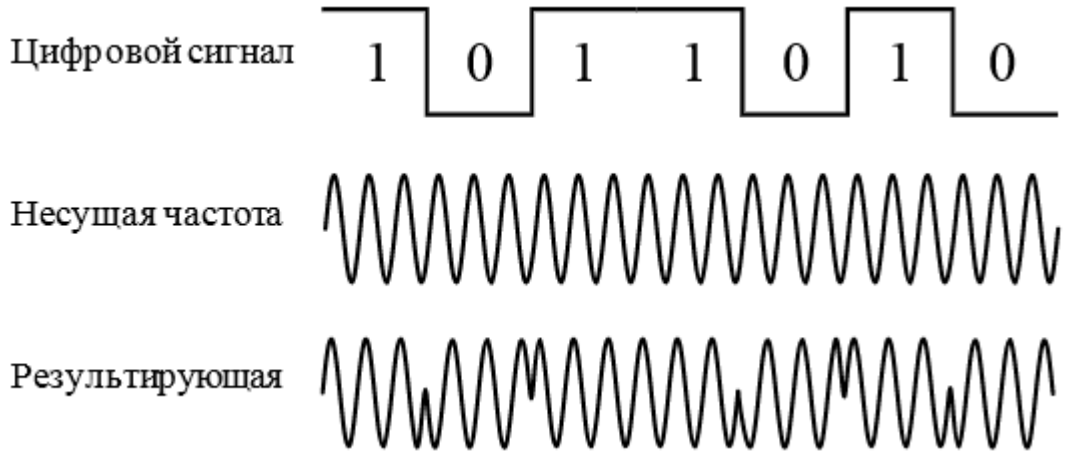


Рисунок 55 – Фазовая модуляция

2.3. Синхронная и асинхронная передача

Существует два способа передачи данных: синхронный и асинхронный.

Асинхронный (т. е. не привязанный ко времени) способ передачи данных – это способ передачи по последовательному интерфейсу, при котором данные передаются в любой момент времени. Для того чтобы приемник инициировал прием данных, вводятся специальные битовые последовательности, обрамляющие данные – это **стартовые** и **стоповые** биты.

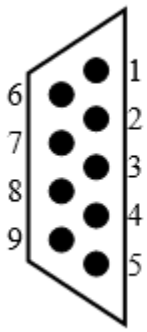
Асинхронный способ передачи используется, например, в последовательных интерфейсах RS-232 и RS-485.

Синхронный (т. е. привязанный ко времени) способ передачи данных – способ передачи по последовательному интерфейсу, при котором приемнику и передатчику известно время передачи данных, то есть передатчик и приемник работают синхронно, в такт.

Асинхронная передача характеризуется легкостью построения системы приема-передачи и, как следствие, низкой стоимостью интерфейсного оборудования. Но при этом асинхронный способ имеет меньшую пропускную способность (скорость) за счет передачи служебных битов (стартовых и стоповых).

2.4. Последовательный интерфейс RS-232

Интерфейс RS-232 представляет собой обычный **COM**-порт персонального компьютера (рис. 56) в виде разъема **DB-9** (семейства **D-Sub**). Передача осуществляется асинхронно.



Контакты разъема:

2 – RxD – прием данных;

3 – TxD – передача данных;

5 – GROUND – земля.

Рисунок 56 – Внешний вид COM-разъема *(со стороны компьютера)*

Интерфейс RS-232 позволяет соединить между собой два устройства на расстоянии 3-5 м.

Передача данных (рис. 57) может осуществляться в обоих направлениях одновременно (полный дуплекс).

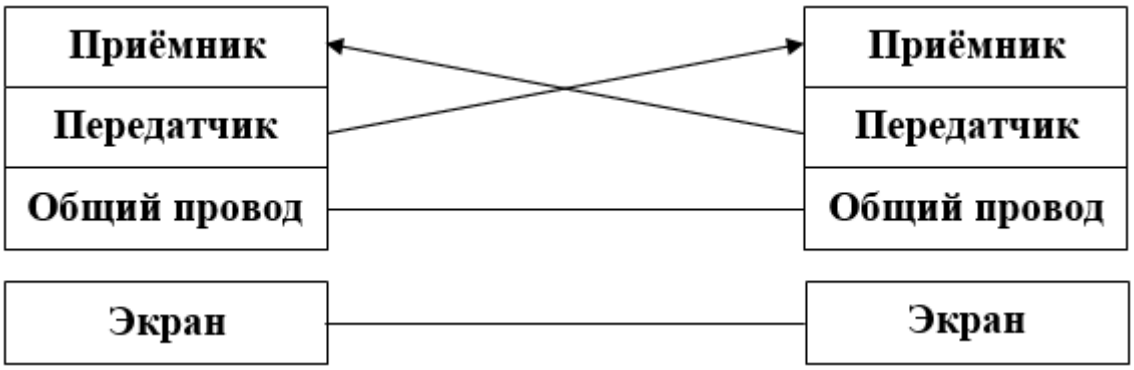


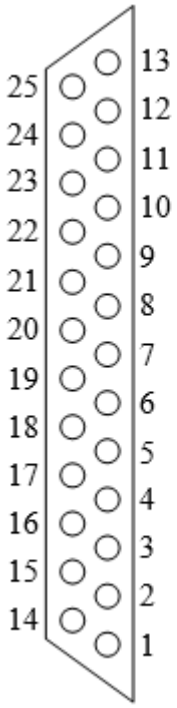
Рисунок 57 – Соединение по RS-232

2.5. Последовательный интерфейс RS-485

В промышленности большее распространение получил последовательный интерфейс **RS-485**, позволяющий соединить по топологии «**шина**» до 32 устройств на расстоянии в **1200** м. Соединение осуществляется по витой паре по двум проводам (полудуплекс) или, в более редком случае, по четырем проводам (полный дуплекс). Обычно применяется схема – один «ведущий» и несколько «ведомых». Как правило используется совместно с протоколом **ModBus**. На концах линии RS-485 обязательно устанавливаются **терминаторы!**

2.6. Параллельный порт

Рассмотрим принцип работы параллельного порта на примере LPT-порта, который служил ранее для подключения принтера и в настоящее время считается устаревшим. Параллельный LPT-порт (рис. 58) позволяет передавать одновременно 8 бит данных и принимать 5 бит состояния. Разъем **DB-25** (**D-Sub**).



Контакты разъема:

2-9 – D0-D7 –выходные биты данных;

18 – питание +5В;

10-13,15 – входные биты состояния;

18-25 – GND – корпус.

Рисунок 58 – Внешний вид LPT-разъема *(со стороны компьютера)*

LPT не предназначался для соединения компьютер-компьютер, хотя такая связь и возможна (рис. 59). В этом случае в каждом из направлений одновременно можно передавать не более пяти бит данных.

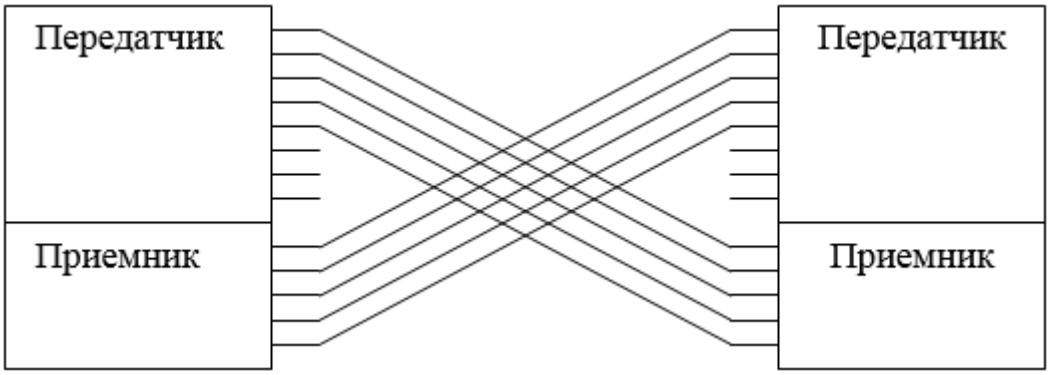


Рисунок 59 – Соединение двух компьютеров по LPT

LPT-порт устарел, а почти все внешние устройства (включая принтеры) подключаются теперь по **USB**. Но параллельная передача все еще массово используется, но теперь только внутри ПК. Параллельное подключение имеют, например, процессор или оперативная память. В **USB 3.0** также было увеличено количество пар проводов с одной до трех.

2.7. ЛАБОРАТОРНАЯ РАБОТА № 2
Передача данных по последовательному порту

Установить соединение через **COM**-порт можно при помощи одной из множества терминальных программ, например, таких как:

- **Hyper Terminal**;
- **PuTTY**;
- **Serial Monitor** среды Arduino IDE.

В Windows XP программа **Hyper Terminal** («Гипертерминал»), расположена в меню ПУСК/Программы/Стандартные/Связь.

После запуска программы необходимо:

- указать название (рис. 60) соединения (без использования русских букв и пробелов);

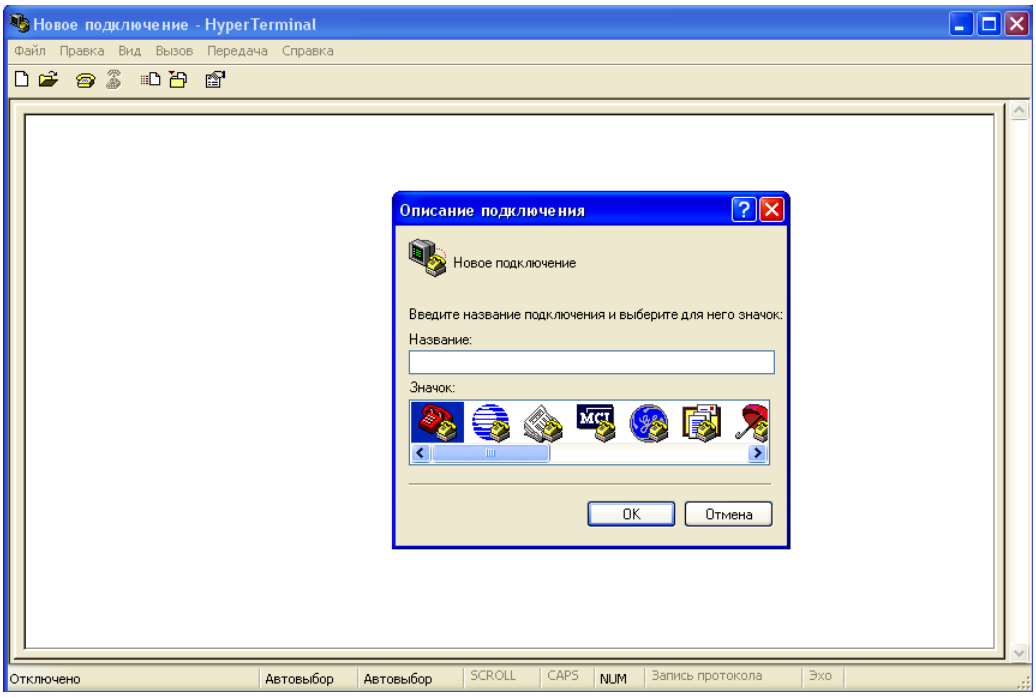


Рисунок 60 – Внешний вид программы Hyper Terminal

- выбрать порт (рис. 61), к которому физически подсоединен кабель (например, COM1);

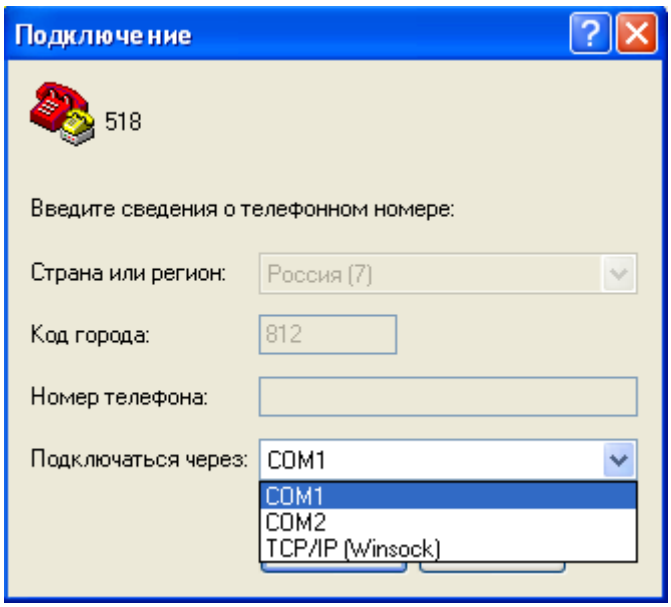


Рисунок 61 – Выбор COM-порта

– указать параметры (рис. 62) порта (скорость, биты данных, четность, стоповые биты), одинаковые для обоих компьютеров.

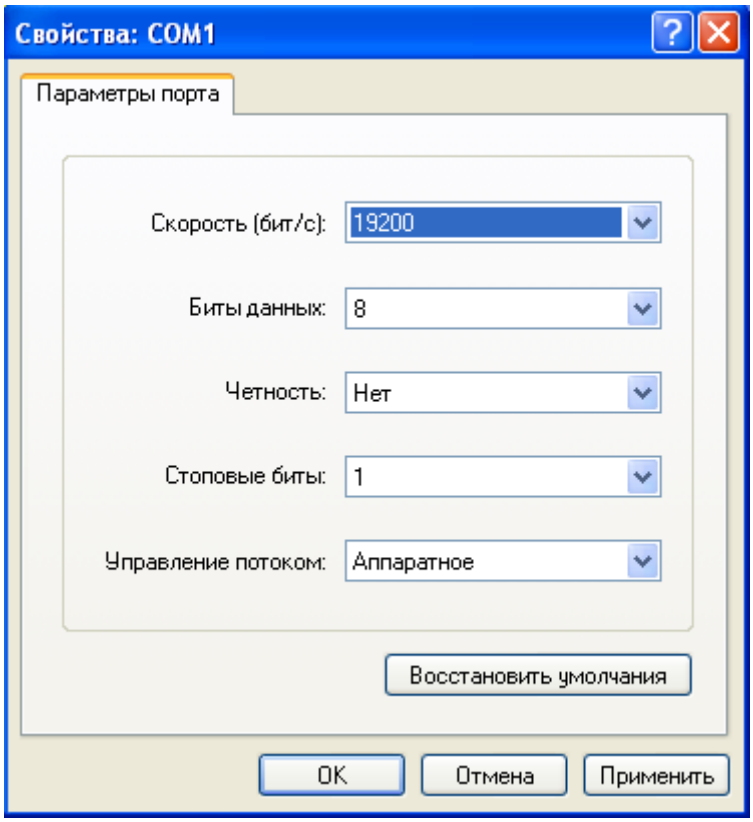


Рисунок 62 – Настройка COM-порта

В случае, если соединение установлено правильно, текст, вводимый в окне одного компьютера, будет виден в окне другого компьютера. Также программа позволяет передавать через COM-порт файлы.

2.8. *Serial Monitor среды Arduino IDE

Среда программирования микроконтроллеров Arduino IDE (рис. 63), является свободно распространяемым программным обеспечением с открытым исходным кодом. В комплект Arduino IDE входит монитор последовательного порта Serial Monitor (рис. 64), позволяющий осуществлять обмен данными с контроллером в консольном режиме (аналогично описанной выше программе Hyper Terminal).

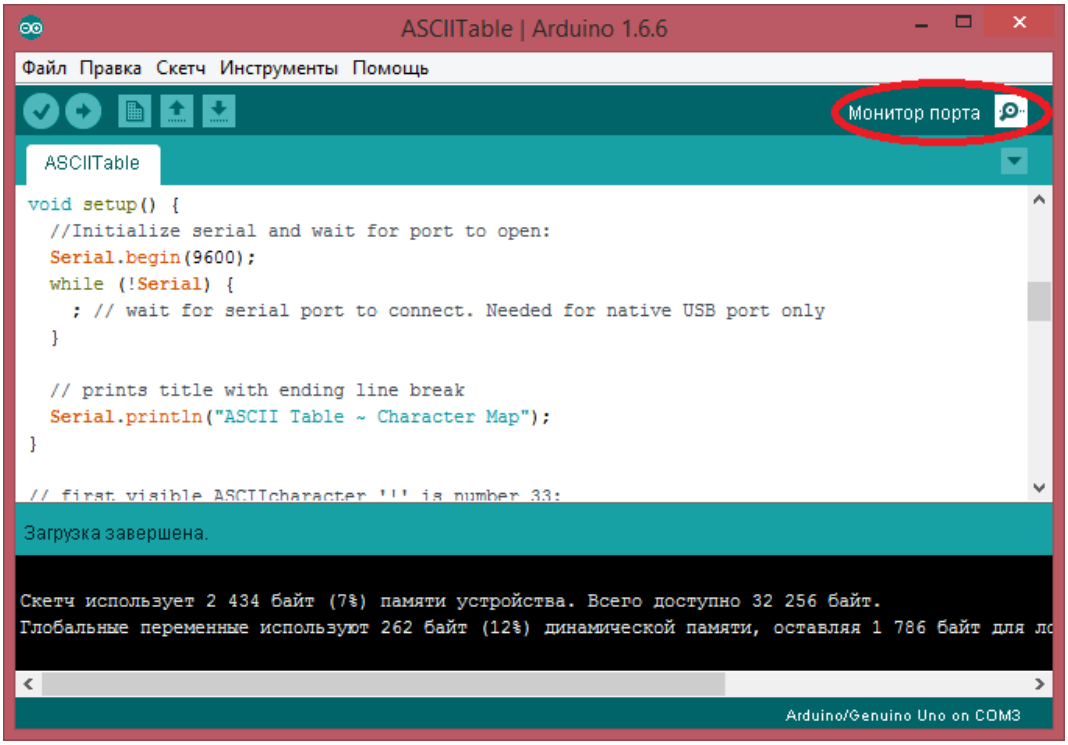


Рисунок 63 – Среда программирования Arduino IDE

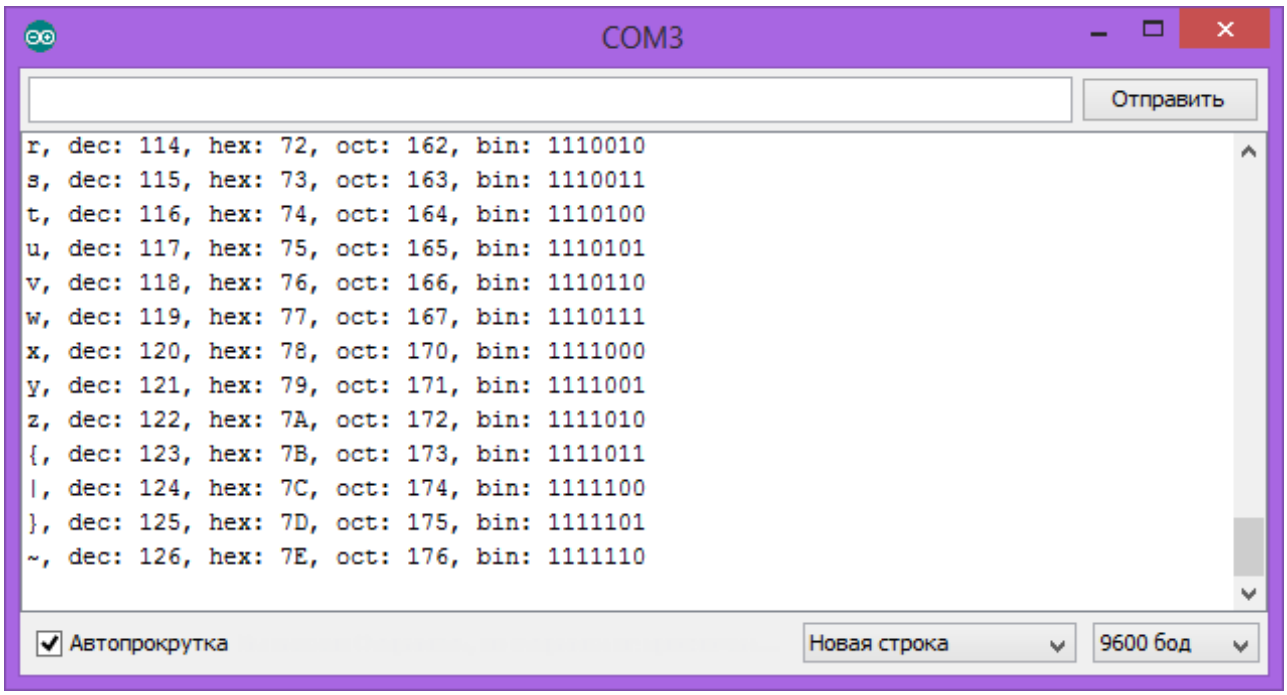


Рисунок 64 – Serial Monitor

Для правильной работы Serial Monitor необходимо задать скорость передачи данных (например, «9600» на рис. 64), которая ранее была прописана в программе контроллера.

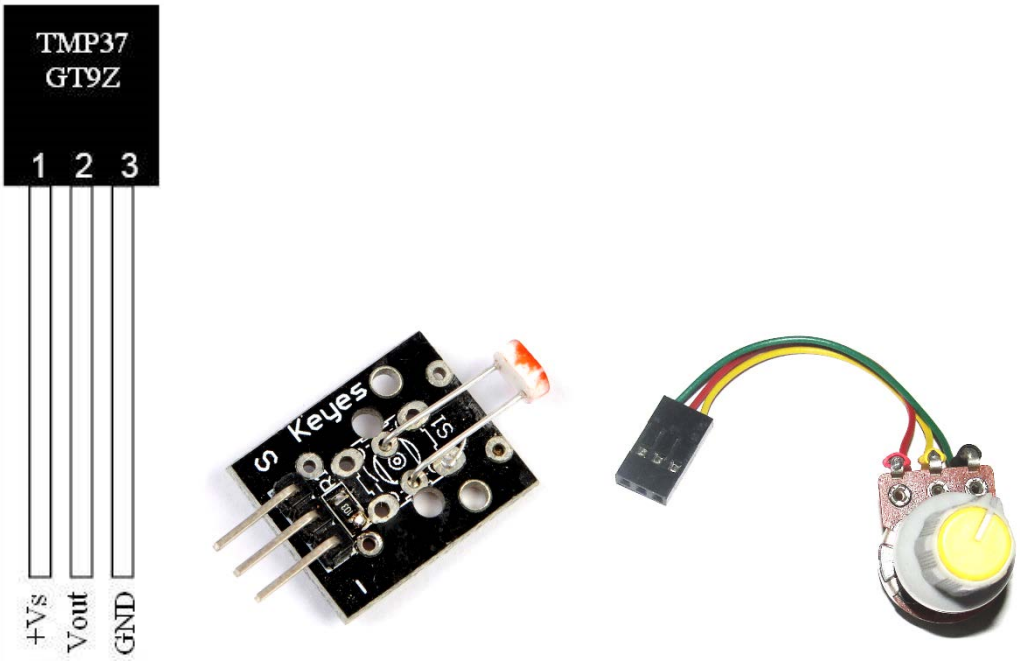
2.9. ЛАБОРАТОРНАЯ РАБОТА № 3
Web-сервер на микроконтроллере Arduino

Arduino – аппаратно-программная платформа с открытой архитектурой и исходным кодом, основанная на специально разработанной среде разработки и плате ввода-вывода информации. На плате имеются разъемы для подключения внешних устройств и установлен разъем USB для связи с компьютером, по которому осуществляется программирование микроконтроллера.

Микроконтроллер Arduino совместно с модулем Ethernet Shield (рис. 65) позволяет реализовать различные протоколы, такие как: TCP, UDP, Telnet, DHCP, NTP, а также создать web-сервер с протоколом HTTP.

Рисунок 65 – Модуль Arduino Ethernet Shield

К аналоговым входам контроллера подключаются датчик температуры TMP37, фоторезистор и потенциометр (рис. 66). При необходимости аналогично могут быть подключены и любые другие аналоговые датчики с выходным сигналом, лежащим в интервале 0-5 В. *Для подключения дискретных датчиков необходимо использовать команду digitalRead вместо analogRead.*



а) б) в)

Рисунок 66 – Аналоговые датчики:
а) датчик температуры; *б)* фоторезистор; *в)* потенциометр

Запустим среду Arduino IDE и откроем в ней пример Ethernet/WebServer (рис. 67). **MAC-адрес** и **IP-адрес** для модуля Ethernet Shield прописываются непосредственно в коде программы (по умолчанию DE-AD-BE-EF-FE-ED и 192.168.1.177).

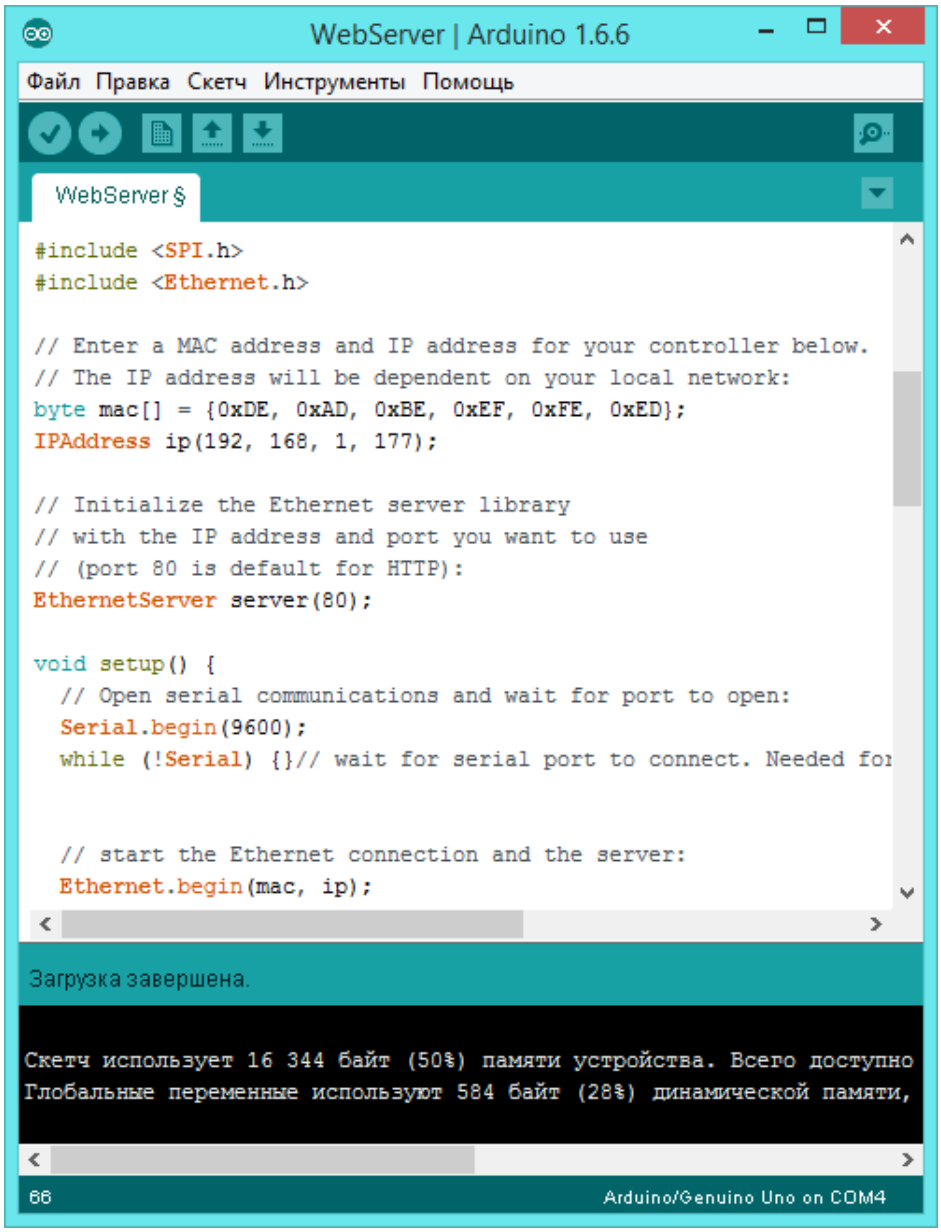


Рисунок 67 – Код программы «WebServer»

Для HTTP-протокола используется порт 80. Запуск сервера осуществляется по команде:

Ethernet.begin(mac, ip);

Требуемые для работы библиотеки SPI.h и Ethernet.h уже подключены к данному примеру. Необходимо только осуществить загрузку этого кода программы в контроллер Arduino с установленным Ethernet-модулем и подключенными аналоговыми датчиками.

На стороне клиента (ПК) необходимо осуществить настройку сетевого соединения с указанием IP-адреса (рис. 68), лежащего в той же подсети, что и IP-адрес сервера, который указан в коде программы (см. рис. 67).

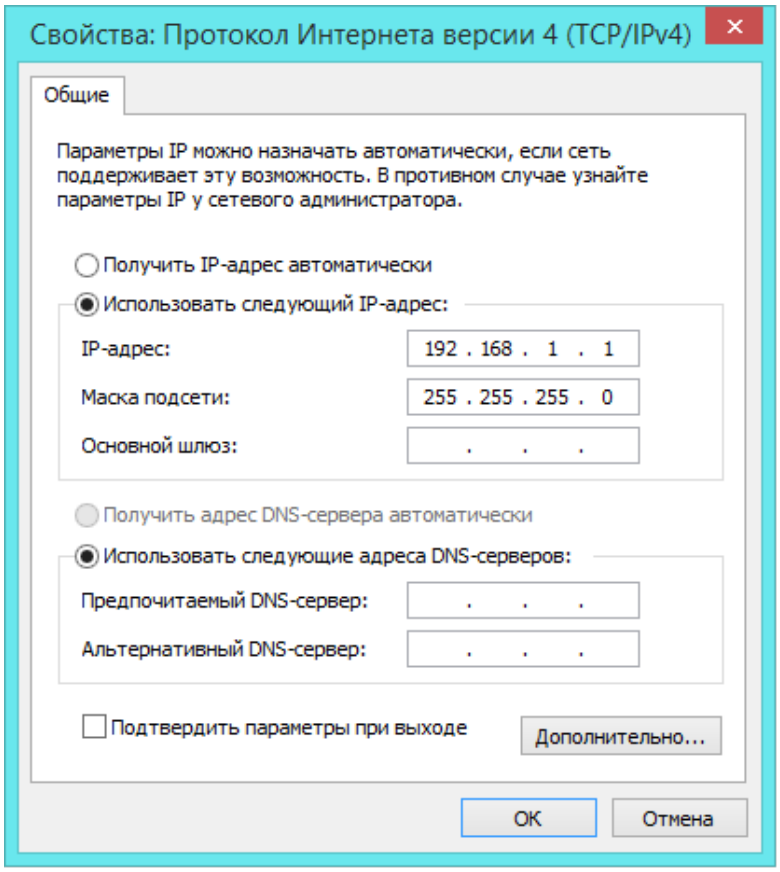


Рисунок 68 – Настройка ПК для связи с web-сервером

Проверка соединения с сервером осуществляется при помощи любого браузера (рис. 69). При этом в качестве адреса страницы указывается тот самый IP-адрес сервера, который ранее был задан в коде программы.

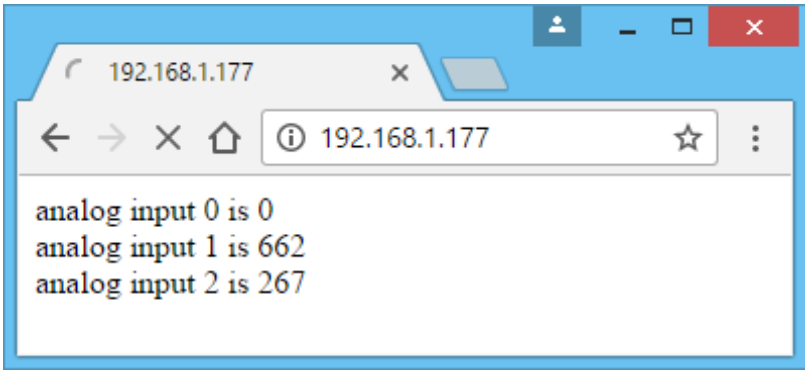


Рисунок 69 – Результат работы web-сервера

В результате в браузере отобразятся текущие показания подключенных датчиков, значения которых будут автоматически обновляться через заданный в программе интервал времени.

3. АДРЕСАЦИЯ В СЕТИ

3.1. Доменные имена

Когда вы обращаетесь к Web-странице или посылаете e-mail, вы используете доменное имя [3]. Например, адрес **http://www.microsoft.com/** содержит доменное имя microsoft.com. Аналогично e-mail адрес **myname@mail.ru** содержит доменное имя mail.ru.

В доменной системе имен реализуется принцип назначения имен с определением ответственности за их подмножество. И если каждая группа придерживается этого простого правила и всегда получает подтверждение, что имена, которые она присваивает, единственны среди множества ее непосредственных подчиненных, то никакие две системы, где бы те ни находились в сети Интернет, не смогут получить одинаковые имена.

Также уникальны адреса, указываемые на конвертах при доставке писем обычной почтой. Таким образом, адрес на основе географических и административных названий однозначно определяет точку назначения. Домены тоже имеют аналогичную иерархию.

Домены отделяются друг от друга точками: companya.msk.ru, companyb.spb.ru. В имени может быть различное количество доменов, но обычно их не больше пяти.

Каждый раз, когда вы используете доменное имя, вы также используете DNS-серверы для того, чтобы перевести буквенное доменное имя в IP-адрес на машинном языке.

В качестве примера рассмотрим адрес **www.pc.dpt1.company.spb.ru**.

Первым в имени стоит название рабочей машины – реального компьютера с IP-адресом. Это имя создано и поддерживается группой dpt1. Группа входит в более крупное подразделение company, далее следует домен spb – он определяет имена петербургской части сети, а ru – российской.

Каждая страна имеет свой домен. Так, au – соответствует Австралии, be – Бельгии и т. д. Это географические домены верхнего уровня. Помимо географического признака, используется тематический, в соответствии с которым существуют следующие доменные имена первого уровня:

- com – обозначает коммерческие предприятия;
- edu – образовательные (для США);
- gov – государственные (для США);
- mil – военные (для США);
- net – сетевые;
- org – другие организации;
- и т. д.

Внутри каждого доменного имени первого уровня находится целый ряд доменных имен второго уровня. Домены верхних уровней располагаются в имени правее, а домены нижних уровней – левее.

Например, на рис. 70 показана структура адреса ряда «организаций» на примере российского домена.

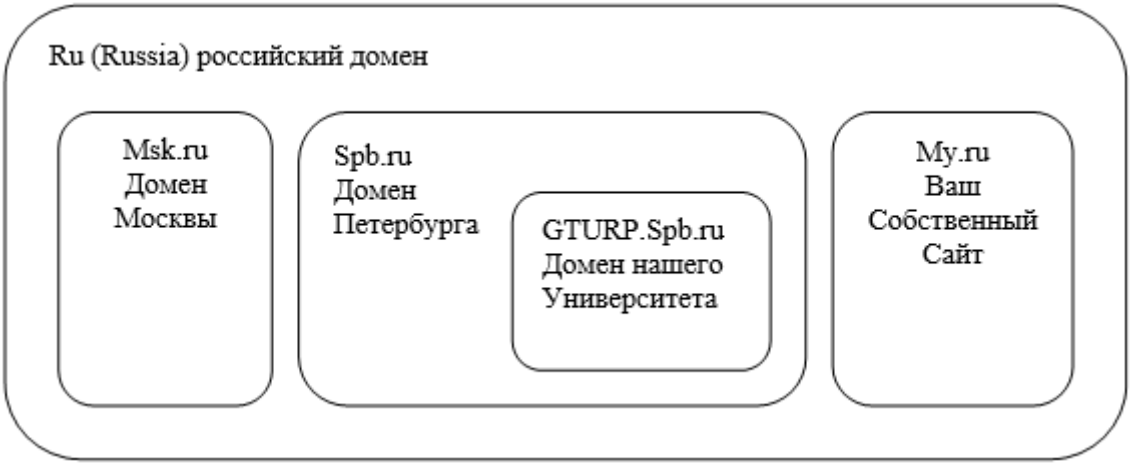


Рисунок 70 – Структура адреса на примере российского домена

Рассмотрим адрес **http://www.gturp.spb.ru/**. Домен верхнего уровня **ru** указывает на то, что адрес принадлежит российской части Интернета, **spb** – определяет город, следующий уровень – домен конкретной организации.

3.2. DNS-сервер

DNS-сервер принимает запрос на конвертацию доменного имени в IP-адрес. При этом DNS-сервер выполняет следующие действия:

- отвечает на запрос, выдав IP-адрес, поскольку уже знает адрес запрошенного домена;
- контактирует с другим DNS-сервером для того, чтобы найти IP-адрес запрошенного имени. Этот запрос может проходить по цепочке несколько раз;
- выдает сообщение: «Я не знаю IP-адрес домена, запрошенного вами, но вот IP-адрес DNS-сервера, который знает больше меня»;
- сообщает, что такой домен не существует.

Допустим, что вы набрали адрес ***http://www.pc.dpt1.company.com/*** *в вашем браузере, который имеет адрес в домене верхнего уровня COM (рис. 71).*

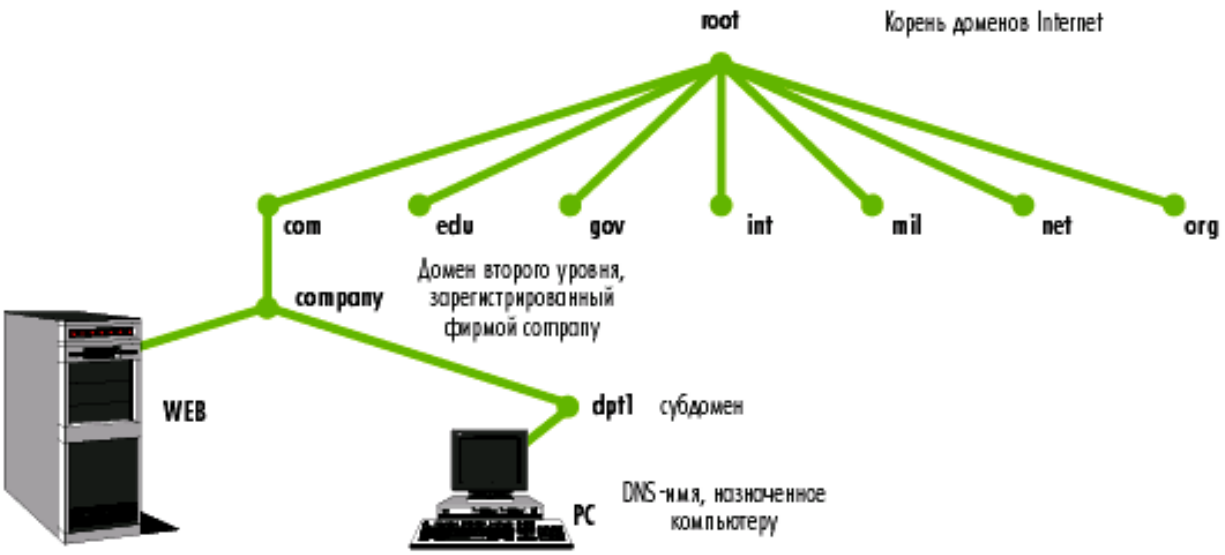


Рисунок 71 – Структура адреса для pc.dpt1.company.com

В простейшем варианте ваш браузер контактирует с DNS-сервером для того, чтобы получить IP-адрес искомого компьютера, и DNS-сервер возвращает искомый IP-адрес (рис. 72).

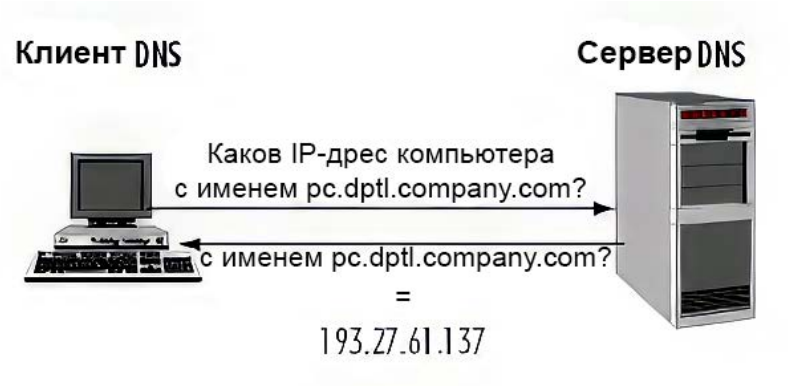


Рисунок 72 – Запрос к DNS-серверу на получение IP-адреса

На практике в Сети, где объединены миллионы компьютеров, найти DNS-сервер, который знает нужную вам информацию – это целая проблема. Иными словами, если вы ищете какой-то компьютер в Сети, то прежде всего вам необходимо найти DNS-сервер, на котором хранится нужная вам информация. При этом в поиске информации может быть задействована целая цепочка серверов (рис. 73).

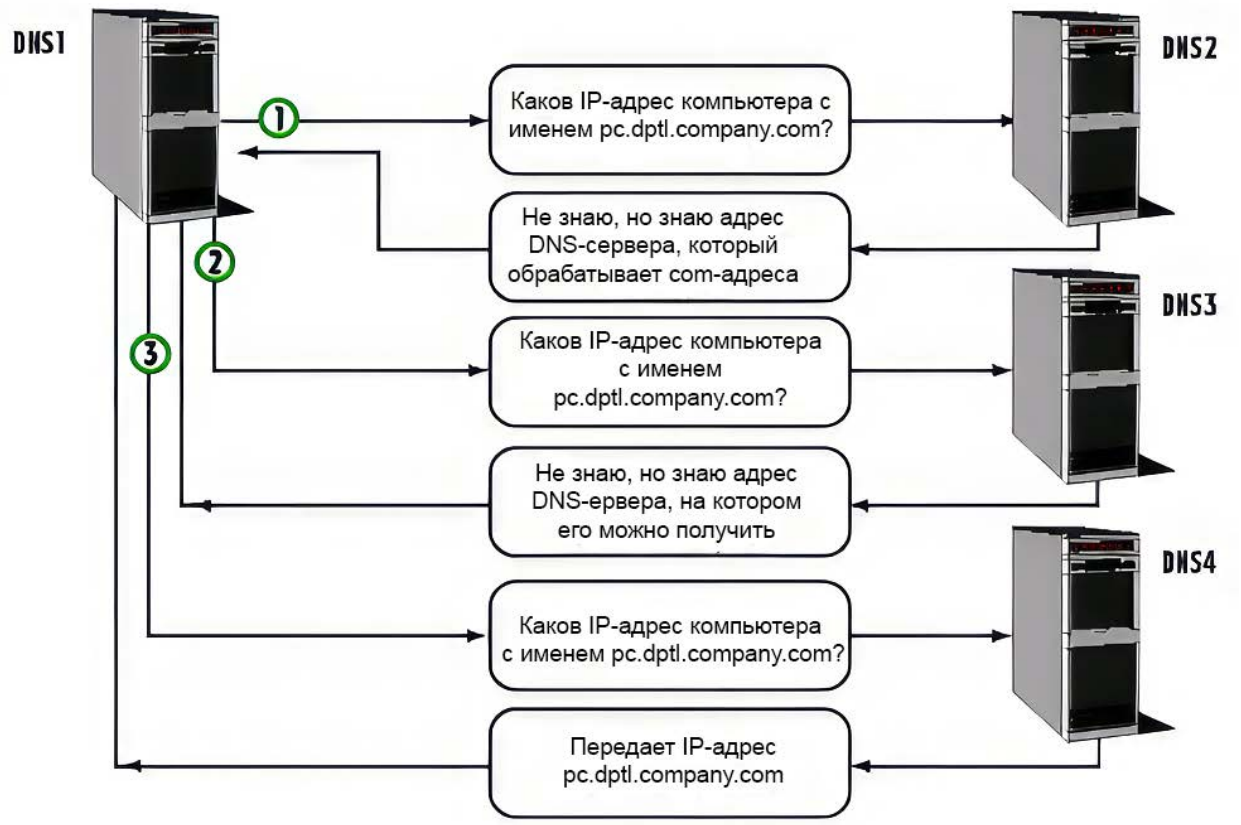


Рисунок 73 – Поиск DNS-сервера

Предположим, что тот DNS-сервер, к которому вы обратились (на рис. 73 он обозначен как DNS1), не имеет нужной информации. DNS1 начнет поиск IP-адреса с обращения к одному из корневых DNS-серверов. Корневые DNS-серверы знают IP-адреса всех DNS-серверов, отвечающих за доменные имена верхнего уровня (RU, COM, EDU, GOV, MIL, NET, ORG и т.д.).

Например, ваш сервер DNS1 может запросить адрес у корневого DNS-сервера. Если корневой сервер не знает данного адреса, возможно, он даст ответ: «Я не знаю IP-адреса для ***http://www.pc.dpt1.company.com/****, но могу предоставить IP-адрес COM DNS-сервера».*

После этого ваш DNS1 посылает запрос на COM DNS с просьбой сообщить искомый IP-адрес. Так происходит до тех пор, пока не найдется DNS-сервер, который выдаст нужную информацию.

Одна из причин, по которой система работает надежно – это ее избыточность. Существует множество DNS-серверов на каждом уровне, и поэтому, если один из них не может дать ответ, наверняка существует другой, на котором есть необходимая вам информация. Другая технология, которая делает поиск более быстрым – это система кэширования. Как только DNS-сервер выполняет запрос, он кэширует полученный IP-адрес. Однажды сделав запрос на корневой DNS (root DNS) и получив адрес DNS-сервера, обслуживающего COM-домены, в следующий раз он уже не должен будет повторно обращаться с подобным запросом. Кэширование происходит с каждым запросом, что постепенно оптимизирует скорость работы системы. Несмотря на то, что

пользователям работа DNS-сервера не видна, эти серверы каждый день выполняют миллиарды запросов, обеспечивая работу миллионов пользователей.

Рассмотрим сеть небольшой компании (рис. 74), подключенной к Интернет.

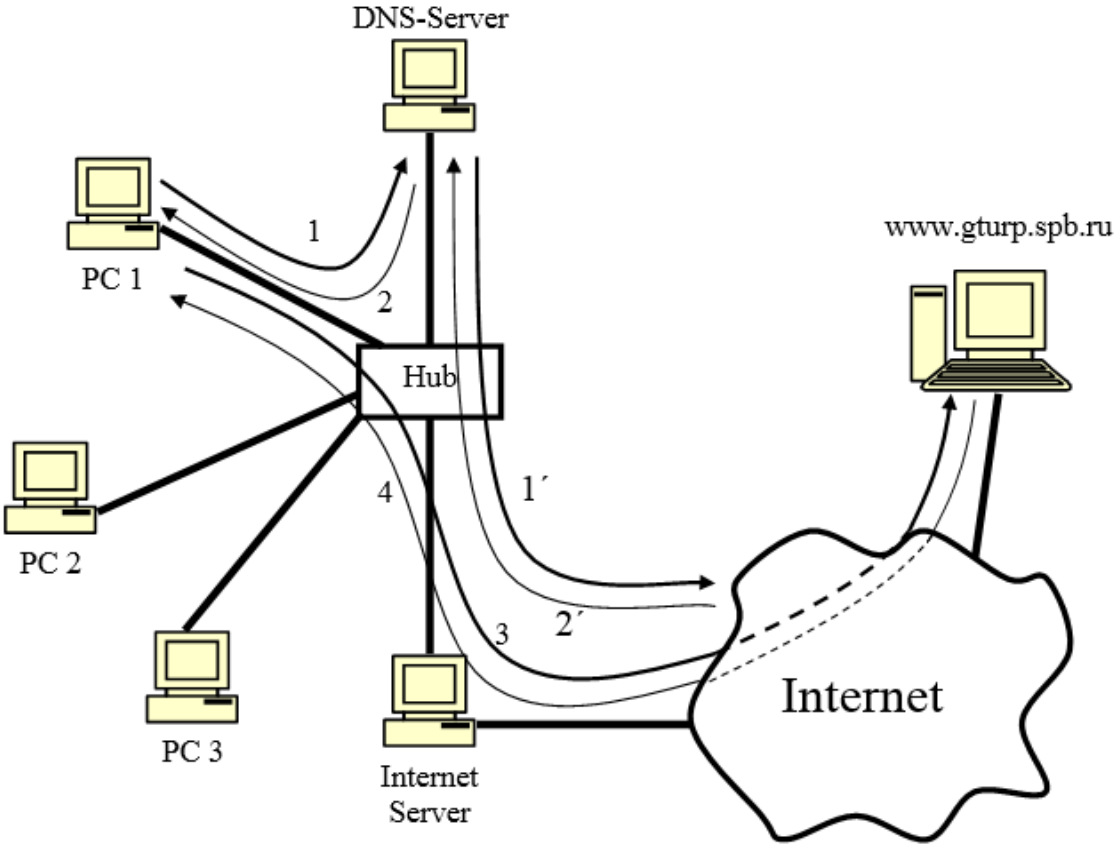


Рисунок 74 – Принцип работы DNS-сервера:

1 – запрос на получение IP-адреса по имени www.gturp.spb.ru;

2 – передача IP-адреса www.gturp.spb.ru;

3 – запрос на получение данных интернет странички;

4 – передача интернет странички www.gturp.spb.ru;

1´ – перенаправление запроса на получение IP-адреса по имени;

2´ – ответ на перенаправленный запрос (1´)

Пусть пользователь станции PC1 хочет загрузить страничку сайта, расположенного по адресу ***www.gturp.spb.ru****. Для этого необходимо получить IP-адрес сервера, на котором хранится страничка. PC1 отправляет запрос (1) на получение IP-адреса DNS-серверу. DNS-сервер возвращает ответ (2) с результатом обработки запроса (в случае необходимости DNS-сервер может перенаправить запрос (1′) на другие DNS-сервера, получая (2′) от них требуемую информацию). Если в результате обработки запроса DNS-серверу удалось найти IP-адрес соответствующего сервера, то PC1 отправляет запрос (3) на данный сервер и получает в ответ (4) запрашиваемую информацию.*

Нужно заметить, что DNS-сервер может выполнять и обратную задачу, т. е. получить доменное имя по IP-адресу.

3.3. Типы адресов (MAC, IP, DNS)

Каждый компьютер в сети TCP/IP имеет адреса трех уровней:

- **Локальный адрес узла**, т. е. MAC-адрес сетевого адаптера или порта маршрутизатора. Локальные адреса записываются в шестнадцатеричном виде, например, **11-A0-17-3D-BC-01** (или, иногда, 11:A0:17:3D:BC:01). Эти адреса назначаются производителями оборудования и являются уникальными адресами, так как управляются централизованно. Для всех существующих технологий локальных сетей MAC-адрес имеет формат 6 байтов: старшие 3 байта – идентификатор фирмы производителя, а младшие 3 байта назначаются уникальным образом самим производителем.
- **IP-адрес**, состоящий из 4 байт, например, **109.26.17.100**. Этот адрес используется на сетевом уровне. Он назначается администратором во время конфигурирования компьютеров и маршрутизаторов. IP-адрес состоит из двух частей: номера сети и номера узла. Номер сети может быть выбран администратором произвольно, либо назначен по рекомендации специального подразделения Internet (Network Information Center, **NIC**), если сеть должна работать как составная часть Internet. Обычно провайдеры услуг Internet получают диапазоны адресов у подразделений NIC, а затем распределяют их между своими абонентами.

Номер узла в протоколе IP назначается независимо от локального адреса узла. Деление IP-адреса на поле номера сети и номера узла – гибкое, и граница между этими полями может устанавливаться весьма произвольно. *Узел может входить в несколько IP-сетей. В этом случае узел должен иметь несколько IP-адресов, по числу сетевых связей. Таким образом, IP-адрес характеризует не отдельный компьютер или маршрутизатор, а одно сетевое соединение.*

- **Символьный идентификатор** – имя вида: **SERV1.IBM.COM**. Этот адрес назначается администратором и состоит из нескольких частей, например, имени машины, имени организации, имени домена. Такой адрес, называемый также **DNS**-именем, используется на прикладном уровне, например, в протоколах FTP или telnet.

3.4. ПРАКТИЧЕСКАЯ РАБОТА
Структура IP-адреса

IP-адрес состоит из 4 байтов (рис. 75) и записывается в виде четырех чисел, содержащихся в каждом байте, разделенных точками, например:

128.10.2.30 – десятичная форма представления адреса;

10000000 00001010 00000010 00011110 – двоичная форма.

Рисунок 75 – Структура IP-адреса

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

- Если адрес начинается с **0**, то сеть относят к классу **A**, номер сети занимает один байт, остальные 3 байта интерпретируются как номер узла в сети. Из этого следует, что сети класса A имеют номера в диапазоне от 1 до 127. В сетях класса A количество узлов должно быть больше 2^{16} – не превышая 2^{24}. Это сети больших размеров.
- Если первые два бита адреса равны **10**, то сеть относится к классу **B** и является сетью средних размеров с числом узлов $2^8 – 2^{16}$. В сетях класса **B** под адрес сети отводится 14 бит, т. е. 2^{14} сетей, а под адрес узла – 16 бит.
- Если адрес начинается с последовательности **110,** то сеть относится к классу **C**, в котором не может быть больше, чем 256 узлов, а под адрес сети отводится 21 двоичный разряд. Это сети малых размеров.
- Еще имеются два типа IP-адресов: адрес класса **D**, который начинается с последовательности **1110** и обозначает особый адрес **multicasting**, и адрес класса **E**, который начинается с кода **11110** и зарезервирован для будущих применений.

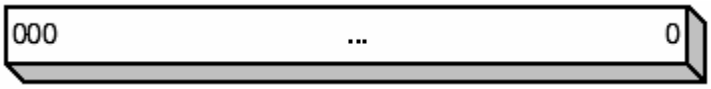
Такая структура адреса позволяет маршрутизаторам очень быстро извлекать из IP-адреса адрес сети.

До этого мы считали, что каждому узлу сети соответствует один IP-адрес. Однако, как быть с маршрутизатором, который связывает несколько сетей? Легко представить и такую ситуацию, когда компьютер входит в несколько сетей. В этом случае сетевой узел должен иметь несколько IP-адресов (по числу сетевых связей). Таким образом, ***IP-адрес характеризует не отдельный компьютер или маршрутизатор, а одно сетевое соединение****.*

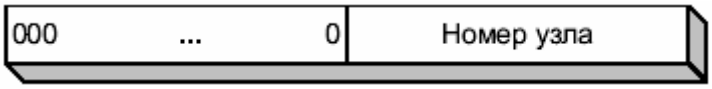
Особые IP-адреса

В протоколе IP существует несколько соглашений об особой интерпретации IP-адресов:

1) если IP-адрес состоит только из двоичных нулей, то он обозначает тот узел, который сгенерировал этот пакет:

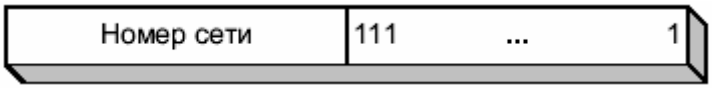


2) если в поле номера сети стоят нули, то интерпретируется только номер узла и предполагается, что этот узел принадлежит той же самой сети, что и узел, который отправил пакет:



3) если все двоичные разряды IP-адреса равны 1, то этот пакет посылается всем узлам, но только относящимся к данной сети (ограниченное широковещательное сообщение – limited broadcast):

4) если сплошные единицы стоят только в поле адреса узла, то это означает широковещательное сообщение для узлов сети с заданным номером:



5) еще один адрес **127.0.0.1** зарезервирован для обозначения обратной связи – «**Loopback**» (замыкание на себя).

Уже упоминавшаяся форма IP-адреса «multicasting» означает, что данный пакет должен быть доставлен сразу нескольким узлам, которые образуют группу, с номером, указанным в поле multicasting. Узлы сами идентифицируют себя, т. е. определяют, к какой из групп они относятся. Один и тот же узел может входить в несколько групп. Такие сообщения, в отличие от широковещательных, называются мультивещательными.

Нужно заметить, что адреса узлов в подсетях классов A, B и C **не могут быть нулевыми**, а также не могут состоять из всех двоичных единиц (в силу правила № 4). Номер сети также не может быть нулевым (правило № 2).

Диапазоны адресов

В табл. 2 приведены диапазоны адресов для сетей классов A, B и C.

Таблица 2 – Диапазоны адресов

Класс	Наименьший адрес	Наибольший адрес
A	1.*.*.*	126.*.*.*
B	128.0.*.*	191.255.*.*
C	192.0.0.*	223.255.255.*

Маска подсети

*Номер сети назначается специальным подразделением Network Information Center (**NIC**). Большие сети получают адреса класса A, средние – класса B, а маленькие – класса C. Получив номер, сетевой администратор часто сталкивается с проблемой, каким образом локализовать трафик в некоторых частях своей сети. Решение может быть найдено путем использования так называемых масок, которые позволяют делить сеть на подсети.*

Маска – это число, двоичная запись которого содержит **«единицы»** в тех разрядах, которые интерпретируются как **адрес сети**, а **«нули»** в тех, которые отводятся под **номер узла.**

Например:

255.0.0.0 – маска для сети класса A;

255.255.0.0 – маска для сети класса B;

255.255.255.0 – маска для сети класса C.

Установив новое значение маски, можно заставить маршрутизатор интерпретировать адрес по-другому. Например, пусть сеть относится к классу B. В одной сети циркулирует единый трафик. Но среди всех станций сети есть некоторые, слабо взаимодействующие между собой. Эти станции желательно изолировать в разных сетях. Пусть это будут узел 129.34.17.15 и узел 129.34.20.1, которые в исходной ситуации относятся к одной сети класса B с номером 129.34. Если задать в качестве маски число 255.255.255.0, то адреса этих двух узлов будут интерпретироваться маршрутизаторами как адреса узла 15 сети класса C с номером 129.34.17 и узла 1 сети класса C с номером 129.34.20. Извне сеть по-прежнему будет выглядеть как единая сеть класса B, а на местном уровне это будет несколько отдельных сетей класса C.

3.5. Протоколы ARP и RARP

В протоколе IP адрес узла, т. е. адрес компьютера или порта маршрутизатора, назначается произвольно администратором сети и прямо не связан с его локальным адресом. Подход, используемый в IP, удобно использовать в крупных сетях по причине его независимости от формата локального адреса, и по причине стабильности, так как в противном случае, при смене на компьютере сетевого адаптера это изменение должны бы были учитывать все адресаты всемирной сети Internet (в том случае, конечно, если сеть подключена к Интернету).

Локальный адрес используется в протоколе IP только в пределах локальной сети при обмене данными между маршрутизатором и узлом этой сети. Маршрутизатор, получив пакет для узла одной из сетей, непосредственно подключенных к его портам, должен для передачи пакета сформировать кадр в соответствии с требованиями принятой в этой сети технологии и указать в нем локальный адрес узла, например, его MAC-адрес. В пришедшем пакете этот адрес не указан, поэтому перед маршрутизатором встает задача поиска его по известному IP-адресу, который указан в пакете в качестве адреса назначения. С

аналогичной задачей сталкивается и конечный узел, когда он хочет отправить пакет в удаленную сеть через маршрутизатор, подключенный к той же локальной сети, что и данный узел.

Для определения локального адреса по IP-адресу используется протокол разрешения адреса **ARP** (Address Resolution Protocol). *Протокол ARP работает различным образом в зависимости от того, какой протокол канального уровня используется в данной сети (Ethernet, Token Ring, FDDI)*. Существует также протокол, решающий обратную задачу – нахождение IP-адреса по известному локальному адресу. Он называется реверсивный ARP (**RARP**, Reverse Address Resolution Protocol) и используется при старте бездисковых станций, не знающих в начальный момент своего IP-адреса, но знающих адрес своего сетевого адаптера.

В локальных сетях протокол ARP использует широковещательные кадры протокола канального уровня для поиска в сети узла с заданным IP-адресом.

Узел, которому нужно выполнить отображение IP-адреса на локальный адрес, формирует ARP запрос, вкладывает его в кадр протокола канального уровня, указывая в нем известный IP-адрес, и рассылает запрос широковещательно. Все узлы локальной сети получают ARP-запрос и сравнивают указанный там IP-адрес с собственным. В случае их совпадения узел формирует ARP-ответ, в котором указывает свой IP-адрес и свой локальный адрес, и отправляет его уже направленно, так как в ARP-запросе отправитель указывает свой локальный адрес. ARP-запросы и ответы используют один и тот же формат пакета. Так как локальные адреса могут в различных типах сетей иметь различную длину, то формат пакета протокола ARP зависит от типа сети.

3.6. Протокол DHCP

Как уже было указано, IP-адреса могут назначаться администратором сети вручную. Это представляет для администратора утомительную процедуру. Ситуация усложняется еще тем, что многие пользователи не обладают достаточными знаниями для того, чтобы конфигурировать свои компьютеры для работы в сети, и должны поэтому полагаться на администраторов.

Протокол **DHCP** (Dynamic Host Configuration Protocol) был разработан для того, чтобы освободить администратора от этих проблем. Основным назначением DHCP является динамическое назначение IP-адресов. Однако, кроме динамического, DHCP может поддерживать и более простые способы ручного и автоматического статического назначения адресов.

В ручной процедуре назначения адресов активное участие принимает администратор, который предоставляет DHCP-серверу информацию о соответствии IP-адресов физическим адресам или другим идентификаторам клиентов. Эти адреса сообщаются клиентам в ответ на их запросы к DHCP-серверу.

При автоматическом статическом способе DHCP-сервер присваивает IP-адрес (и, возможно, другие параметры конфигурации клиента) из пула

имеющихся IP-адресов без вмешательства оператора. Границы пула назначаемых адресов задает администратор при конфигурировании DHCP-сервера. Между идентификатором клиента и его IP-адресом по-прежнему, как и при ручном назначении, существует постоянное соответствие. Оно устанавливается в момент первичного назначения сервером DHCP IP-адреса клиенту. При всех последующих запросах сервер возвращает тот же самый IP-адрес.

При динамическом распределении адресов DHCP-сервер выдает адрес клиенту на ограниченное время, что дает возможность впоследствии повторно использовать IP-адреса другими компьютерами. Динамическое распределение адресов позволяет строить IP-сеть, количество узлов в которой намного превышает количество имеющихся в распоряжении администратора IP-адресов.

DHCP обеспечивает надежный и простой способ конфигурации сети, гарантируя отсутствие конфликтов адресов за счет централизованного управления их распределением. Администратор управляет процессом назначения адресов с помощью параметра «продолжительности аренды» (lease duration), который определяет, как долго компьютер может использовать назначенный IP-адрес, перед тем как снова запросить его от сервера DHCP в аренду.

Примером работы протокола DHCP может служить ситуация, когда компьютер, являющийся клиентом DHCP, удаляется из подсети. При этом назначенный ему IP-адрес автоматически освобождается. Когда компьютер подключается к другой подсети, то ему автоматически назначается новый адрес. Ни пользователь, ни сетевой администратор не вмешиваются в этот процесс. Это свойство очень важно для мобильных пользователей.

Протокол DHCP использует модель клиент-сервер. Во время старта системы компьютер-клиент DHCP, находящийся в состоянии «инициализация», посылает сообщение «исследовать» (discover), которое широковещательно распространяется по локальной сети и передается всем DHCP-серверам частной сети. Каждый DHCP-сервер, получивший это сообщение, отвечает на него сообщением «предложение» (offer), которое содержит IP-адрес и конфигурационную информацию.

DHCP-клиент переходит в состояние «выбор» и собирает конфигурационные предложения от DHCP-серверов. Затем он выбирает одно из этих предложений, переходит в состояние «запрос» и отправляет сообщение «запрос» (request) тому DHCP-серверу, чье предложение было выбрано.

Выбранный DHCP-сервер посылает сообщение «подтверждение» (DHCP-acknowledgment), содержащее тот же IP-адрес, который уже был послан ранее на стадии исследования, а также параметр аренды для этого адреса. Кроме того, DHCP-сервер посылает параметры сетевой конфигурации. После того, как клиент получит это подтверждение, он переходит в состояние «связь», находясь в котором, он может принимать участие в работе сети TCP/IP. Компьютеры-клиенты, которые имеют локальные диски, сохраняют полученный адрес для использования при последующих стартах системы. При приближении момента

истечения срока аренды адреса, компьютер пытается обновить параметры аренды у DHCP-сервера, а если этот IP-адрес не может быть выделен снова, то ему возвращается другой IP-адрес.

В протоколе DHCP описывается несколько типов сообщений, которые используются для обнаружения и выбора DHCP-серверов, для запросов информации о конфигурации, для продления и досрочного прекращения лицензии на IP-адрес. Все эти операции направлены на то, чтобы освободить администратора сети от утомительных рутинных операций по конфигурированию сети.

Однако использование DHCP несет в себе и некоторые проблемы. Во-первых, это проблема согласования информационной адресной базы в службах DHCP и DNS. Как известно, DNS служит для преобразования символьных имен в IP-адреса. Если IP-адреса будут динамически изменяться сервером DHCP, то эти изменения необходимо также динамически вносить в базу данных сервера DNS.

Во-вторых, нестабильность IP-адресов усложняет процесс управления сетью. Системы управления, основанные на протоколе SNMP, разработаны с расчетом на статичность IP-адресов. Аналогичные проблемы возникают и при конфигурировании фильтров маршрутизаторов, которые оперируют с IP-адресами.

Наконец, централизация процедуры назначения адресов снижает надежность системы: при отказе DHCP-сервера все его клиенты оказываются не в состоянии получить IP-адрес и другую информацию о конфигурации. Последствия такого отказа могут быть уменьшены путем использования в сети нескольких серверов DHCP, каждый из которых имеет свой пул IP-адресов.

3.7. Протокол ICMP

ICMP (Internet Control Message Protocol) – протокол управляющих сообщений.

Компьютер получает эти сообщения постоянно, а иногда и отправляет, например:

- если адрес не доступен;
- если порт не доступен;
- если используется команда **ping**, вы получаете сообщение ICMP;
- и т. д.

Сообщение ICMP инкапсулируется прямо в IP-пакет (поле данных), т. е. протоколы транспортного уровня не используются.

Протокол ICMP представляет собой механизм передачи сообщений об ошибках, которые возникают в процессе информационного обмена в сети Internet. На данный протокол не возлагаются функции локализации и устранения причин, которые привели к возникновению этих ошибок.

Сообщения делятся на два типа:

- парные (вопрос/ответ);

– непарные (например: посылаете запрос к серверу, но сервер не доступен, и последний маршрутизатор (или сервер) отправляет ICMP-сообщение (Destination Unreachable) вам).

Сообщение Time Exceeded (истекло время) – принадлежит к непарным сообщениям ICMP. Это сообщение должно быть сформировано в том случае, если в процессе передачи дейтаграммы истекло допустимое время ее существования в сети или на хосте.

Непарное сообщение формируется, если заголовок IP-дейтограммы содержит неверный параметр.

Непарное сообщение формируется, если возникла перегрузка маршрутизатора, пакет не может быть помещен в буфер, так как он переполнен.

Непарное сообщение формируется, если изменен маршрут для пакета.

Например (рис. 76), хост A(10.40.0.2) отправляет дейтаграмму в направлении хоста B(10.10.0.2), используя для этого в качестве шлюза маршрутизатор R2. После того, как маршрутизатор R2 получает дейтаграмму, он определяет, что данная дейтаграмма адресована в направлении 10.10.0.0. Кратчайший маршрут для достижения этой сети для маршрутизатора R2 лежит через маршрутизатор R4, который в данном случае подключен к тому сегменту сети, из которого была получена принятая дейтаграмма.

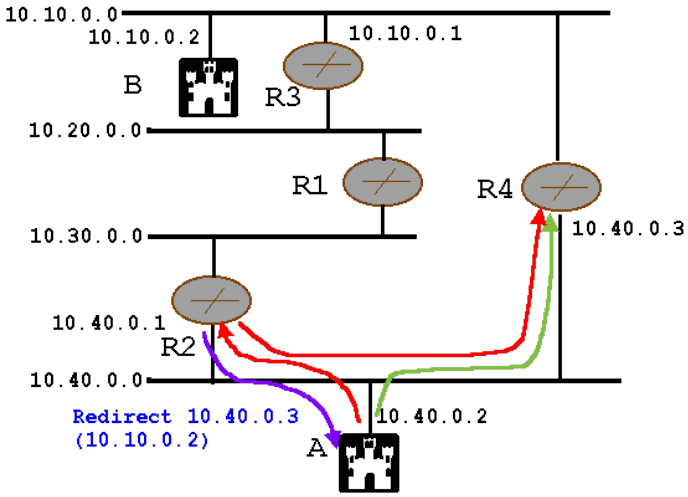


Рисунок 76 – Принцип работы ICMP

Маршрутизатор R2 направляет дейтаграмму по направлению R4 и одновременно формирует сообщение ICMP Redirect («Перенаправление»), в котором он рекомендует хосту A впредь для передачи дейтаграмм в направлении сети использовать в качестве шлюза маршрутизатор R4.

3.8. *Адреса IPv6

Все IP-адреса, о которых рассказывалось ранее, относятся к **IPv4**. Но существует и более новая версия – **IPv6**. Эти адреса обладают значительно большим адресным пространством, чем IPv4.

IP-адрес версии 6 имеет следующий формат записи:

2001:0db8:85a3:0000:0000:8a2e:0370:7334

Он состоит из **128 бит** *(вместо 32 бит у IPv4)*, т. е. из 16 байт. Байты записываются в шестнадцатеричном виде, группами по 2 байта *(т. е. 8 групп по 4 шестнадцатеричных символа)*.

Запись адреса

- Ведущие нули в каждом поле можно опустить (однако каждая группа должна иметь хотя бы один знак). Тогда, рассмотренный выше адрес, можно также записать как:

  2001:db8:85a3:0:0:8a2e:370:7334

- Самая длинная последовательность нулевых полей может заменяется двумя двоеточиями («::»). Т. е. тот же адрес, можно записать как:

  2001:db8:85a3::8a2e:370:7334

  Но двойное двоеточие можно использовать в адресе только один раз!
- Буквы в шестнадцатеричных символах могут записываться в любом регистре, но стандартом рекомендуется только нижний регистр. Браузеры, например, будут автоматически изменять регистр адресов на нижний.
- Адрес «**localhost**» (который для IPv4 записывается как 127.0.0.1) в IPv6 будет выглядеть как:

  ::1

  Или в полных версиях как:

  0:0:0:0:0:0:0:1
  0000:0000:0000:0000:0000:0000:0000:0001

- Для плавного перехода от IPv4 к IPv6 предусмотрена специальная запись, при которой IPv4-адреса могут быть записаны в IPv6, например:

  ::ffff:192.168.0.1

- При записи в адресной строке браузера IPv6-адреса записываются в квадратных скобках.

Например, если мы знаем, что по адресу 192.168.0.99 расположен наш роутер *(или другое устройство)*, то мы можем зайти на него и по IPv6-адресу. Для этого откроем браузер и введем в адресной строке «[**::ffff:192.168.0.99**]»

(рис. 77). При этом браузер автоматически пересчитает этот адрес в шестнадцатеричный формат.

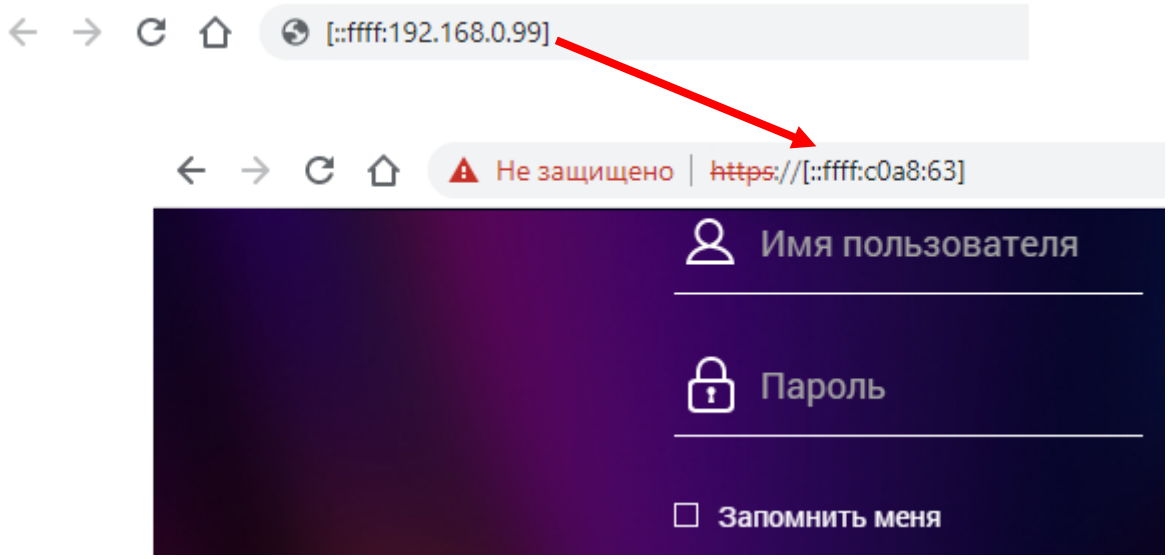


Рисунок 77 – IPv6-адрес

3.9. *Механизм NAT

NAT (Network Address Translation, «преобразование сетевых адресов») – это механизм в сетях TCP/IP, позволяющий преобразовывать IP-адреса транзитных пакетов.

Как правило, стандартный (домашний) **Роутер** имеет один «глобальный» порт **WAN** для кабельного подключения к сети Интернет, несколько «локальных» портов **LAN**, а также встроенную **Wi-Fi** точку доступа (которая также относится к LAN).

Суть механизма NAT состоит в замене адреса источника (при прохождении пакета в одну сторону и обратной замене адреса назначения в ответном пакете. Наряду с адресами источник/назначение могут также заменяться номера портов источника и назначения.

Принимая пакет от локального компьютера, роутер смотрит на IP-адрес назначения. Если это локальный адрес, то пакет пересылается другому локальному компьютеру. Если нет, то пакет надо переслать наружу в интернет. Но ведь обратным адресом в пакете указан локальный адрес компьютера, который из интернета будет недоступен. Поэтому роутер «на лету» транслирует (подменяет) обратный IP-адрес пакета на свой внешний (видимый из интернета) IP-адрес и меняет номер порта (чтобы различать ответные пакеты, адресованные разным локальным компьютерам). Комбинацию, нужную для обратной подстановки, роутер сохраняет у себя во временной таблице. Через некоторое время после того, как клиент и сервер закончат обмениваться пакетами, роутер сотрет у себя в таблице запись об этом порте за сроком давности.

NAT выполняет три важных функции:

– позволяет сэкономить IP-адреса, транслируя несколько внутренних IP-адресов в один внешний «публичный» IP-адрес. По такому принципу построено большинство сетей в мире: на небольшой район домашней сети местного провайдера или на офис выделяется 1 публичный (внешний) IP-адрес, за которым работают и получают доступ интерфейсы с приватными (внутренними) IP-адресами;

– позволяет предотвратить или ограничить обращение снаружи ко внутренним хостам, оставляя возможность обращения изнутри наружу. При инициации соединения изнутри сети создается трансляция. Ответные пакеты, поступающие снаружи, соответствуют созданной трансляции и поэтому пропускаются. Если для пакетов, поступающих снаружи, соответствующей трансляции не существует, они не пропускаются;

– позволяет скрыть определенные внутренние сервисы внутренних хостов/серверов. По сути, выполняется та же указанная выше трансляция на определенный порт, но возможно подменить внутренний порт официально зарегистрированной службы (например, 80-й порт TCP (http-сервер) на внешний 54055-й). Тем самым, снаружи, на внешнем IP-адресе после трансляции адресов на сайт для осведомленных посетителей можно будет попасть по адресу http://example.org:54055, но на внутреннем сервере, находящемся за NAT, он будет работать на обычном 80-м порту. Это повышает безопасность и обеспечивает сокрытие «непубличных» ресурсов.

3.10. *Структура IP-пакета

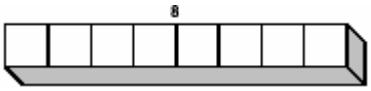
Структура IP-пакета представлена на рис. 78. Назначение полей этого пакета следующее:

- *номер версии* указывает версию протокола IP;
- *длина заголовка* пакета IP меняется в зависимости от числа параметров;
- поле *тип сервиса* задает вид обслуживания. В нем предусмотрены биты указания задержки передачи, биты производительности и надежности локальной сети. Эти биты используются для задания приоритетов при формировании маршрута, например, в качестве критерия выбора маршрута может быть задана либо длина маршрута, либо надежность, либо сбалансированность трафика;
- *общая длина* пакета с учетом заголовка и поля данных;
- поле *идентификатор пакета* используется для распознавания продублированных пакетов, а также при распознавании одинаковых пакетов, получившихся после фрагментации;
- *флаг* указывает на целесообразность фрагментации пакета, а также наличие или отсутствие последующих пакетов при фрагментации;
- поле *смещение фрагмента* используется для указания количества байтов пакета, переданных до его фрагментации;

1) Версия протокола

2) Длина заголовка

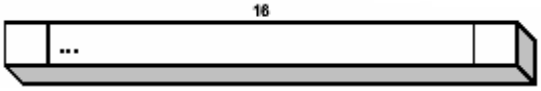
3) Тип сервиса



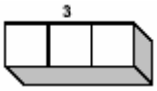
- критерий маршрутизации (длина пути, надежность, баланс трафика и т. п.)

4) Общая длина

5) Идентификатор пакета



6) Флаг

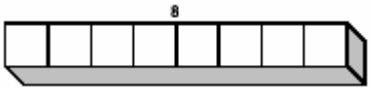


- признак фрагментации

7) Смещение фрагмента

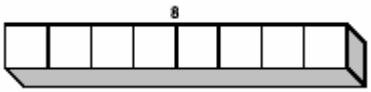
Смещение от начала всего сообщения

8) Время жизни



- оставшееся время жизни

9) Протокол верхнего уровня



10) Контрольная сумма

11) Адрес источника

12) Адрес назначения

13) Резерв

14) Данные

Рисунок 78 – Структура пакета протокола IP

- поле *время жизни* (TTL) указывает продолжительность жизни данного пакета и задается протоколом IP источника передачи. На шлюзах и в других узлах сети по истечении каждой секунды от текущего времени жизни вычитается единица. Единица вычитается также при каждой транзитной передаче (даже если не прошла секунда). При истечении срока жизни пакет аннулируется;
- идентификатор пакета *протокола верхнего уровня* указывает, какому протоколу высокого уровня принадлежит пакет;
- *контрольная сумма* рассчитывается по всему заголовку;
- *адрес источника и адрес назначения* имеют одинаковую длину 32 бита и одинаковую структуру.

4. ПРОТОКОЛЫ МАРШРУТИЗАЦИИ

EIGRP (Enhanced Interior Gateway Routing Protocol) – дистанционно векторный протокол маршрутизации [4], разработанный фирмой **Cisco** на основе протокола **IGRP** той же фирмы. Протокол IGRP был создан как альтернатива протоколу **RIP** до того, как был разработан **OSPF**.

После появления OSPF Cisco представила EIGRP – переработанный и улучшенный вариант IGRP, свободный от основного недостатка дистанционно-векторных протоколов, заключающегося в появлении особых ситуаций с зацикливанием маршрутов.

Несмотря на то, что в общем случае протоколы состояния связей (OSPF) отрабатывают изменения в топологии сети быстрее, чем EIGRP, а также OSPF имеет ряд дополнительных возможностей, но EIGRP более прост в реализации и менее требователен к вычислительным ресурсам маршрутизатора.

4.1. Протокол EIGRP

EIGRP-маршрутизатор обнаруживает своих соседей путем периодической рассылки сообщений «**Hello**». Эти же сообщения используются для мониторинга состояния связи с соседом (рассылаются каждые **5** секунд в сетях с большой пропускной способностью – например, **Ethernet**, и каждые **60** с в «медленных» сетях). Такой мониторинг позволяет рассылать в сеть вектора расстояний не периодически, а только при изменении топологии сети.

EIGRP использует комплексное значение **метрики** (цены связи), вычисляемое на основании показателей пропускной способности и задержки при передаче данных в сети. *Также в расчет метрики могут быть включены показатели загрузки и надежности сети.* В отличие от протокола RIP, метрика в EIGRP не является фактором, ограничивающим размер системы.

При получении от соседей векторов расстояний, маршрутизатор для каждой сети назначения выбирает соседа, через которого лежит кратчайший путь в эту сеть, а также запоминает «**вероятных заместителей**».

Вероятным заместителем становится маршрутизатор, объявивший метрику маршрута от себя до данной сети меньшую, чем полная метрика установленного маршрута.

Рассмотрим пример на рис. 79.

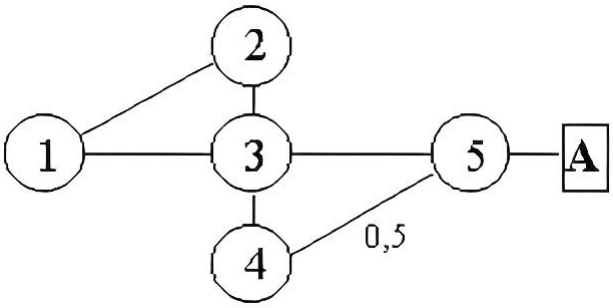


Рисунок 79 – Пример EIGRP-системы

Кружками обозначены маршрутизаторы, прямоугольником – сеть назначения А. Для простоты, метрики всех связей, кроме (4) - (5), считаем равными единице; метрика связи (4) - (5) равна 0,5.

Маршрутизатор (3) получает от (5) элемент вектора расстояний (А=1), а от (4) – (А=1,5). В таблице маршрутизатора (3) узел (5) становится следующим маршрутизатором на пути в сеть А, а узел (4) – вероятным заместителем, так как заявленное им расстояние до А (1,5) меньше полной метрики установленного маршрута (3) - (5) - А, которая равна 2.

Обратим внимание на маршрутизаторы (1) и (2). Они присылают узлу (3) элемент (А=3) и, следовательно, не являются вероятными заместителями маршрута из (3) в А.

Если связь между узлами (3) и (5) обрывается, то (3) ищет в своей EIGRP-таблице вероятного заместителя (4) и немедленно устанавливает маршрут в сеть А через него. *Таким образом, время, в течение которого маршрут в сеть А отсутствовал, существенно сокращается по сравнению с протоколами, где требуется ждать, когда соседи пришлют очередные вектора расстояний.*

Если же ни одного вероятного заместителя не найдено (допустим, связь (3) - (4) тоже обрывается), то маршрутизатор переходит в активное состояние и начинает опрос всех своих соседей на предмет наличия маршрута в сеть А, сообщая при этом, что его собственное расстояние до А равно бесконечности. Сосед отвечает на запрос только тогда, когда у него есть либо готовый маршрут в А, либо вероятный заместитель. В любом из этих случаев сосед присылает в узел (3) свое расстояние до А. Иначе сосед сам переходит в активное состояние и процесс повторяется *(разумеется, с той разницей, что к маршрутизатору (3) запрос не посылается)*. Таким образом, область «активизированных» маршрутизаторов расширяется до тех пор, пока не будет обнаружен маршрут в сеть А или доказано его отсутствие. После чего волна сходится в обратном направлении к инициировавшему процесс узлу, при этом все маршрутизаторы вносят в свои таблицы надлежащие изменения.

В этом простом примере, после того как (3) переходит в активное состояние, узлы (1) и (2) получают от него запрос о маршруте в сеть А с пометкой, что расстояние от (3) до А теперь равно бесконечности. Каждый из них, поскольку ранее он добирался в А через (3), помечает этот маршрут как

недостижимый и, не найдя вероятного заместителя, активизируется и опрашивает своего соседа. Получив эти запросы, (1) и (2) отвечают друг другу, что сеть А недостижима, переходят в пассивное состояние и возвращают узлу (3) информацию о недостижимости сети А.

4.2. Протокол RIP

Протокол **RIP** является дистанционно-векторным протоколом внутренней маршрутизации [4]. Процесс работы протокола состоит в рассылке, получении и обработке векторов расстояний до IP-сетей, находящихся в области действия протокола *(т. е. в данной RIP-системе)*. Результатом работы протокола на конкретном маршрутизаторе является таблица, где для каждой сети данной RIP-системы указано расстояние до этой сети (в **хопах**) и адрес следующего маршрутизатора.

Алгоритм построения таблицы маршрутов

Для простоты будем называть таблицей маршрутов таблицу, состоящую из строк с полями «Сеть», «Расстояние», «Следующий маршрутизатор». Записывать строку в таблице маршрутов будем следующим образом:

A=2→③

Читается как «А равно 2 через третий». *Это означает, что расстояние от данного маршрутизатора до сети А равно 2, а дейтаграммы, следующие в сеть А, надо пересылать маршрутизатору (3).*

Вектором расстояний называется пара («Сеть», «Расстояние до этой сети»), извлеченная из таблицы маршрутов. Каждую такую пару называют элементом вектора расстояний. Будем записывать вектор расстояний в виде (A=2, B=1, C=…). Это означает, что расстояние от данного маршрутизатора до сети А равно 2, до сети В равно 1 и т. д.

Расстояние до сети, к которой маршрутизатор подключен непосредственно, примем равным 1.

Каждый маршрутизатор периодически широковещательно распространяет свой вектор расстояний. Вектор распространяется через все интерфейсы маршрутизатора.

Каждый маршрутизатор также периодически получает вектора расстояний от других маршрутизаторов. Расстояния в этих векторах увеличиваются на 1, после чего сравниваются с данными в таблице маршрутов, и, если расстояние до какой-то из сетей в полученном векторе оказывается меньше расстояния, указанного в таблице, значение из таблицы замещается новым (меньшим) значением. Адрес маршрутизатора, приславшего этот вектор, записывается в поле «Следующий маршрутизатор» (в этой строке таблицы). После этого вектор расстояний, рассылаемый данным маршрутизатором, соответственно изменится.

4.3. ПРАКТИЧЕСКАЯ РАБОТА
Пример построения таблицы маршрутов

Рассмотрим этот процесс на примере следующей сети (рис. 80).

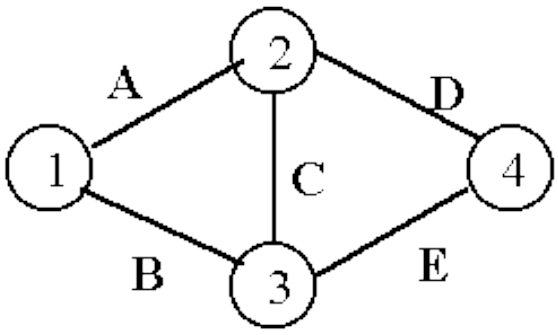


Рисунок 80 – Пример RIP-системы

Здесь ①, ②, ③, ④ – маршрутизаторы, A, B, C, D, E – сети. Хосты в сетях не показаны за ненадобностью. Мы будем следить за формированием таблицы маршрутов в узле ①.

В начальный момент времени (например, после подачи питания на маршрутизаторы) таблица маршрутов в узле ① выглядит следующим образом (так как узел ① знает только о тех сетях, к которым подключен непосредственно):

A=1→①
B=1→①

Следовательно, узел ① рассылает в сети A и B вектор расстояний (A=1,B=1).

Аналогично узел ② рассылает в сети A, C, D вектор (A=1,C=1,D=1). Узел ① получает этот вектор из сети A, увеличивает расстояния на 1 (A=2, C=2, D=2) и сравнивает с данными в своей таблице маршрутов. Новое расстояние до сети A оказывается больше, чем уже внесенное в таблицу (A=1), следовательно, новое значение игнорируется. Поскольку сети C и D вовсе не фигурируют в его таблице маршрутов, они туда вносятся. В узле ① получаем:

A=1→①
B=1→①
C=2→②
D=2→②

Узел ④, в свою очередь, рассылает вектор (D=1, E=1) в сети D и E. Узел ② получает этот вектор из сети D, увеличивает расстояния на 1, после чего добавляет себе в таблицу данные о сети E (E=2→②). Ранее из узла ① он получил информацию о сети B и добавил себе в таблицу строку B=2→①. Узел ②

рассылает в сети A, C, D свой обновленный вектор расстояний (A=1, B=2, C=1, D=1, E=2).

Узел ① получает этот вектор от ② из сети A, увеличивает расстояния на 1 (A=2, B=3, C=2, D=2, E=3) и замечает, что все указанные расстояния, кроме расстояния до сети E, больше либо равны значениям, имеющимся в его таблице. Сеть E в таблице узла ① отсутствует, следовательно, она туда вносится, и в узле ① мы получаем:

A=1→①
B=1→①
C=2→②
D=2→②
E=3→②

Далее маршрутизатор ③ (ранее не работавший по каким-либо причинам) рассылает в сети B, C, E свой вектор (B=1, C=1, E=1). Узел ① получает этот вектор из сети B, увеличивает расстояния на 1 и обнаруживает, что расстояние E=2 меньше имеющегося в таблице E=3, следовательно, запись о сети E в таблице заменяется на E=2→③. Остальные элементы, полученного от ③ вектора, не вызывают обновление таблицы.

Итоговая таблица маршрутов маршрутизатора ①:

A=1→①
B=1→①
C=2→②
D=2→②
E=2→③

На этом алгоритм сходится, т. е. при неизменной топологии системы, никакие вектора расстояний, получаемые маршрутизатором ①, больше не внесут изменений в таблицу маршрутов. Аналогичным образом алгоритм работает и сходится и на других маршрутизаторах.

Отметим, что несмотря на то, что таблицы маршрутов построены, вектора расстояний продолжают периодически широковещательно рассылаться каждым маршрутизатором. Это требуется для оперативного реагирования на внезапные изменения топологии системы.

Очевидно, что вид построенной таблицы маршрутов может зависеть от порядка получения маршрутизатором векторов расстояний. Например, если бы узел ① получил вектор от узла ③ раньше, чем от узла ②, то дейтаграммы в сеть C посылались бы от ① через ③.

Изменение состояния RIP-системы

Выясним, что происходит в случае, когда состояние системы неожиданно изменяется, например, маршрутизатор ① отключается от сети A (рис. 81).

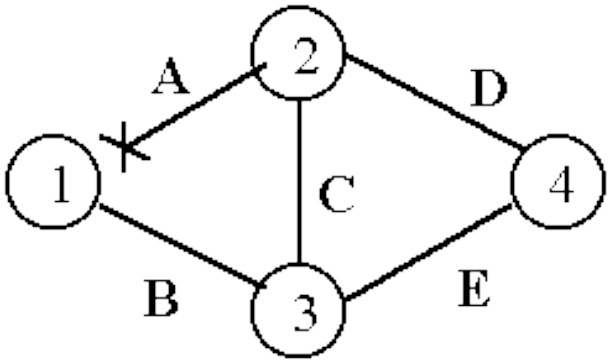


Рисунок 81 – Изменение состояния RIP-системы

Узел ① обнаруживает свое отсоединение от сети A и меняет таблицу маршрутов, устанавливая бесконечное расстояние до всех сетей, ранее достижимых через маршрутизаторы, подключенные к сети A. В протоколе RIP значение бесконечности равно 16.

A=16→①
B=1→①
C=16→②
D=16→②
E=2→③

Вектор расстояний, построенный на основании этой таблицы, рассылается в сеть B, чтобы маршрутизаторы, направлявшие свои данные через ① в ставшие недоступными сети, соответственно изменили свои маршрутные таблицы.

Допустим, в узле ③ имелась следующая таблица маршрутов:

A=2→②
B=1→③
C=1→③
D=2→④
E=1→③

Узел ③ периодически и широковещательно рассылает в сети B, C, E свой вектор расстояний (A=2, B=1, C=1, D=2, E=1). Узел ① получает этот вектор, увеличивает расстояния на 1 (A=3, B=2, C=2, D=3, E=2) и замечает, что расстояния A=3, C=2 и D=3 меньше бесконечности, следовательно, соответствующие записи таблицы маршрутов модифицируются, и она принимает вид:

A=3→③
B=1→①
C=2→③
D=3→③
E=2→③

Таким образом, узел ① построил маршруты в обход поврежденного участка и восстановил достижимость всех сетей.

***Особые случаи. Зацикливание**

К сожалению, поведение дистанционно-векторных протоколов (и в частности, протокола RIP) при изменении топологии системы не всегда корректно и предсказуемо.

Рассмотрим вышеописанную ситуацию с отсоединением узла ① от сети A (см. рис. 81).

Мы предполагали, что узел ③ не отправлял дейтаграмм через узел ① *(и, следовательно, изменение таблицы маршрутов в узле ① не повлияло на таблицу узла ③)*. Предположим теперь, что ③ отправлял дейтаграммы в сеть A через ①, то есть таблица в узле ③ имела вид:

A=2→①
B=1→③
C=1→③
D=2→④
E=1→③

После отсоединения ① от сети A узел ③ получает от ① вектор (A=16, B=1, C=16, D=16, E=2). Проанализировав этот вектор, ③ делает вывод, что все указанные в нем расстояния больше значений, содержащихся в его маршрутной таблице, на основании чего этот вектор узлом ③ игнорируется.

В свою очередь, узел ③ рассылает в сети B, C, E вектор (A=2, B=1, C=1, D=2, E=1). Узел ① получает этот вектор, увеличивает расстояния на 1 (A=3, B=2, C=2, D=3, E=2) и замечает, что расстояния A=3, C=2 и D=3 меньше бесконечности, следовательно, соответствующие записи таблицы маршрутов в узле ① модифицируются, и она принимает вид:

A=3→③
B=1→①
C=2→③
D=3→③
E=2→③

Очевидно, после этого содержимое таблиц узлов ① и ③ стабилизируется.

Рассмотрим теперь записи о достижении сети A в таблицах маршрутизаторов ① и ③.

В узле ①: A=3→③
В узле ③: A=2→①

Таким образом, возникло зацикливание – данные, адресованные в сеть A, будут пересылаться между узлами ① и ③ до тех пор, пока не истечет время жизни дейтаграмм и они не будут уничтожены.

Для того чтобы избежать зацикливания, в алгоритм рассылки векторов расстояний вносятся дополнения. Тем не менее, и в этом случае особые ситуации все еще остаются.

4.4. Протокол OSPF

Протокол маршрутизации **OSPF** (Open SPF) представляет собой протокол состояния связей, использующий алгоритм **SPF** поиска кратчайшего пути в графе [4]. OSPF применяется для внутренней маршрутизации в системах сетей любой сложности.

Построение маршрутов

Рассмотрим работу алгоритма SPF и построение маршрутов на примере системы, изображенной на рис. 82. Для простоты будем рассматривать OSPF-систему, состоящую только из маршрутизаторов, соединенных линиями связи типа «точка-точка».

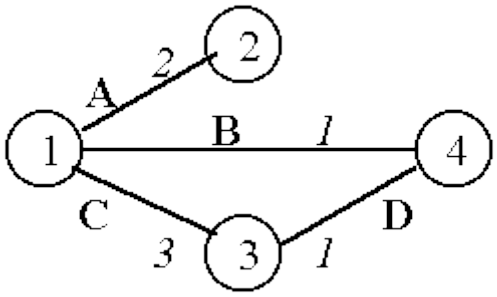


Рисунок 82 – Пример OSPF-системы

①, ②, ③, ④ – маршрутизаторы; A, B, C, D – линии связи (или просто связи), цифры означают метрику каждой связи.

Метрика

Метрика представляет собой оценку качества связи в данной сети *(на данном физическом канале)*. Чем меньше метрика, тем лучше качество соединения. Метрика маршрута равна сумме метрик всех связей (сетей), входящих в маршрут. В простейшем случае (как это имеет место в протоколе RIP) метрика каждой сети равна единице, а метрика маршрута тогда просто является его длиной в хопах.

Для OSPF метрика сети, оценивающая пропускную способность, определяется как количество секунд, требуемое для передачи 100 Мбит. Например, метрика сети на базе 10Base-T Ethernet равна 10, а метрика выделенной линии 56 кбит/с равна 1785. Метрика канала со скоростью передачи данных 100 Мбит/с и выше равна единице.

Порядок расчета метрик, оценивающих надежность, задержку и стоимость, не определен. Администратор, желающий поддерживать маршрутизацию по этим типам сервисов, должен сам назначить разумные и согласованные метрики по этим параметрам.

База данных состояния связей

Для работы алгоритма SPF на каждом маршрутизаторе строится база данных состояния связей. Базы данных на всех маршрутизаторах идентичны.

База данных состояния связей для рассмотренного примера приведена в табл. 3.

Таблица 3 – База данных состояния связей

От → До	**Сеть**	**Метрика**
①→②	A	2
①→③	C	3
①→④	B	1
③→④	D	1
②→①	A	2
③→①	C	3
④→①	B	1
④→③	D	1

4.5. *Таблица маршрутизации Windows

Маршрутизатор (Router) – работает на сетевом уровне, определяя путь (маршрут) для пересылки проходящих через него пакетов. Может выполнять фильтрацию пакетов (решение о возможности передачи пакета принимается на основе анализа его содержимого). Маршрутизатором может служить специализированное оборудование или обычный компьютер. Каждый из интерфейсов маршрутизатора имеет собственный IP-адрес.

Пример таблицы маршрутизации в ОС Windows представлен в табл. 4.

Таблица 4 – Пример таблицы маршрутизации в ОС Windows

Сетевой адрес	Маска сети	Адрес шлюза	Интерфейс	Метрика
0.0.0.0	0.0.0.0	192.168.0.1	192.168.0.10	1
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.0.0	255.255.255.0	192.168.0.10	192.168.0.10	1
192.168.0.1	255.255.255.255	127.0.0.1	127.0.0.1	1
192.168.0.255	255.255.255.255	192.168.0.10	192.168.0.10	1
224.0.0.0	224.0.0.0	192.168.0.10	192.168.0.1	1
255.255.255.255	255.255.255.255	192.168.0.10	192.168.0.10	1

Таблицы маршрутизации содержат записи, состоящие из следующих полей:

- адрес или множество адресов назначения (адрес и маска сети);
- адрес шлюза, которому передается пакет, адресованный в указанное множество;
- интерфейс маршрутизатора, который связывает его со шлюзом;
- метрика маршрута – число, определяющее его качество (расстояние до адресата, пропускная способность, надежность).

Множества адресов назначения в таблице маршрутизации могут содержаться одно в другом (в частности, множество, определенное в примере выше первой строкой, включает все остальные возможные множества и определяет маршрут по умолчанию). Если адрес входит в несколько множеств, для определения маршрута пересылки используется множество минимальной мощности (наименьшее).

Для просмотра и настройки таблицы в ОС Windows используется команда ROUTE:

- **ROUTE PRINT** – для отображения маршрута;
- **ROUTE ADD** – для добавления маршрута;
- **ROUTE DELETE** – для удаления маршрута;
- **ROUTE CHANGE** – для изменения существующего маршрута.

5. МОДЕЛЬ ВЗАИМОДЕЙСТВИЯ ОТКРЫТЫХ СИСТЕМ

С появлением вычислительных сетей возникла необходимость в разработке концепции, устанавливающей стандартные правила взаимодействия разнотипных станций (узлов), входящих в состав сети. Для этого была создана многоуровневая архитектура с выделением 7 уровней иерархии. Разработчики – ряд международных организаций по стандартизации **ISO**, **IEEE** и др. назвали созданную ими в начале 80-х гг. структуру *моделью взаимодействия открытых систем* (Open Systems Interconnection, OSI). Согласно модели OSI, функции взаимодействия объектов в сети разделены на следующие уровни: физический, канальный, сетевой, транспортный, сеансовый, представительный и прикладной.

Модель OSI описывает только системные функции, реализуемые операционной системой, системными сетевыми компонентами, системными аппаратными средствами, и не включает средства взаимодействия приложений конечных пользователей.

Приложения конечных пользователей взаимодействуют, обращаясь к системным средствам.

Поскольку в процессе обмена сообщениями в сети участвуют две стороны, то необходимо организовать согласованную работу двух «иерархий», работающих на разных узлах. Оба участника сетевого обмена должны принять множество соглашений. Например, они должны согласовать уровни и форму электрических сигналов, способ определения размера сообщений, договориться о методах контроля достоверности и т. п. Другими словами, соглашения должны быть приняты для всех уровней, начиная от самого низкого – уровня передачи битов – до самого высокого, реализующего сервис для пользователей сети.

Иерархически организованный набор протоколов, достаточный для организации взаимодействия узлов в сети, называется **стеком** коммуникационных протоколов.

Коммуникационные протоколы могут быть реализованы как программно, так и аппаратно. Протоколы нижних уровней часто реализуются комбинацией программных и аппаратных средств, а протоколы верхних уровней, как правило, чисто программными средствами.

Протоколы реализуются различными сетевыми устройствами – концентраторами, мостами, коммутаторами, маршрутизаторами и пр. Действительно, в общем случае связь узлов в сети осуществляется не напрямую, а через различные коммуникационные устройства.

5.1. Уровни модели OSI

Модель OSI состоит из семи уровней.

Физический уровень

Физический уровень (**Physical layer**) имеет дело с передачей битов по физическим каналам связи, таким как коаксиальный кабель, витая пара,

оптоволоконный кабель или радиоканал. К этому уровню имеют отношение характеристики физической среды передачи данных и характеристики электрических сигналов, такие, как:

- полоса пропускания;
- помехозащищенность;
- волновое сопротивление;
- крутизна фронтов импульсов;
- уровни напряжения или тока передаваемого сигнала;
- тип кодирования;
- скорость передачи сигналов;
- и др.

Кроме этого, здесь стандартизуются типы разъемов и назначение каждого контакта [5].

Функции физического уровня реализуются во всех устройствах, подключенных к сети, и выполняются сетевым адаптером или последовательным портом. Примером протокола физического уровня может служить спецификация 10Base-T технологии Ethernet, которая определяет в качестве используемого кабеля неэкранированную витую пару категории 3 с волновым сопротивлением 100 Ом, разъем RJ-45, максимальную длину физического сегмента 100 м, манчестерский код для представления данных в кабеле, а также некоторые другие характеристики среды и электрических сигналов.

Канальный уровень

На физическом уровне просто пересылаются биты. При этом не учитывается, что в некоторых сетях, в которых линии связи используются (разделяются) попеременно несколькими парами взаимодействующих станций, физическая среда передачи может быть занята.

Одной из задач канального уровня (**Data Link layer**) является проверка доступности среды передачи. Другой задачей канального уровня является реализация механизмов обнаружения и коррекции ошибок. Для этого на канальном уровне биты группируются в наборы, называемые кадрами (frames). *Канальный уровень обеспечивает корректность передачи каждого кадра, помещая специальную последовательность битов в начало и конец каждого кадра для его выделения, а также вычисляет контрольную сумму, обрабатывая все байты кадра определенным способом и добавляя контрольную сумму к кадру. Когда кадр приходит по сети, получатель снова вычисляет контрольную сумму полученных данных и сравнивает результат с контрольной суммой из кадра. Если они совпадают, кадр считается правильным и принимается. Если же контрольные суммы не совпадают, то фиксируется ошибка.*

Канальный уровень может не только обнаруживать ошибки, но и исправлять их за счет повторной передачи поврежденных кадров. *Необходимо отметить, что функция исправления ошибок не является обязательной для канального уровня, поэтому в некоторых протоколах этого уровня она отсутствует, например, в Ethernet [5].*

Хотя канальный уровень и обеспечивает доставку кадра между любыми двумя узлами локальной сети, он это делает только в сети с совершенно определенной топологией связей, именно той топологией, для которой он был разработан. К таким типовым топологиям, поддерживаемым протоколами канального уровня локальных сетей, относятся общая шина, кольцо и звезда, а также структуры, полученные из них с помощью мостов и коммутаторов.

Примерами протоколов канального уровня являются протоколы Ethernet, Token Ring, FDDI. В локальных сетях протоколы канального уровня используются компьютерами, мостами, коммутаторами и маршрутизаторами. В компьютерах функции канального уровня реализуются совместными усилиями сетевых адаптеров и их драйверов.

Для обеспечения качественной транспортировки сообщений в сетях любых топологий и технологий функций канального уровня оказывается недостаточно, поэтому в модели OSI решение этой задачи возлагается на два следующих уровня – сетевой и транспортный.

Канальный уровень обеспечивает передачу пакетов данных, поступающих от протоколов верхних уровней, узлу назначения, адрес которого также указывает протокол верхнего уровня. Протоколы канального уровня оформляют переданные им пакеты в кадры собственного формата, помещая указанный адрес назначения в одно из полей такого кадра, а также сопровождая кадр контрольной суммой. Протокол канального уровня имеет локальный смысл, он предназначен для доставки кадров данных, как правило, в пределах сетей с простой топологией связей и однотипной или близкой технологией, например, в односегментных сетях Ethernet или же в многосегментных сетях Ethernet и Token Ring иерархической топологии, разделенных только мостами и коммутаторами. Во всех этих сетях адрес назначения имеет локальный смысл для данной сети и не изменяется при прохождении кадра от узла источника к узлу назначения.

Сетевой уровень

Сетевой уровень (**Network layer**) служит для образования единой транспортной системы, объединяющей несколько сетей, причем эти сети могут использовать совершенно разные принципы передачи сообщений между конечными узлами и обладать произвольной структурой связей. *Функции сетевого уровня достаточно разнообразны. Протоколы канального уровня обеспечивают доставку данных между любыми узлами только в сети с соответствующей типовой топологией (например, топологией звезды). Это очень жесткое ограничение, которое не позволяет строить сети с развитой структурой, например, сети, объединяющие несколько сетей предприятия в единую сеть, или высоконадежные сети, в которых существуют избыточные связи между узлами. Чтобы, с одной стороны, сохранить простоту процедур передачи данных для типовых топологий, а с другой – допустить использование произвольных топологий, вводится дополнительный сетевой уровень [5].*

Внутри сети доставка данных обеспечивается соответствующим канальным уровнем, а доставкой данных между сетями занимается сетевой

уровень, который и поддерживает возможность правильного выбора маршрута передачи сообщения даже в том случае, когда характер структуры связей между составляющими сетями отличается от принятого в протоколах канального уровня.

Сети соединяются между собой специальными устройствами, называемыми маршрутизаторами. **Маршрутизатор** – это устройство, которое собирает информацию о топологии межсетевых соединений и на ее основании пересылает пакеты сетевого уровня в сеть назначения. *Чтобы передать сообщение от отправителя, находящегося в одной сети, получателю, находящемуся в другой сети, нужно совершить некоторое количество транзитных передач между сетями, каждый раз выбирая подходящий маршрут. Таким образом, маршрут представляет собой последовательность маршрутизаторов, через которые проходит пакет.* **Проблема выбора наилучшего пути называется маршрутизацией, и ее решение является одной из главных задач сетевого уровня.** *Эта проблема осложняется тем, что самый короткий путь не всегда самый лучший. Часто критерием при выборе маршрута является время передачи данных по этому маршруту; оно зависит от пропускной способности каналов связи и интенсивности трафика, которая может изменяться с течением времени. Некоторые алгоритмы маршрутизации пытаются приспособиться к изменению нагрузки, в то время как другие принимают решения на основе средних показателей за длительное время. Выбор маршрута может осуществляться и по другим критериям, например, надежности передачи. В общем случае функции сетевого уровня шире, чем функции передачи сообщений по связям с нестандартной структурой. Сетевой уровень решает также задачи согласования разных технологий, упрощения адресации в крупных сетях и создания надежных и гибких барьеров на пути нежелательного трафика между сетями.*

Сообщения сетевого уровня принято называть **пакетами** (packet). При организации доставки пакетов на сетевом уровне используется понятие «номер сети». В этом случае адрес получателя состоит из старшей части – номера сети и младшей – номера узла в этой сети. Все узлы одной сети должны иметь одну и ту же старшую часть адреса, поэтому термину «сеть» на сетевом уровне можно дать и другое определение: сеть – это совокупность узлов, сетевой адрес которых содержит один и тот же номер сети.

На сетевом уровне определяются два вида протоколов:

- **сетевые протоколы** – реализуют продвижение пакетов через сеть. Именно эти протоколы обычно имеют в виду, когда говорят о протоколах сетевого уровня;
- **протоколы маршрутизации.** С помощью этих протоколов маршрутизаторы собирают информацию о топологии межсетевых соединений. Протоколы сетевого уровня реализуются программными модулями операционной системы, а также программными и аппаратными средствами маршрутизаторов.

На сетевом уровне работают протоколы еще одного типа, которые отвечают за отображение адреса узла, используемого на сетевом уровне, в локальный адрес сети. Такие протоколы часто называют ***протоколами разрешения адресов.*** *Иногда их относят не к сетевому уровню, а к канальному.*

Примерами протоколов сетевого уровня являются протокол межсетевого взаимодействия **IP** стека TCP /IP и протокол межсетевого обмена пакетами **IPX** стека Novell.

Транспортный уровень

На пути от отправителя к получателю пакеты могут быть искажены или утеряны. *Хотя некоторые приложения имеют собственные средства обработки ошибок, существуют и такие, которые предпочитают сразу иметь дело с надежным соединением.*

Транспортный уровень (**Transport layer**) обеспечивает приложениям или верхним уровням стека (прикладному и сеансовому) передачу данных с той степенью надежности, которая им требуется. *Модель OSI определяет пять классов сервиса, предоставляемых транспортным уровнем. Эти виды сервиса отличаются качеством предоставляемых услуг: срочностью, возможностью восстановления прерванной связи, наличием средств мультиплексирования нескольких соединений между различными прикладными протоколами через общий транспортный протокол.* А также способен обнаруживать и исправлять ошибки передачи, такие как искажение, потеря и дублирование пакетов [5].

Выбор класса сервиса транспортного уровня определяется, с одной стороны, тем, в какой степени задача обеспечения надежности решается самими приложениями и протоколами более высоких, чем транспортный, уровней, а с другой стороны, этот выбор зависит от того, насколько надежной является система транспортировки данных в сети, обеспечиваемая уровнями, расположенными ниже транспортного (сетевым, канальным и физическим). Так, например, если качество каналов передачи связи очень высокое и вероятность возникновения ошибок, не обнаруженных протоколами более низких уровней, невелика, то разумно воспользоваться одним из облегченных сервисов транспортного уровня, не обремененных многочисленными проверками, квитированием и другими приемами повышения надежности. Если же транспортные средства нижних уровней изначально ненадежны, то целесообразно обратиться к наиболее развитому сервису транспортного уровня, который работает, используя максимум средств для обнаружения и устранения ошибок, включая предварительное установление логического соединения, контроль доставки сообщений по контрольным суммам циклической нумерации пакетов, установление тайм-аутов доставки и т. п.

Сетевой транспорт

Протоколы нижних четырех уровней обобщенно называют сетевым транспортом или транспортной подсистемой (*так как они полностью решают задачу транспортировки сообщений с заданным уровнем качества в составных сетях с произвольной топологией и различными технологиями).* Оставшиеся три

верхних уровня решают задачи предоставления прикладных сервисов на основании имеющейся транспортной подсистемы.

Сеансовый уровень

Сеансовый уровень обеспечивает управление взаимодействием станций в сети: фиксирует, какая из сторон является активной в настоящий момент, предоставляет средства синхронизации. Последние позволяют вставлять контрольные точки в длинные передачи, чтобы в случае отказа можно было вернуться назад к последней контрольной точке, а не начинать все с начала.

На практике немногие приложения используют сеансовый уровень (**Session layer**), и он редко реализуется в виде отдельных протоколов, хотя функции этого уровня часто объединяют с функциями прикладного уровня и реализуют в одном протоколе [5].

Представительный уровень

Представительный уровень (**Presentation layer**) имеет дело с формой представления передаваемой по сети информации, не меняя при этом ее содержания. *За счет уровня представления информация, передаваемая прикладным уровнем одной системы, всегда понятна прикладному уровню другой системы. С помощью средств данного уровня протоколы прикладных уровней могут преодолеть синтаксические различия в представлении данных или же различия в кодах символов, например, кодов ASCII.*

На этом уровне может выполняться шифрование и дешифрование данных, благодаря которым секретность обмена данными обеспечивается сразу для всех прикладных служб. Примером такого протокола является протокол **SSL** (Secure Socket Layer), который обеспечивает секретный обмен сообщениями для протоколов прикладного уровня.

Прикладной уровень

Прикладной уровень (**Application layer**) – это просто набор разнообразных протоколов, с помощью которых пользователи сети получают доступ к различным ресурсам, таким как файлы, принтеры, web-страницы и т. д.

Единица данных, которой оперирует прикладной уровень, называется **сообщением** (message).

Сетезависимые и сетенезависимые уровни

Функции уровней модели OSI могут быть отнесены к одной из двух групп: либо к функциям, зависящим от конкретной технической реализации сети, либо к функциям, ориентированным на работу с приложениями.

Три нижних уровня (физический, канальный и сетевой) – являются сетезависимыми, т. е. протоколы этих уровней тесно связаны с технической реализацией сети и используемым коммуникационным оборудованием. *Например, переход на оборудование FDDI означает полную смену протоколов физического и канального уровней во всех узлах сети.*

Три верхних уровня (прикладной, представительный и сеансовый) – ориентированы на приложения и мало зависят от технических особенностей

построения сети. На протоколы этих уровней не влияют какие бы то ни было изменения в топологии сети: замена оборудования или переход на другую сетевую технологию.

Транспортный уровень является промежуточным, он скрывает детали функционирования нижних уровней от верхних.

5.2. Стек протоколов TCP/IP

TCP/IP – собирательное название для набора (стека) сетевых протоколов разных уровней, используемых в Интернет.

Особенности TCP/IP:

- открытые стандарты протоколов, разрабатываемые независимо от программного и аппаратного обеспечения;
- независимость от физической среды передачи;
- система уникальной адресации;
- стандартизованные протоколы высокого уровня для распространенных пользовательских сервисов.

Стек протоколов TCP/IP (рис. 83) делится на 4 уровня: **Прикладной** (Application, Уровень приложений), **Транспортный** (Transport), **Межсетевой** (Internet) и **Уровень доступа к среде передачи** (Network access). Термины, применяемые для обозначения блока передаваемых данных, различны при использовании разных протоколов транспортного уровня – TCP и UDP, поэтому на рис. 83 изображено два стека. Как и в модели OSI, данные более верхних уровней инкапсулируются в пакеты нижних уровней (рис. 84).

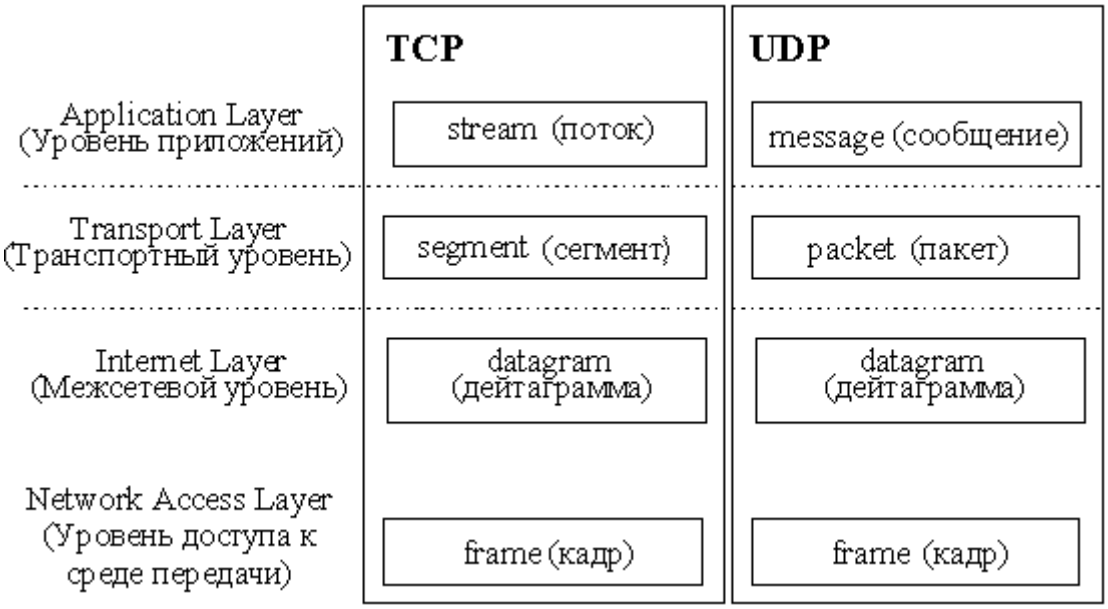


Рисунок 83 – Стек протоколов TCP/IP

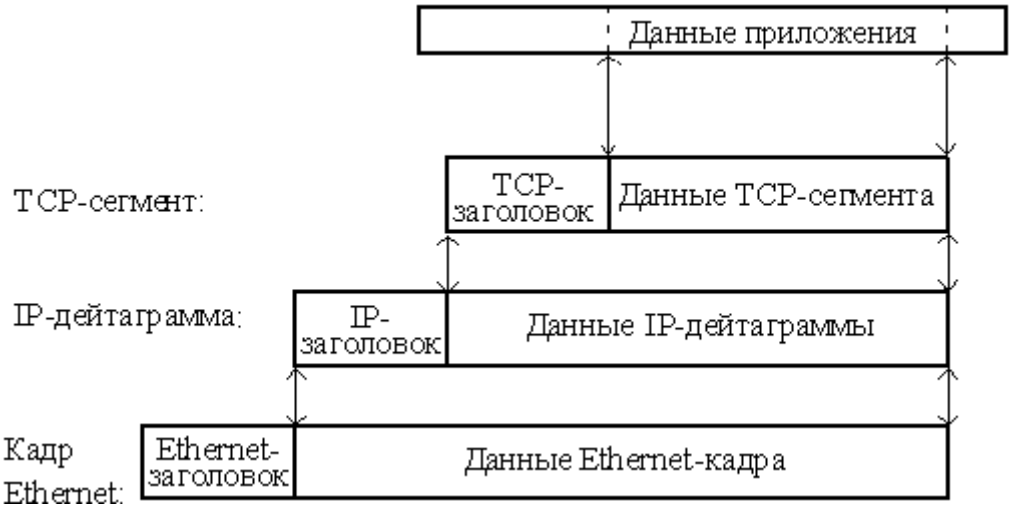


Рисунок 84 – Пример инкапсуляции пакетов в стеке TCP/IP

5.3. Сравнение стеков OSI и TCP/IP

Примерное соотношение уровней стеков OSI и TCP/IP показано на рис. 85. Ниже кратко рассматриваются функции каждого из уровней.

Рисунок 85 – Соотношение уровней стеков OSI и TCP/IP

Прикладной уровень (уровень приложений)

Примерами приложений являются HTTP-серверы и HTTP-клиенты (web-браузеры), программы работы с электронной почтой.

Для пересылки данных другому приложению приложение обращается к тому или иному модулю транспортного уровня.

Приложения, работающие со стеком TCP/IP, могут также выполнять функции уровней представления и частично сеансового модели OSI; например, преобразование данных к внешнему представлению, группировка данных для передачи и т. п.

Транспортный уровень

Протоколы транспортного уровня обеспечивают прозрачную (сквозную) доставку данных между двумя прикладными процессами. Процесс, получающий или отправляющий данные с помощью транспортного уровня, идентифицируется на этом уровне числом, которое называется номером порта. Таким образом, роль адреса отправителя и получателя на транспортном уровне выполняет номер порта (или кратко «порт»).

Номер порта – это число от 0 до 65535. Все порты разделены на три диапазона:

- порты 0–1023 – общеизвестные (или системные);
- порты 1024–49151 – зарегистрированные (или пользовательские);
- 49152–65535 – динамически выделяемые (или частные).

К общеизвестным относятся, например:

- **HTTP** – порт 80;
- **HTTPS** – порт 443;
- **FTP** – порт 21;
- **DNS** – порт 53.

Анализируя заголовок своего пакета, полученного от межсетевого уровня, транспортный модуль определяет по номеру порта получателя, какому из прикладных процессов направлены данные, и передает эти данные соответствующему прикладному процессу (возможно, после проверки их на наличие ошибок и т.п.). Номера портов получателя и отправителя записываются в заголовок транспортным модулем, отправляющим данные; заголовок транспортного уровня содержит также и другую служебную информацию. Формат заголовка зависит от используемого транспортного протокола.

На транспортном уровне работают два основных протокола: UDP и TCP.

Межсетевой уровень

Основным протоколом этого уровня является протокол IP (Internet Protocol).

Протокол IP доставляет блоки данных, называемые дейтаграммами, от одного IP-адреса к другому. *IP-адрес является уникальным 32-битным идентификатором компьютера (точнее, его сетевого интерфейса). Данные для дейтаграммы передаются IP-модулю транспортным уровнем. IP-модуль предваряет эти данные заголовком, содержащим IP-адреса отправителя и получателя и другую служебную информацию, и сформированная таким образом дейтаграмма передается на уровень доступа к среде передачи (например, одному из физических интерфейсов) для отправки по каналу передачи данных.*

Не все компьютеры могут непосредственно связаться друг с другом. Для того чтобы передать дейтаграмму по назначению, требуется направить ее через одно или несколько промежуточных устройств, по тому или иному маршруту.

Задача определения маршрута для каждой дейтаграммы решается протоколом IP.

Уровень доступа к среде передачи

К функциям этого уровня относятся:

- отображение IP-адресов в физические адреса сети (MAC-адреса). Эту функцию выполняет протокол ARP (см. раздел **3.5**);
- инкапсуляция IP-дейтаграмм в кадры для передачи по физическому каналу и извлечение дейтаграмм из кадров. При этом не требуется какого-либо контроля безошибочности передачи (хотя он может и присутствовать), поскольку в стеке TCP/IP такой контроль возложен на транспортный уровень или на само приложение;
- определение метода доступа к среде передачи – т. е. способа, с помощью которого компьютер устанавливает свое право на передачу данных;
- определение представления данных в физической среде;
- пересылка и прием кадра.

6. *ПРОМЫШЛЕННЫЕ СЕТИ

6.1. *HART-протокол

Полевой коммуникационный протокол HART широко применяется в промышленности как стандарт для цифровой коммуникации со smart-приборами. Протокол HART (Highway Addressable Remote Transducer) разработан фирмой Rosemount Inc. в середине 80-х гг., реализует известный стандарт BELL 202 FSK (Frequency Shift Keying).

Его особенность в том, что он использует для передачи цифровых данных низкоуровневую частотную модуляцию, наложенную на аналоговый сигнал **4-20 mA** (токовая петля). Обмен данными по HART протоколу происходит на скорости 1200 Бод. Эта единица названа в честь Эмиля Бодо (Jean Maurice-Emile Baudot) (1845-1903), французского инженера по телеграфии, изобретателя первого печатающего устройства для телеграфа. Схема, поясняющая работу приборов по HART протоколу, представлена на рис. 86.

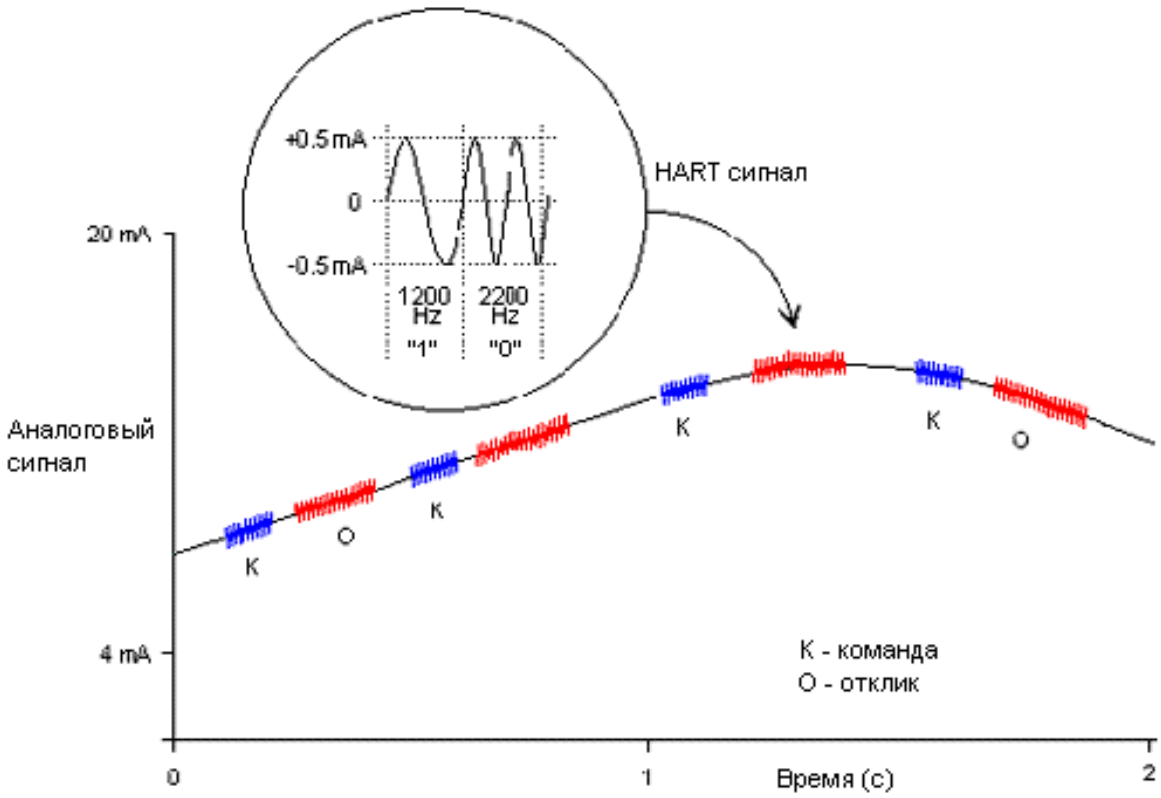


Рисунок 86 – Схема работы HART протокола

Для передачи логической «1» HART использует один полный период частоты 1200 Гц, а для передачи логического «0» – два неполных периода 2200 Гц. Как видно на рис. 86, HART составляющая накладывается на токовую петлю 4-20 мА. Поскольку среднее значение синусоиды за период равно «0», то

HART сигнал никак не влияет на аналоговый сигнал 4-20 мА, который поэтому тоже может использоваться.

HART протокол построен по принципу «главный – подчиненный», т. е. полевое устройство отвечает по запросу системы. Протокол допускает наличие двух управляющих устройств (управляющая система и коммуникатор).

Существует два режима работы датчиков, поддерживающих обмен данными по HART протоколу.

Одноточечный режим передачи данных

Режим передачи цифровой информации одновременно с аналоговым сигналом представлен на рис. 87.

Обычно в этом режиме датчик работает в аналоговых АСУ ТП, а обмен по HART-протоколу осуществляется посредством HART коммуникатора или компьютера. При этом можно удаленно (расстояние до 3000 м) осуществлять полную настройку и конфигурирование датчика.

Теперь оператору нет необходимости обходить все датчики на предприятии, он может их настроить непосредственно со своего рабочего места.

Рисунок 87 – Одноточечный режим передачи цифровой информации

Многоточечный режим передачи данных

В многоточечном режиме (рис. 88) датчик передает и получает информацию только в цифровом виде. Аналоговый выход автоматически фиксируется на минимальном значении (только питание устройства – 4 мА) и не содержит информации об измеряемой величине. Информация о переменных процесса считывается по HART-протоколу.

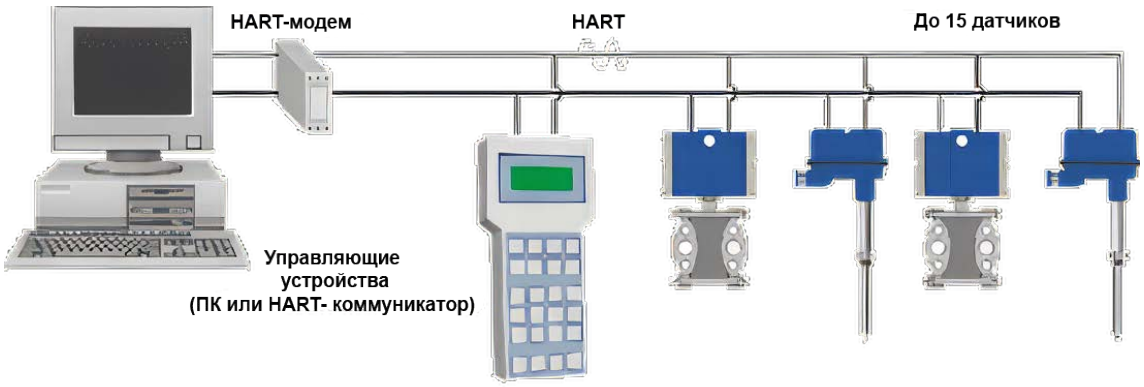


Рисунок 88 – Многоточечный режим передачи информации

К одной паре проводов может быть подключено до 15 датчиков. Их количество определяется длиной и качеством линии, а также мощностью блока питания датчиков. Все датчики в многоточечном режиме имеют свой уникальный адрес от 1 до 15, и обращение к каждому идет по соответствующему адресу. Коммуникатор или система управления определяет все датчики, подключенные к линии, и может работать с любым из них.

Обычно в аналоговой АСУТП присутствует множество интеллектуальных полевых приборов, работающих в режиме 4-20 мА + HART. В этом случае удаленная настройка и конфигурирование датчиков при помощи HART-коммуникатора или HART-модема требует последовательного подключения коммуникационного устройства к каждой линии 4-20 мА, идущей от соответствующих приборов.

Для решения поставленной задачи используется HART-мультиплексор. При таком подходе приборы продолжают передавать измерительную информацию в систему по токовому выходу 4-20 мА, а их конфигурация может быть изменена с одного цифрового выхода управляющей системы.

Связь мультиплексора с системой управления осуществляется по интерфейсу **RS-485** или **RS-232**. При этом можно объединить в сеть около 500 приборов (например, 30 мультиплексоров, соединенных по RS-485, 16 каналов каждый).

Структурная схема работы мультиплексора в аналоговой системе представлена на рис. 89.

Существует возможность построения с помощью мультиплексора цифровой системы сбора и визуализации информации.

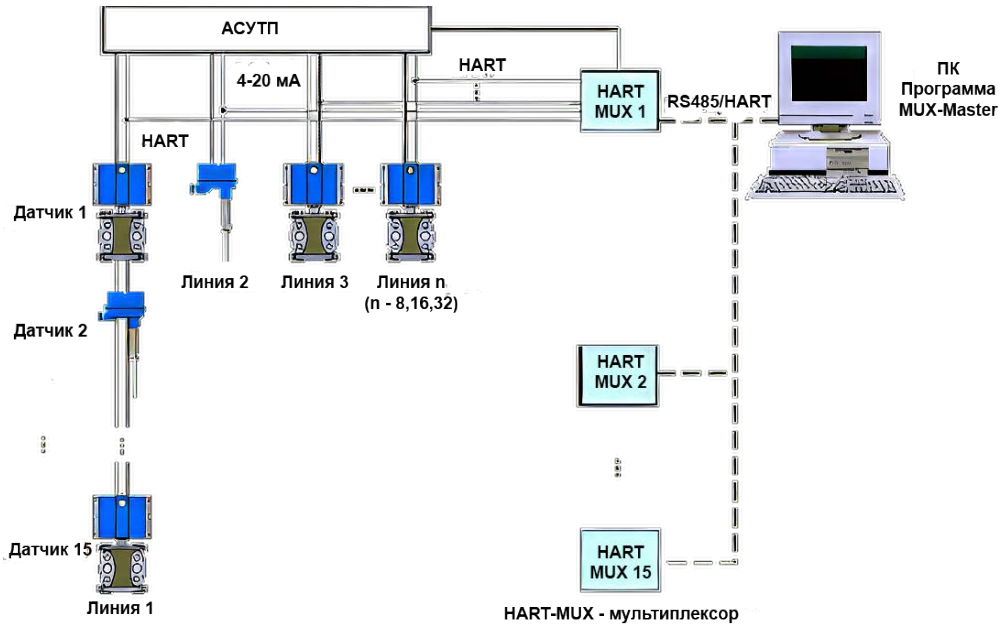


Рисунок 89 – Структурная схема работы мультиплексора

В этом случае каждый канал мультиплексора может опрашивать до 15 датчиков, подключенных к одной токовой петле.

При таком подключении затраты на кабельную продукцию существенно снижаются (см. рис. 89).

6.2. *Определение промышленной сети

Промышленная сеть – это набор стандартных протоколов обмена данными, позволяющих связать воедино оборудование (датчики, исполнительные механизмы, промышленные контроллеры) различных производителей, а также обеспечить взаимодействие нижнего и верхнего уровней АСУ.

В промышленных сетях для передачи данных применяют кабели, оптоволоконные линии, беспроводную связь. Промышленные сети могут взаимодействовать с обычными компьютерными сетями, в частности, использовать глобальную сеть Интернет.

Выделяют три наиболее значимых параметра, по которым классифицируют сети [4]: топология сети, объем информационного сервиса, предоставляемого сетью, и способ доступа к физическому каналу передачи данных.

Топологии сетей (общая шина, кольцо, звезда) рассмотрены ранее в разделе **1.9.**

6.3. *Объем информационного сервиса

О модели, разграничивающей и формализующей функции, выполняемые различными уровнями аппаратного и программного обеспечения сетевой структуры, изложено в Главе **5**. На практике большинство промышленных сетей ограничивается только тремя уровнями: физическим, канальным и прикладным. Дешевые сети (например, **ModBus**) зачастую используют на физическом уровне интерфейс RS-232 или RS-485, а все остальные задачи, начиная с канального уровня, решают программным путем.

6.4. *Сеть ASI (Actuator Sensor Interface)

Основная задача этой сети – связать в единую информационную структуру устройства самого нижнего уровня автоматизируемого процесса (датчики и разнообразные исполнительные механизмы) с системой контроллеров.

ASI-интерфейс позволяет через свои коммуникационные линии не только передавать данные, но и запитывать датчики. Здесь используется принцип последовательной передачи на базовой частоте. Информационный сигнал модулируется на питающую частоту.

В качестве физической среды используется специальный неэкранированный двухпроводный кабель с трапециевидным профилем. Этот кабель позволяет подключать датчики, устанавливаемые на подвижных частях механизмов. Топологией ASI-сети может быть шина, звезда, кольцо или дерево с циклом опроса 31 узла за 5 мс.

6.5. *Сеть FOUNDATION FIELDBUS

Основная область применения этой сети – самый нижний уровень распределенной системы автоматизации с обвязкой устройств, работающих во взрывоопасных средах и использующих сеть как для информационного обмена, так и для собственной запитки.

Foundation Fieldbus (**FF**) – самый молодой и быстро растущий стандарт на промышленную сеть. FF представляет собой двухуровневый сетевой протокол, сочетающий черты мощной информационной магистрали для объединения компьютеров верхнего уровня и управляющей сети, объединяющей контроллеры, управляющие компьютеры, датчики и исполнительные механизмы.

На нижнем уровне в качестве физической среды передачи данных за основу взят стандарт IEC 61158-2, который позволяет использовать сеть FF на взрывоопасных производствах с возможностью запитки датчиков непосредственно от канала связи. Скорость передачи информации на нижнем уровне составляет 31,5 Кбит/с.

На верхнем уровне в настоящее время, как правило, используется FF HSE (High Speed Ethernet), основанный, как видно по названию, на сети **Ethernet** со

скоростью 100 Мбит/с. Особенностью стандарта FF является то, что в нем определен дополнительный пользовательский уровень (User Layer), позволяющий, применяя предопределенные функциональные блоки, строить промышленные сети с распределенным интеллектом.

6.6. *Сеть PROFIBUS

При построении многоуровневых систем автоматизации, как правило, стоят задачи организации информационного обмена между уровнями. В одном случае необходим обмен сообщениями на средних скоростях. В другом – быстрый обмен короткими сообщениями с использованием упрощенного протокола обмена (уровень датчиков). В третьем требуется работа в опасных участках производства (переработка газа, химическое производство). Для всех этих случаев PROFIBUS имеет решение. Сегодня под PROFIBUS понимается совокупность трех отдельных протоколов: PROFIBUS-FMS, PROFIBUS-DP и PROFIBUS-PA. Все три варианта протокола используют общий канальный уровень (уровень 2 OSI-модели).

Протокол PROFIBUS-FMS включает в себя дополнительные типы пакетов (Fieldbus Message Specification), появился первым и был предназначен для работы на так называемом цеховом уровне. Позволяет организовывать в одной сети работу нескольких активных станций.

Протокол PROFIBUS-DP был спроектирован для организации быстрого (до 12 Мбит/с) канала связи с датчиковым уровнем. Физическая среда передачи – экранированная витая пара стандарта RS-485. В основе алгоритма работы лежит модель циклического опроса каналов. Кроме этого, существует набор ациклических функций для конфигурирования, диагностики и поддержки сигналов. В DP-протоколе существуют три типа устройств.

Протокол PROFIBUS-PA – сетевой интерфейс, физическая среда передачи данных которого соответствует требованиям стандарта IEC 61158-2. Может применяться для построения сети, соединяющей исполнительные устройства, датчики и контроллеры, расположенные непосредственно во взрывоопасной зоне. Он может использоваться в качестве замены старой технологии связи 4-20 мА. Для коммутации устройств нужна всего одна витая пара, которая может одновременно использоваться и для информационного обмена, и для запитывания устройств.

На одном физическом канале (RS-485 или оптоволокне) одновременно могут работать устройства PROFIBUS всех трех типов. Рабочая скорость передачи может быть выбрана в диапазоне 9,6-12000 Кбит/с.

PROFIBUS широко используется для модернизации и расширения возможностей существующих систем. Если требуется объединить в детерминированную сеть несколько контроллеров, оптимальным вариантом будет PROFIBUS-FMS. Для создания сети с централизованным интеллектом и распределенным вводом/выводом лучше всего подойдет PROFIBUS-DP.

7. ОПЕРАЦИОННЫЕ СИСТЕМЫ

7.1. Расширения файлов и скрытые файлы

Для изменения настроек отображения папок необходимо через меню пункт «Вид» вызвать «Параметры» папок (рис. 90).

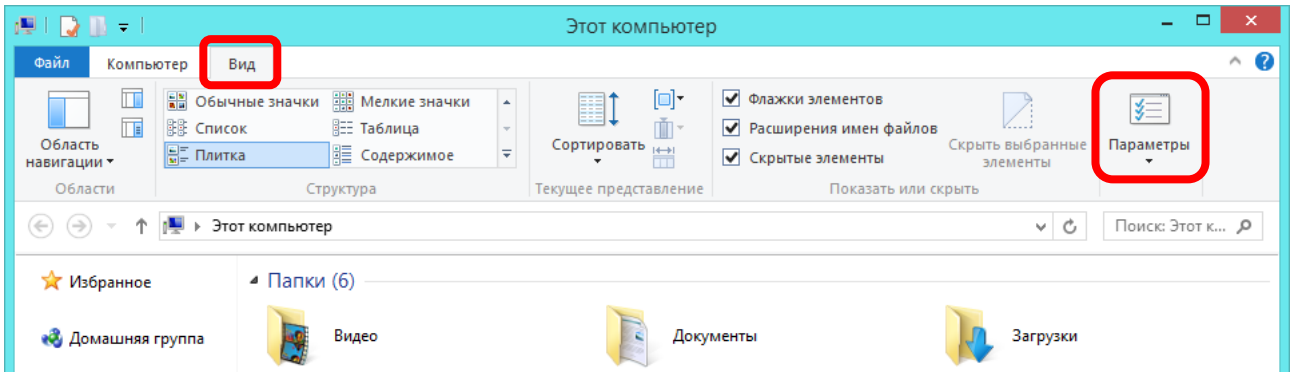


Рисунок 90 – Открытие параметров папок

В открывшемся окне на вкладке «Вид» (рис. 91) разрешаем отображение «скрытых файлов» и отображение «расширений» файлов.

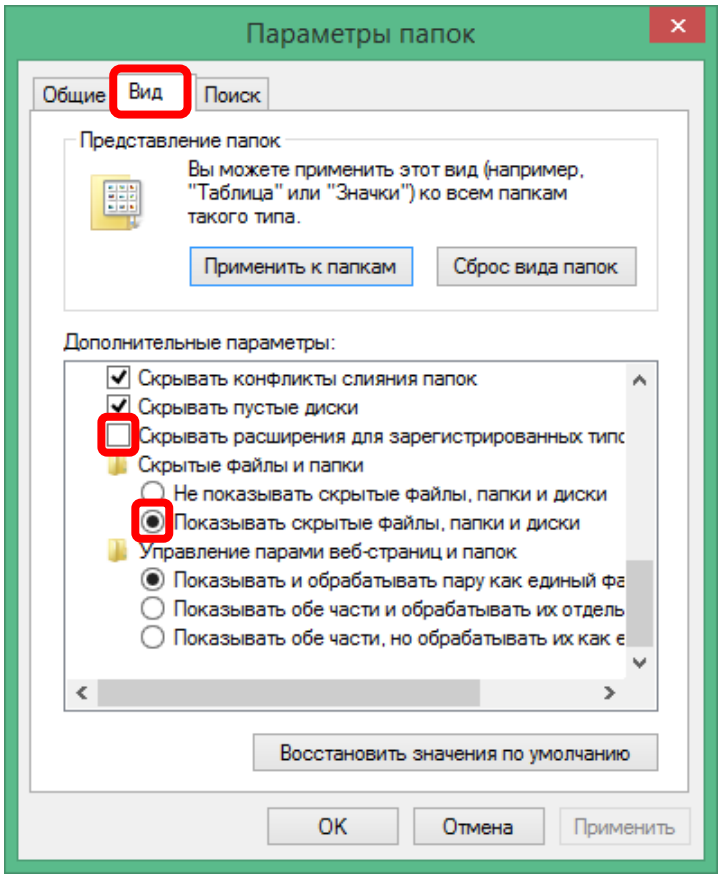


Рисунок 91 – Настройка параметров папок

7.2. Командный процессор CMD

Для вызова командного процессора Windows необходимо нажать «Win+R» (рис. 92) и ввести «cmd».

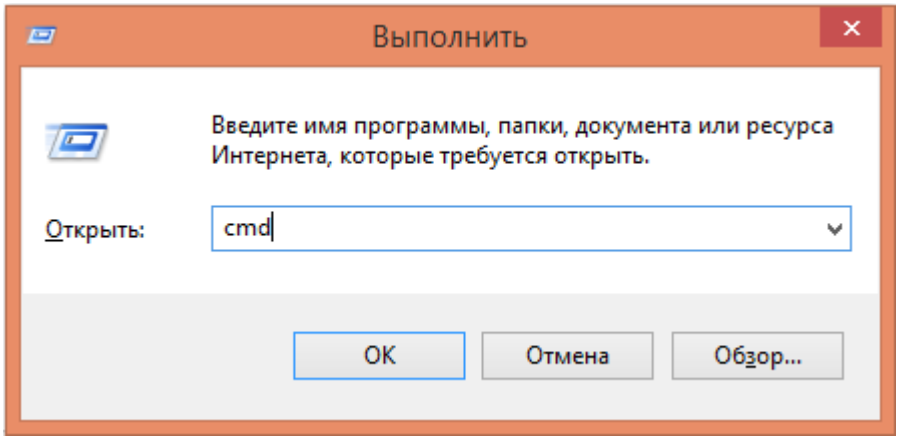


Рисунок 92 – Вызов «cmd»

В открывшемся окне (рис. 93) можно вводить различные команды.

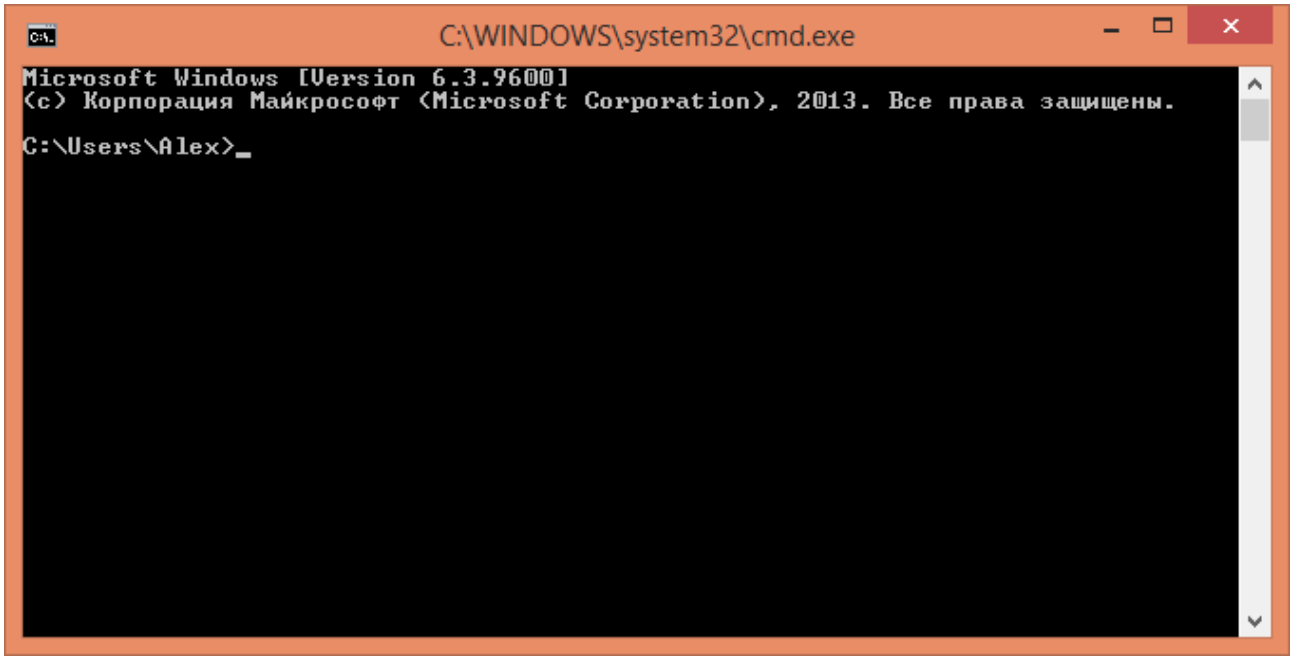


Рисунок 93 – Командный процессор

Вызов справки

Для отображения списка основных команд (рис. 94) достаточно ввести команду «help» («помощь»).

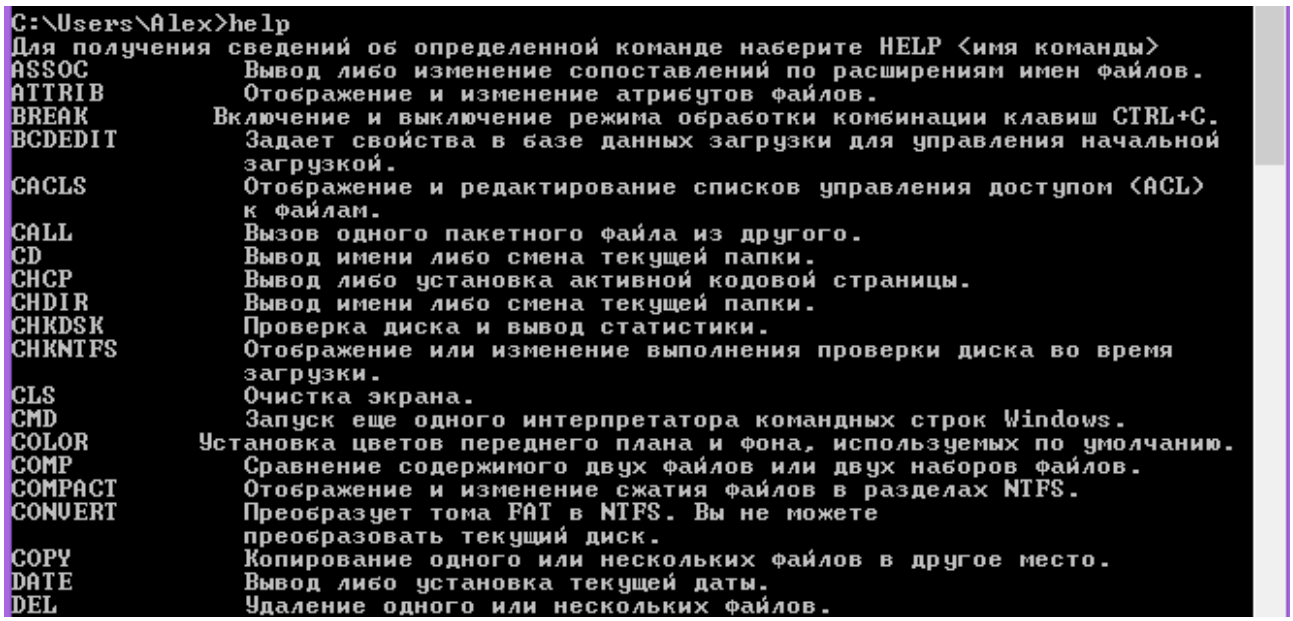


Рисунок 94 – Вывод списка команд

Для вызова справки по конкретной команде необходимо ввести ее имя, после чего через пробел дописать «/?», например, «copy /?» (рис. 95).

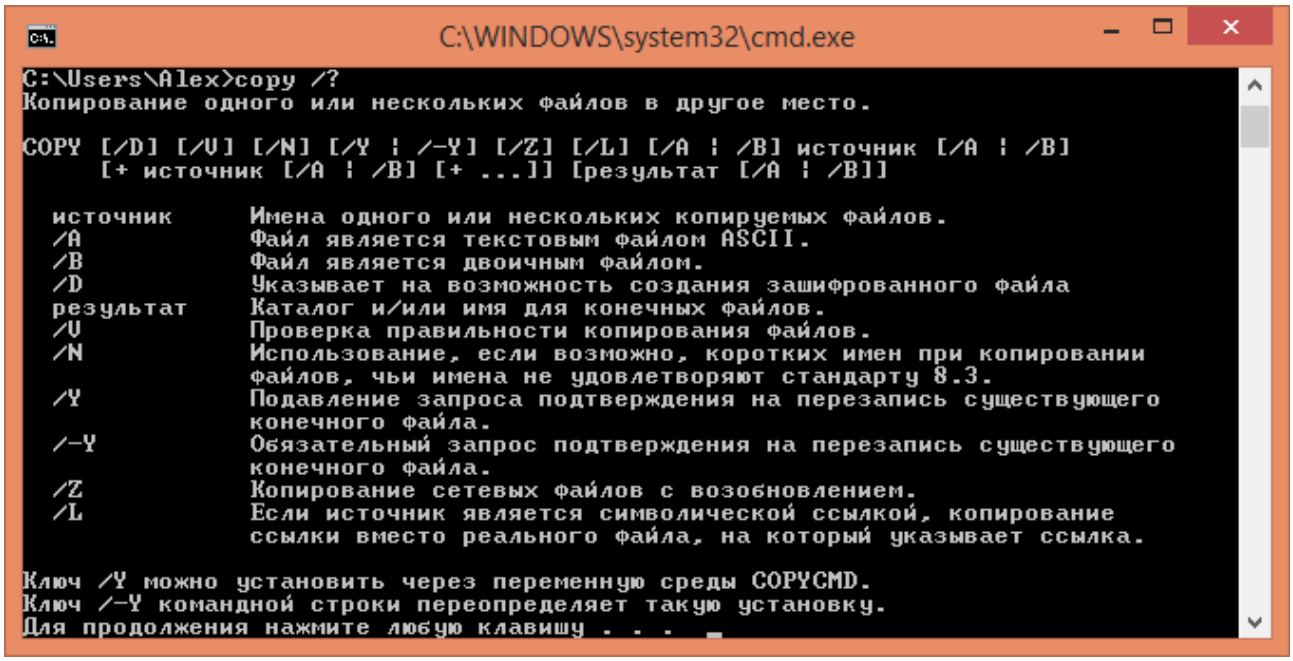


Рисунок 95 – Справка по команде Copy

7.3. Команды для работы с файлами

Основные команды командной строки для работы с файлами и папками (каталогами, директориями) представлены в табл. 5.

Таблица 5 – Команды для работы с файлами

CD *или* **ChDir**	Изменить (или отобразить) текущий каталог, например: CD C:\
Dir	Вывод списка файлов и подкаталогов в указанном каталоге
Tree	Вывод структуры папок
Copy	Копирование одного или нескольких файлов в другое место, например: COPY c:\A\File1.txt c:\B\ *Папка, в которую производится копирование, должна существовать*
XCopy	Расширенная версия, копирующая файлы и дерево каталогов
Ren *или* **Rename**	Переименование одного или нескольких файлов, например: REN "c:\File 1.txt" "File 2.txt"
Move	Перемещение файлов и переименование файлов и папок
MD *или* **MakeDir**	Создание каталога
Del	Удаление одного или нескольких файлов
RD *или* **RmDir**	Удаление каталога

Кроме того, во всех путях к файлам и папкам можно использовать две точки «..», означающие подъем на один уровень (переход к родительской папке). Например, для подъема на два уровня относительно текущей папки (CD) необходимо написать:

CD ..\..

7.4. ЛАБОРАТОРНАЯ РАБОТА № 4
Создание BAT-файла

Пакетный файл (batch file), **BAT**-файл, **командный файл**, сценарий (скрипт) командной строки, сценарий командной оболочки – текстовый файл, содержащий последовательность команд, предназначенных для исполнения командным интерпретатором (**CMD**). Пакетные файлы полезны для автоматического запуска приложений. Основная область применения – автоматизация рутинных операций, которые регулярно приходится совершать пользователю компьютера, например, копирование, перемещение, переименование, удаление файлов; работа с папками и т. п.

В Windows пакетные файлы имеют расширения ***.bat** или ***.cmd**. В Linux сценарии командной строки обычно имеют расширение ***.sh**.

Команды из пакетного файла обычно выполняются последовательно, начиная с первой. Причем каждая следующая команда выполняется только после того, как завершится предыдущая. Но имеются также операторы if, goto и for, позволяющие изменить порядок выполнения команд.

Создание BAT-файла

Пакетный файл – это обычный текстовый файл, поэтому достаточно создать «Текстовый документ» (рис. 96) в выбранной папке, после чего изменить его расширение на ***.bat** (рис. 97). *О том, как включить отображение расширений файлов, было рассказано ранее в разделе* ***7.1.***

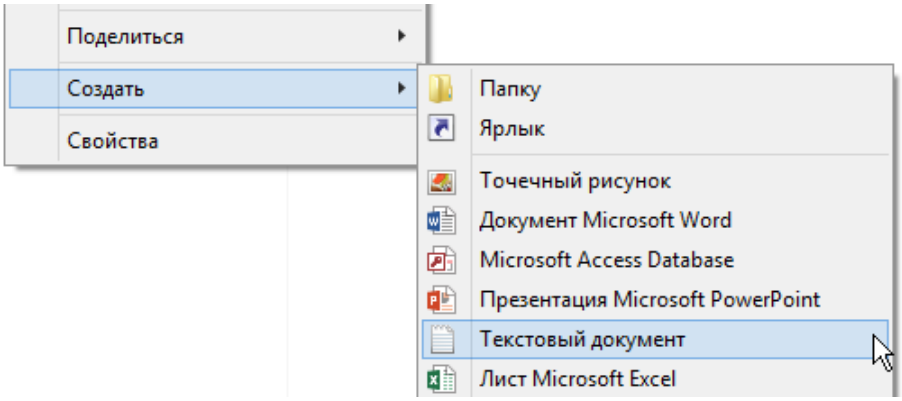


Рисунок 96 – Создание текстового документа

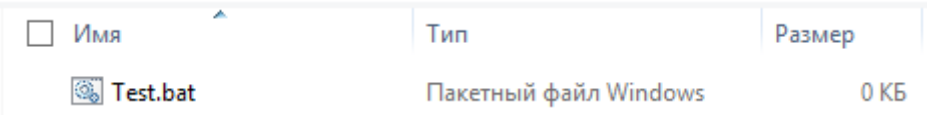


Рисунок 97 – Новый BAT-файл

Если дважды кликнуть по этому файлу, то он будет запущен. А для его редактирования необходимо нажать правую кнопку мышки и выбрать «Изменить» (рис. 98). Редактирование производится в обычном Блокноте.

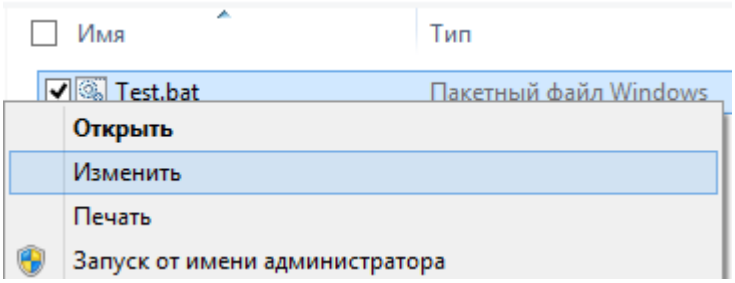


Рисунок 98 – Изменить BAT-файл

Например, напишем простейший код (рис. 99).

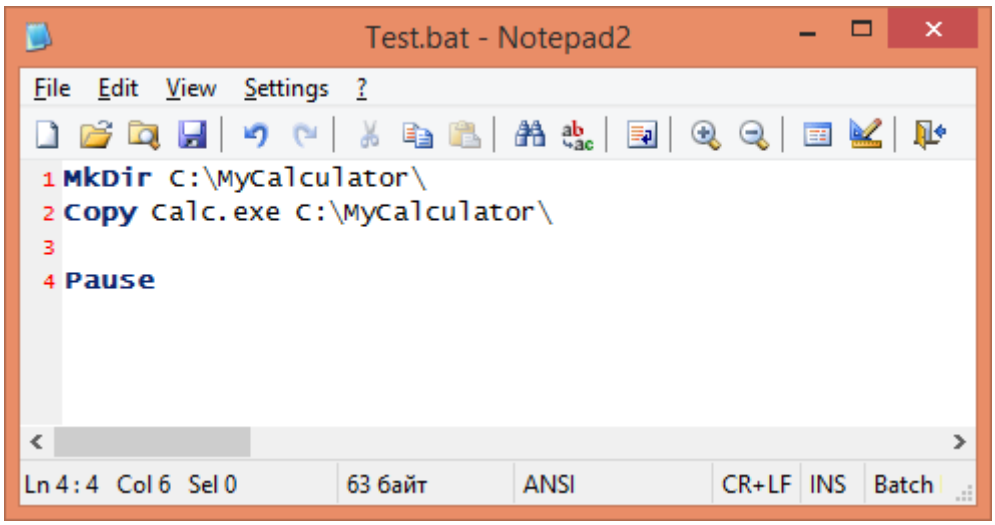


Рисунок 99 – Редактирование BAT-файла

Данный файл создает папку «MyCalculator» на диске «C:\» и копирует в эту папку файл «Calc.exe» (файл необходимо заранее положить в ту же папку, в которой расположен BAT-файл). В конце выполняется команда «**Pause**», ожидающая нажатия клавиши пользователем. Если ее не использовать, то окно CMD (с черным экраном) закроется мгновенно, и мы не успеем увидеть результат. Результат работы командного файла представлен на рис. 100.

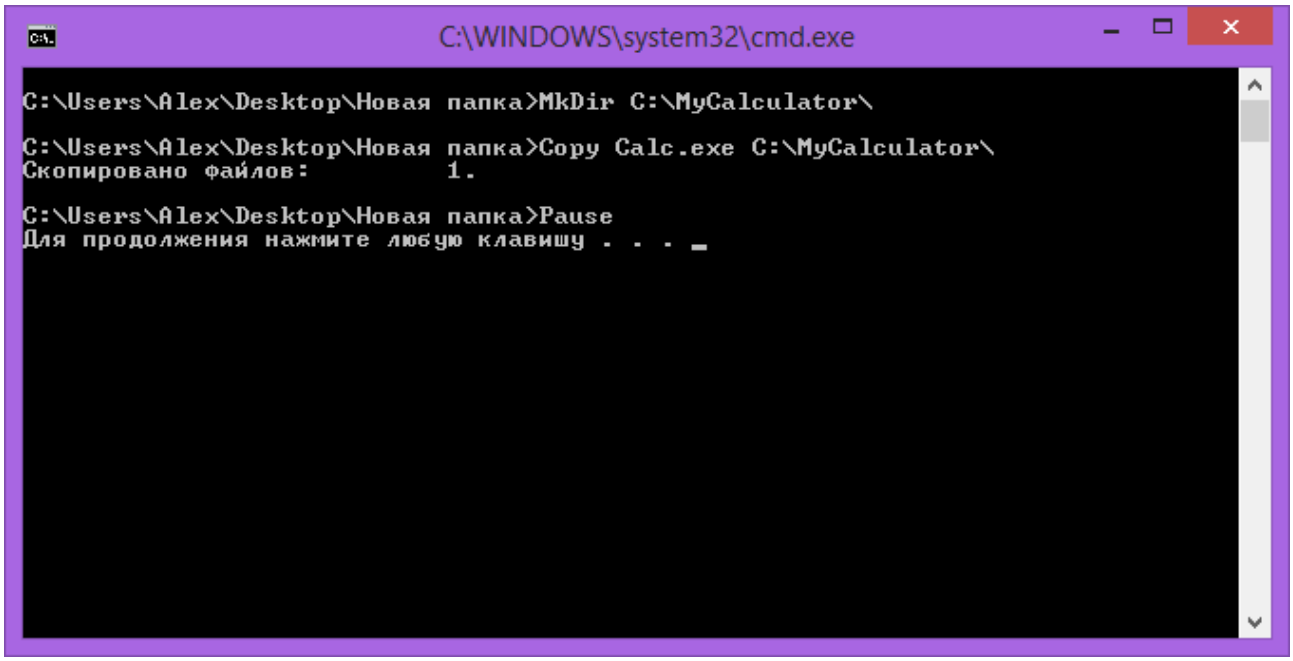


Рисунок 100 – Результат работы BAT-файла

Переменные в командных файлах

В пакетных файлах можно использовать переменные, их имена заключаются между знаками «%», например, **%ProgramFiles%**. Для присвоения значения переменной используется команда «**Set**». Переменные могут хранить

только текстовые значения. *Эти переменные называются «Переменные среды», подробнее про них будет рассказано далее, в разделе* ***7.5****.*

Переменные командной строки

Кроме Переменных среды, в BAT-файлах существуют еще Переменные командной строки (аргументы командной строки). Они используются для передачи входных значений в BAT-файл при его вызове через командную строку.

Переменные командной строки записываются как «**%1**», «**%2**», … «**%9**».

7.5. ПРАКТИЧЕСКАЯ РАБОТА
Переменные среды

Переменная среды (environment variable) – текстовая переменная операционной системы, хранящая какую-либо информацию – например, данные о настройках системы.

Просмотр и настройка переменных среды

Для просмотра и настройки переменных среды Windows необходимо:

– открыть окно «Свойства системы»;
– перейти на вкладку «Дополнительно» (рис. 101);
– нажать кнопку «Переменные среды».

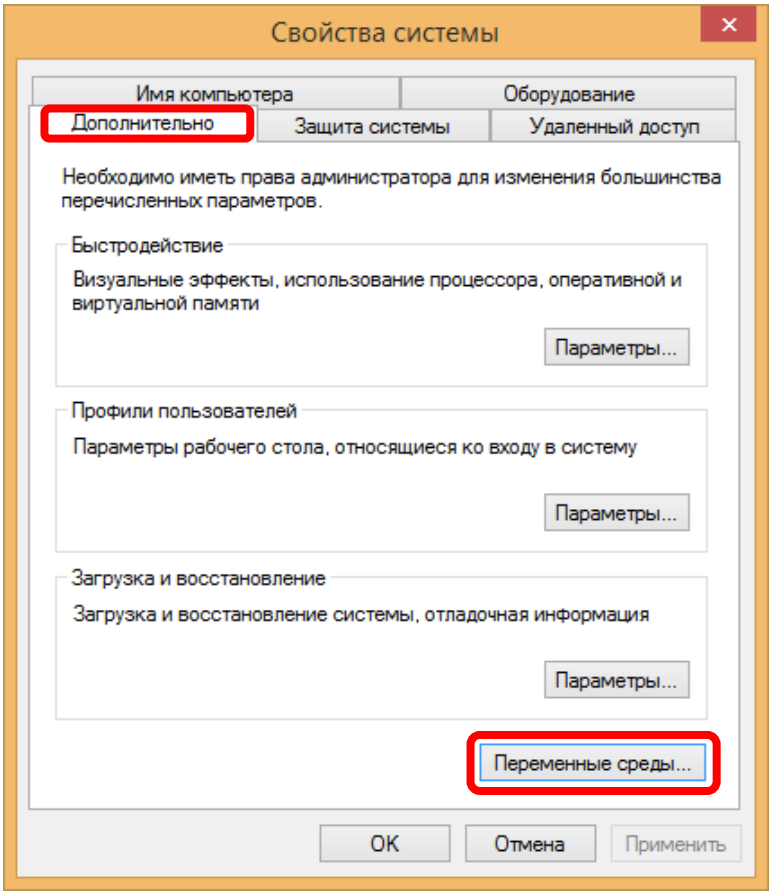


Рисунок 101 – Открытие переменных среды

Переменные среды Windows делятся на две категории (рис. 102):

– переменные среды пользователя – содержат настройки конкретного пользователя, например, указывают путь до пользовательских каталогов;
– системные переменные – хранят данные о некоторых каталогах операционной системы и конфигурации компьютера.

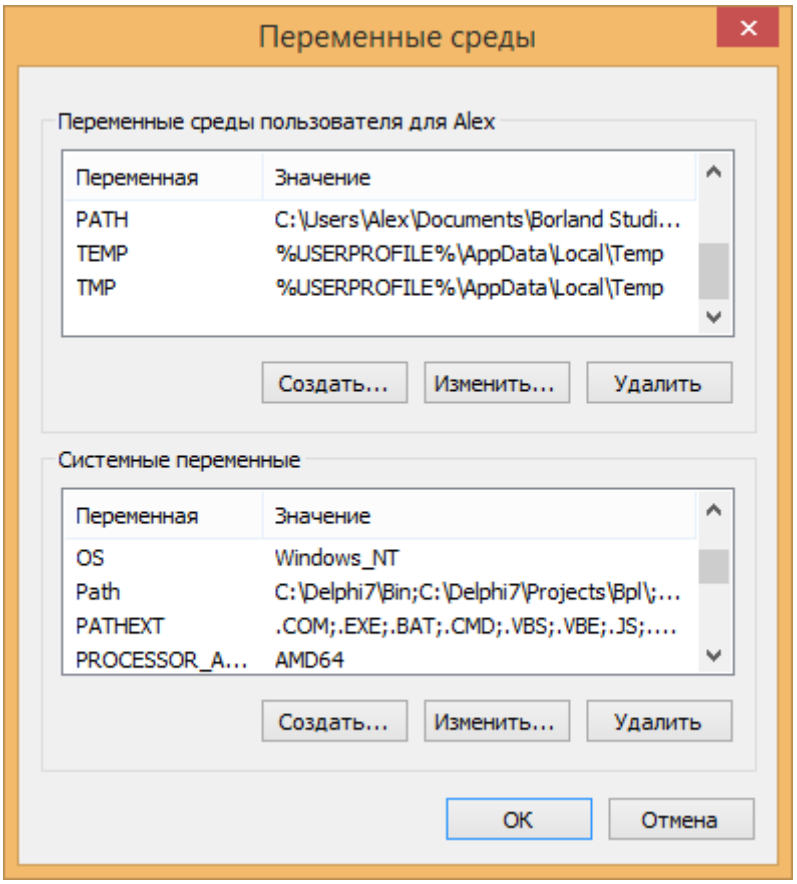


Рисунок 102 – Списки переменных среды

Использование переменных среды

Переменные среды можно использовать в адресной строке проводника Windows. Например, вместо **%ProgramFiles%** автоматически будет подставлен адрес «C:\Program Files» (рис. 103). Аналогично вместо **%ProgramFiles(x86)%** отобразится «C:\Program Files (x86)», вместо **%WINDIR%** – «C:\Windows», а вместо **%USERPROFILE%** – «C:\Users\Alex» (где «Alex» – имя текущего пользователя). *При этом если Windows будет установлен на другой диск, то, например, переменная* ***%WINDIR%*** *будет автоматически выводить правильный адрес (например, «D:\Windows»).*

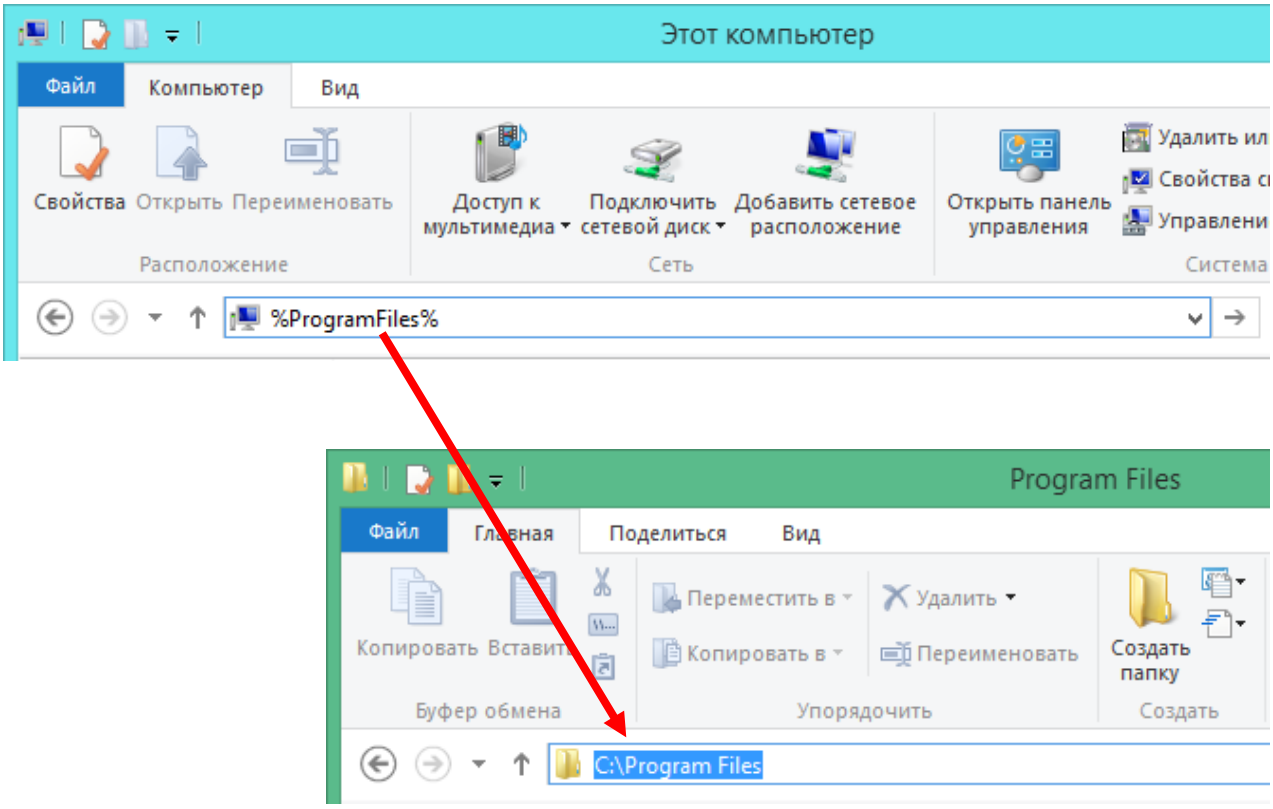


Рисунок 103 – Использование переменой среды

Просмотр переменных среды через командную строку

Для просмотра списка всех переменных среды через командную строку (рис. 104) достаточно ввести команду «**set**».

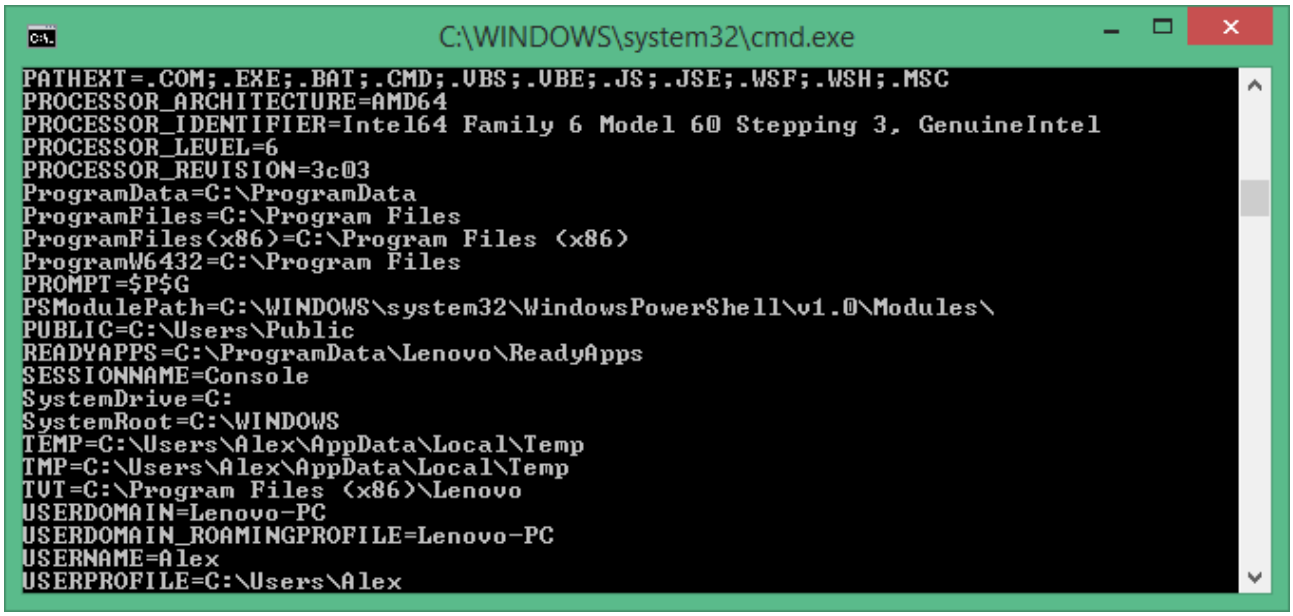


Рисунок 104 – Список переменных среды

Настройка переменных среды через командную строку

Просмотреть справку по команде Set (рис. 105) можно, введя команду «set /?».

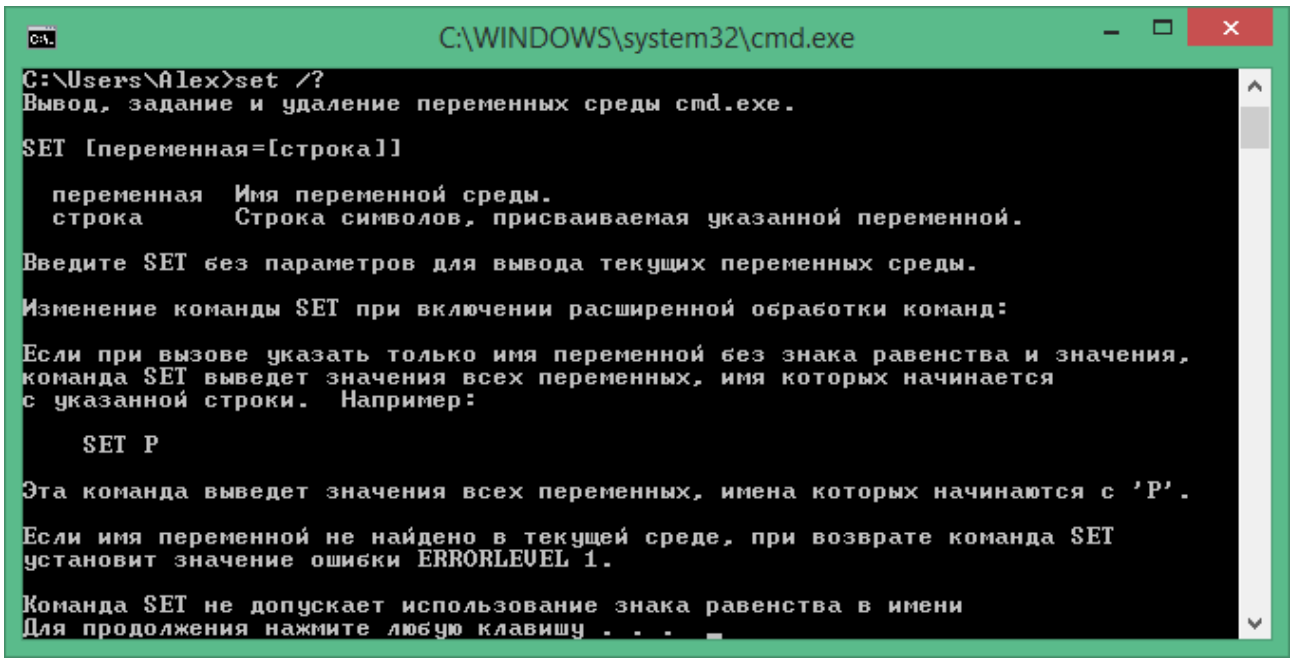


Рисунок 105 – Справка по команде Set

Для того чтобы добавить или изменить значение переменной среды, достаточно написать команду вида «set AAA=55».

7.6. Реестр Windows

Реестр Windows (Windows Registry), или системный реестр – база данных (в виде «дерева») параметров и настроек в операционной системе Microsoft Windows. Реестр содержит информацию и настройки для аппаратного и программного обеспечения, профилей пользователей, панели управления, проводника, ассоциации файлов, системные политики, список установленного ПО и др.

Другим распространенным способом хранения настроек являются INI-файлы. Реестр применяется только в Windows, и подобного нет в других операционных системах

Для вызова редактора реестра Windows необходимо нажать «Win+R» (рис. 106) и ввести «regedit».

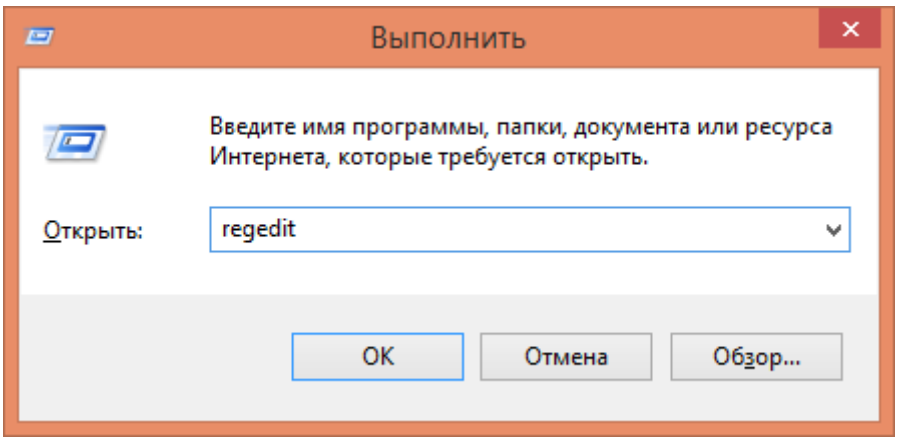


Рисунок 106 – Вызов «regedit»

Открывшееся окно (рис. 107) будет содержать 5 разделов.

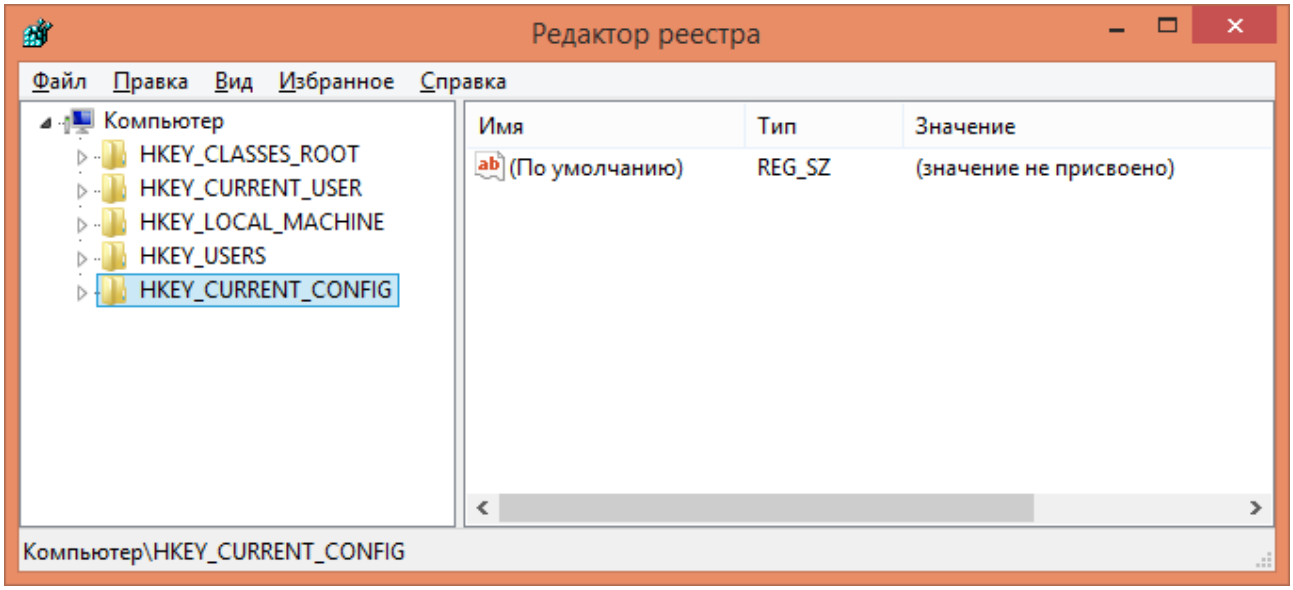


Рисунок 107 – Редактор реестра

Разделы реестра

– **HKEY_CURRENT_USER** (HKCU).

Данный раздел содержит настройки текущего активного пользователя, вошедшего в систему. Здесь хранятся папки пользователя, цвета экрана и параметры панели управления. Эти сведения сопоставлены с профилем пользователя. Вместо полного имени раздела иногда используется аббревиатура HKCU. *Хотя этот раздел выглядит как один из основных в редакторе реестра, он является всего лишь ссылкой на один из профилей HKEY_USERS.*

– **HKEY_USERS** (HKU).

Раздел HKEY_USERS содержит информацию о профилях всех пользователей данного компьютера.

– **HKEY_LOCAL_MACHINE** (HKLM).

Раздел содержит параметры конфигурации, относящиеся к данному компьютеру (для всех пользователей).

– HKEY_CLASSES_ROOT (HKCR).

В основном содержит информацию о зарегистрированных типах файлов и объектах COM и ActiveX.

– **HKEY_CURRENT_CONFIG** (HKCC).

Данный раздел содержит сведения о профиле оборудования, используемом локальным компьютером при запуске системы. Является ссылкой на HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware Profiles\Current.

Типы данных, хранимых в реестре

В реестре можно создавать параметры, имеющие типы данных, представленные на рис. 108.

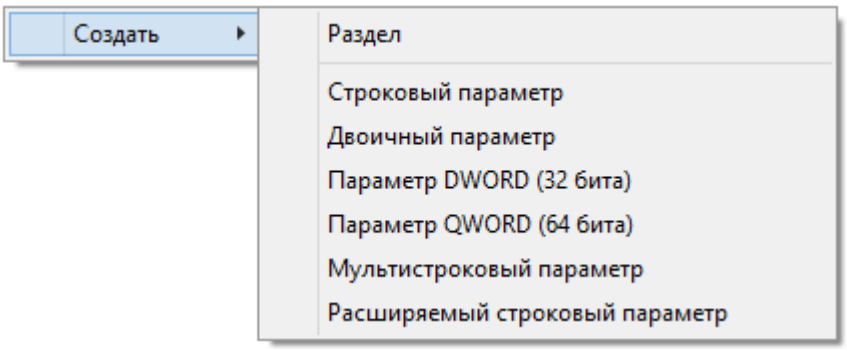


Рисунок 108 – Типы параметров

*Команды для работы с реестром

С реестром можно работать из командной строки (а также из BAT-файла). Для этого используется команда «**reg**». Справка по команде «reg» представлена на рис. 109. Например, для добавления параметра или раздела в реестр используется команда «**reg add**».

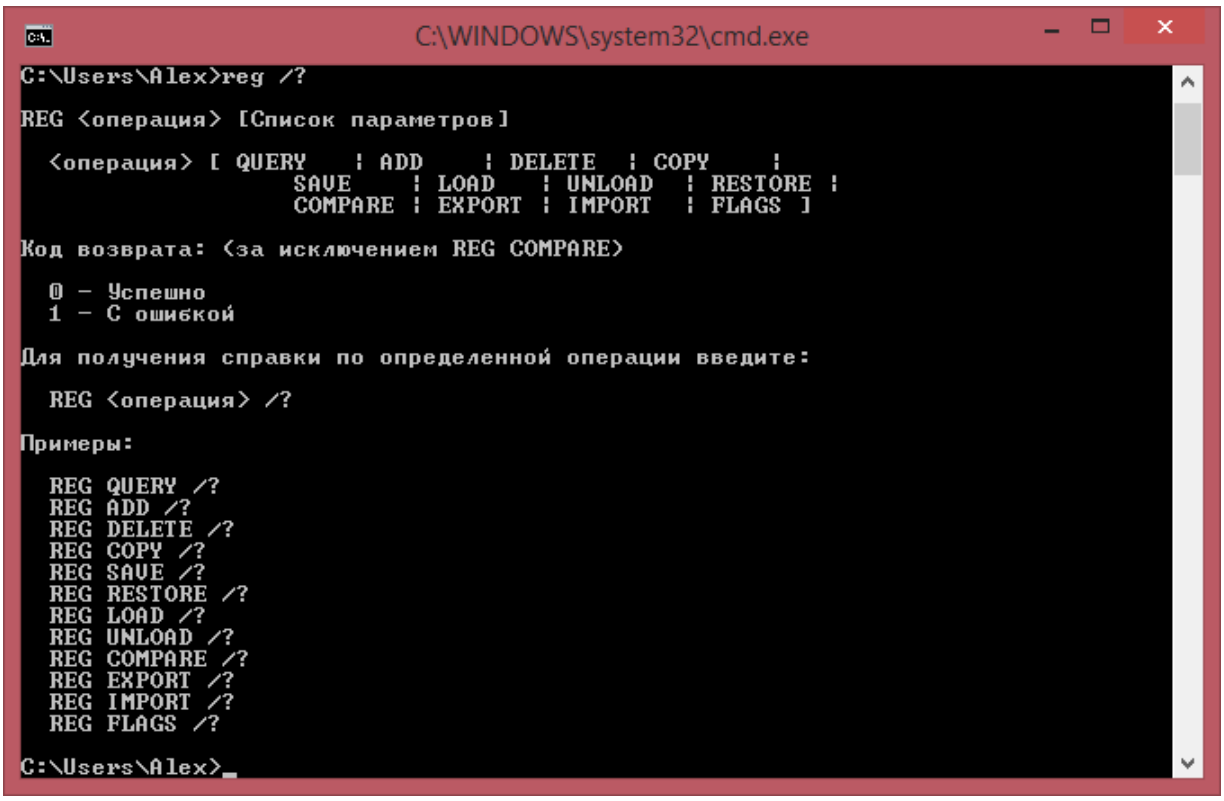


Рисунок 109 – Справка по команде «reg»

7.7. *Примеры настройки операционной системы через реестр

*Автозагрузка

Автозагрузка в реестре находится по адресу HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run (рис. 110).

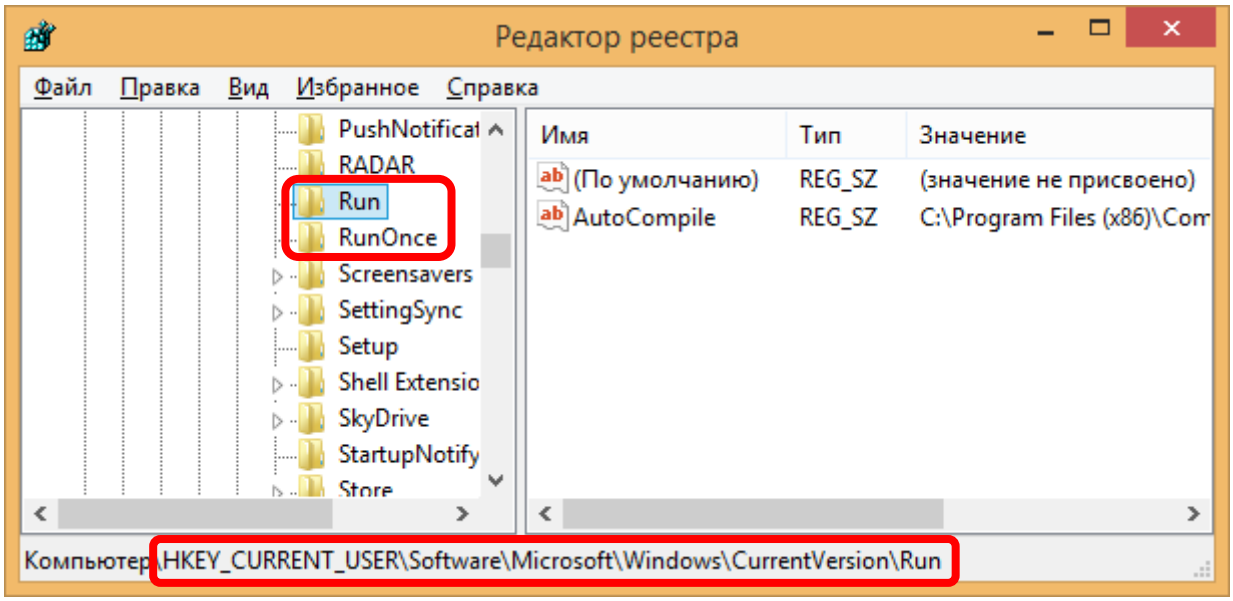


Рисунок 110 – Автозагрузка в реестре

В этом разделе для каждой программы (которую требуется автозагружать) нужно создать «Строковый параметр», в значении которого прописать полный путь к запускаемой программе (т. е. к exe-файлу).

Приведенный адрес используется только для автозагрузки текущего пользователя. Автозагрузка для всех пользователей расположена по адресу HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run.

Если мы хотим, чтобы программа в автозагрузке выполнилась только один раз (при следующей загрузке Windows), то вместо раздела «**Run**» нужно использовать раздел «**RunOnce**».

*Отображение расширений файлов и скрытых файлов через реестр

Настройки отображения расширений файлов, скрытых и системных файлов находятся по адресу HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced (рис. 111). *О том, как осуществить данные настройки без использования реестра, уже рассказывалось ранее в разделе **7.1** (см. рис. 91).*

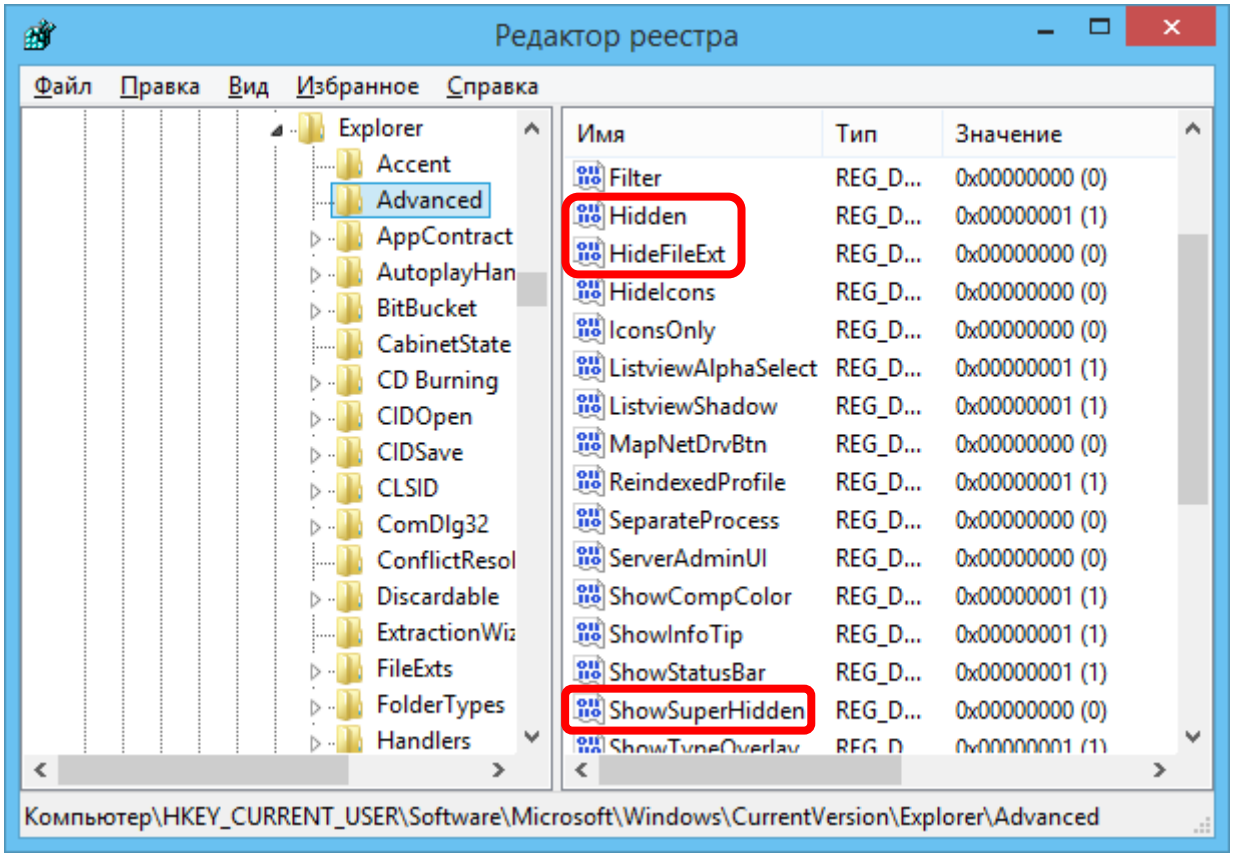


Рисунок 111 – Настройки отображения файлов

Здесь нас интересуют следующие параметры:

- **Hidden** – отобразить скрытые файлы (1 = «отобразить»);
- **HideFileExt** – скрыть расширения файлов (0 = «отобразить»);
- **ShowSuperHidden** – отобразить системные файлы (0 = «скрыть»).

**Сопоставление расширений файлов

Рассмотрим связывание расширения файлов на примере формата *.rtf. Если требуется сопоставить другой тип файлов, то необходимо заменить все надписи «rtf» на название соответствующего расширения.

Настройка осуществляется в разделе HKEY_CLASSES_ROOT\rtffile\shell\open (рис. 112). Запускаемая программа (в нашем примере это Note.exe) указывается в разделе «command» (рис. 113).

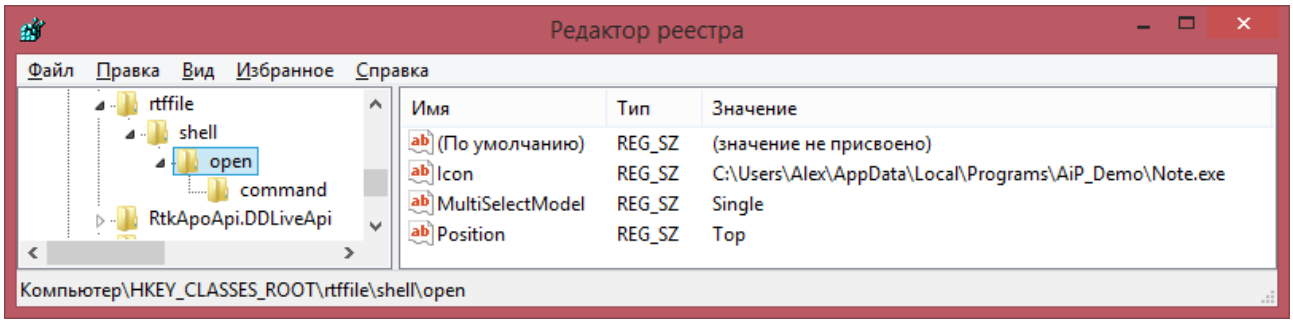


Рисунок 112 – Настройка внешнего вида

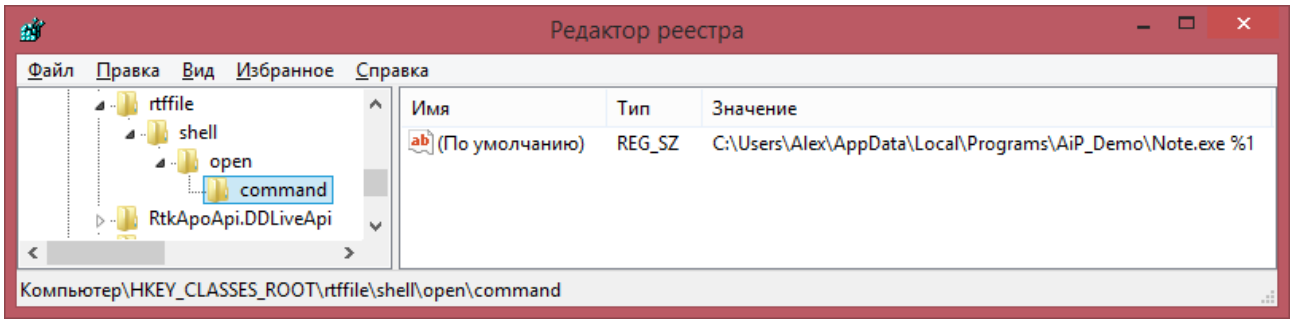


Рисунок 113 – Настройка выполняемой команды

Кроме того, необходимо создать раздел HKEY_CLASSES_ROOT**.rtf** (если он еще не существует) и задать для него значение «rtffile» (рис. 114).

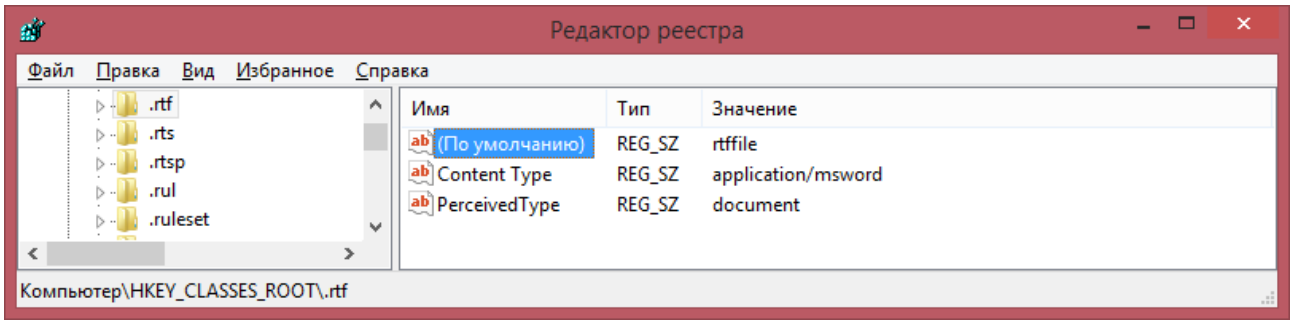


Рисунок 114 – Сопоставление расширения файла

После этих действий наша программа (Note.exe) добавится в список «Открыть с помощью» (рис. 115).

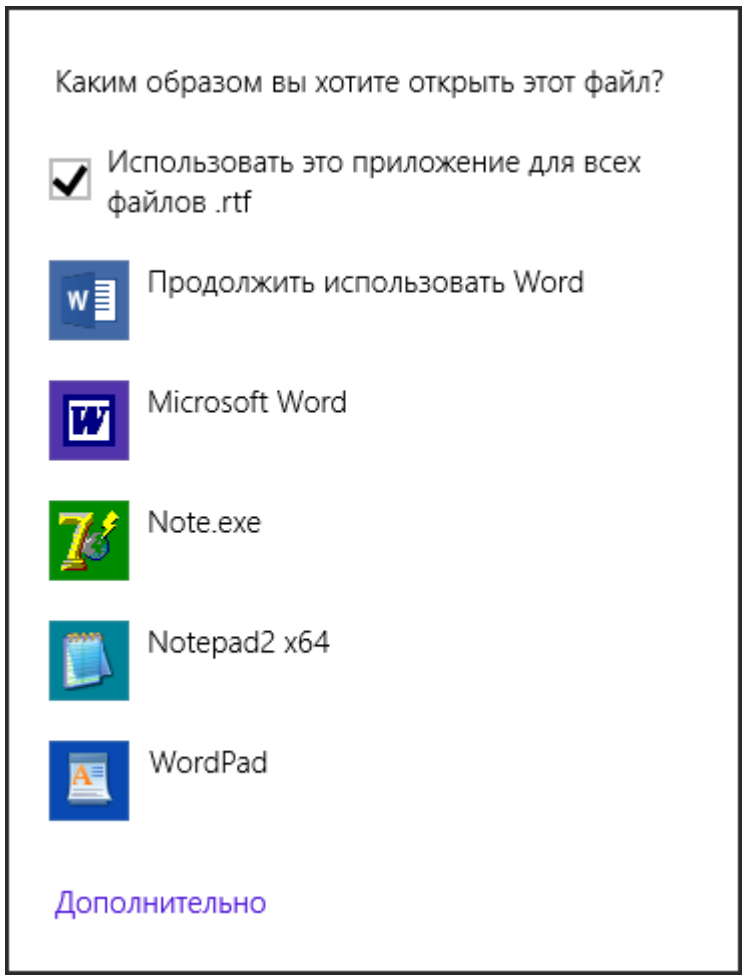


Рисунок 115 – Открыть с помощью

****Настройка меню проводника**

Аналогично описанному выше сопоставлению расширений файлов, можно настроить (рис. 116–117) и содержимое контекстного меню Проводника. Для примера используются файлы с расширением *.bat.

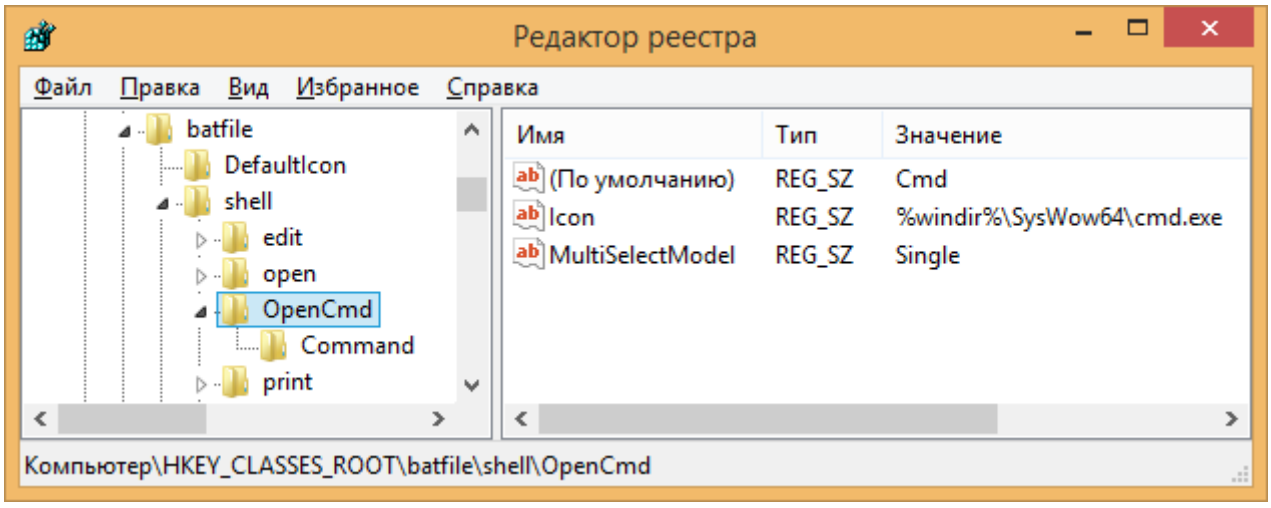


Рисунок 116 – Настройка внешнего вида пункта меню

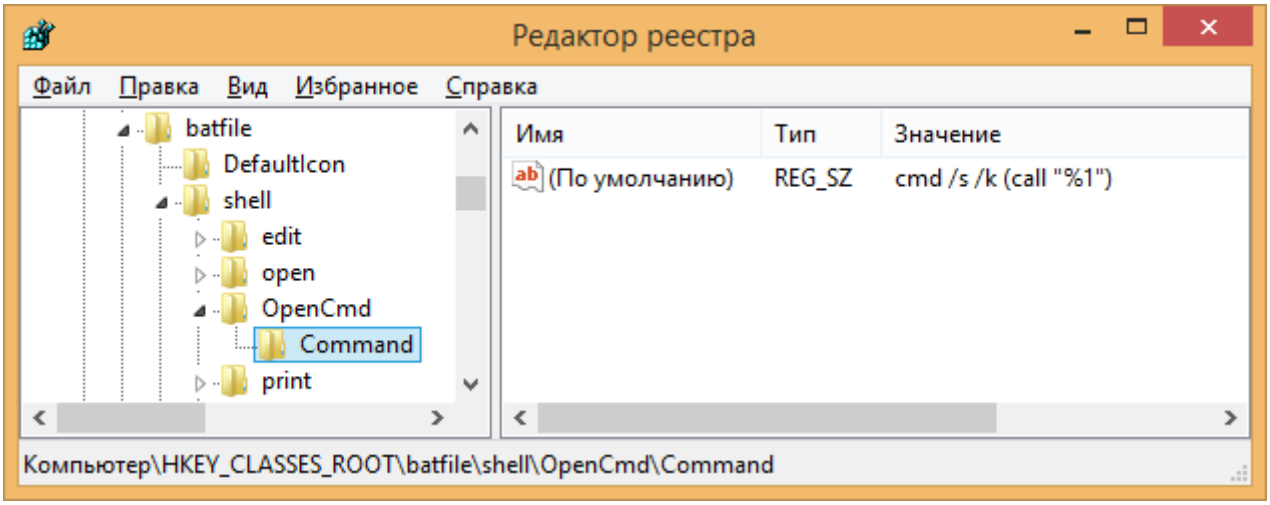


Рисунок 117 – Настройка команды, выполняемой пунктом меню

В результате в меню будет добавлен пункт «Cmd» (рис. 118).

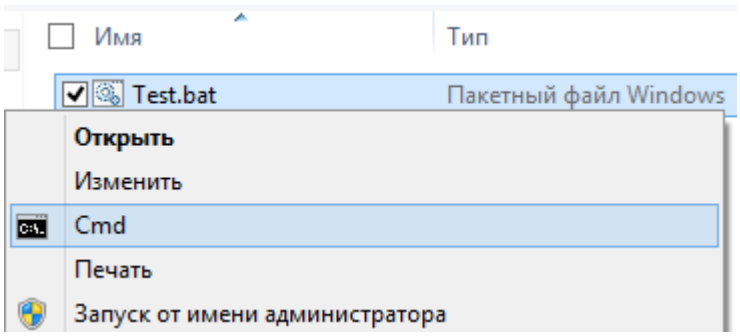


Рисунок 118 – Контекстное меню Проводника

Там же в реестре можно настроить и пункты «Открыть», «Изменить», «Печатать» (см. рис. 118) – «open», «edit», «print» (см. рис. 117).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Холмогоров, В. Компьютерная сеть своими руками: самоучитель / В. Холмогоров. – СПб.: Питер, 2004. – 171 с. – Текст: непосредственный.
2. Лекция 3: Процесс передачи данных. – URL: https://sharovt.narod.ru/l03.htm (дата обращения: 11.01.2024). – Текст: электронный.
3. Прохоров, А. Н. Интернет: как это работает / А. Н. Прохоров. – СПб.: БХВ-Петербург, 2004. – 280 с. – Текст: непосредственный.
4. Мамаев, М. А. Телекоммуникационные технологии (Сети TCP/IP): учеб. пособие / М. А. Мамаев. – Владивосток: ВГУ ЭиС, 2001. – 136 с. – Текст: непосредственный.
5. Олифер, В. Г., Олифер, Н. А. Компьютерные сети / В. Г. Олифер, Н. А. Олифер. – СПб.: Питер, 2003. – 863 с. – Текст: непосредственный.

Учебное издание

Новиков Александр Игоревич
Дятлова Елена Павловна

Операционные системы, сети и телекоммуникации

Учебно-методическое пособие

Редактор и корректор А. А. Чернышева
Техн. редактор Д. А. Романова Темплан 2024 г., поз. 5074/24

Подписано к печати 21.03.2024. Формат 60x84/16. Бумага тип № 1.
Печать офсетная. Печ.л. 6,9. Уч.-изд. л. 6,9.
Тираж 30 экз. Изд. № 5074/24 Цена «С». Заказ №

Ризограф Высшей школы технологии и энергетики СПбГУПТД,
198095, Санкт-Петербург, ул. Ивана Черных, 4.