

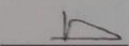
Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ МОРДОВСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ Н. П. ОГАРЁВА»

Институт электроники и светотехники  
Кафедра инфокоммуникационных технологий и систем связи

УТВЕРЖДАЮ

Зав. кафедрой

канд. техн. наук, доцент

  
(подпись) В. В. Никулин

«14» 06 2018г.

БАКАЛАВРСКАЯ РАБОТА

РАЗРАБОТКА КЛИЕНТСКОГО ПРИЛОЖЕНИЯ ДЛЯ УДАЛЕННОГО  
АДМИНИСТРИРОВАНИЯ ПК

Автор бакалаврской работы

  
(подпись)

14.06.18

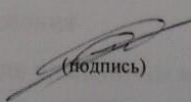
(дата)

Ю. Ю. Воробьев

Обозначение бакалаврской работы БР-02069964-11.03.02-05-18

Направление 11.03.02 Инфокоммуникационные технологии и системы связи

Руководитель работы

  
(подпись)

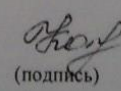
14.06.18

(дата)

С. Д. Шибайкин

канд. техн. наук, доцент

Нормоконтролер

  
(подпись)

14.06.18г.

(дата)

Е. А. Кошечая

канд. культурологии, доцент

Саранск

2018


Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ МОРДОВСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ Н. П. ОГАРЁВА»

Институт электроники и светотехники  
Кафедра инфокоммуникационных технологий и систем связи

УТВЕРЖДАЮ

Зав. кафедрой

канд. техн. наук, доцент

 В.В.Никулин  
(подпись)

«27» 11 2017г.

ЗАДАНИЕ НА БАКАЛАВРСКУЮ РАБОТУ

Студент Воробьев Юрий Юрьевич

1 Тема «Разработка клиентского приложения для удаленного администрирования ПК»

Утверждена приказом № 9516-с от «24» ноября 2017 г.

2 Срок предоставления работы к защите 06.06.2018 г.

3 Содержание выпускной квалификационной работы

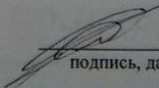
3.1 Обзор существующих приложений

3.2 Разработка проекта приложения

3.3 Реализация приложения для удаленного администрирования ПК


4 Приложения

Руководитель работы

 27.11.17  
подпись, дата

С. Д. Шибайкин  
инициалы, фамилия

Задание принял к исполнению

 27.11.17  
подпись, дата

Ю. Ю. Воробьев  
инициалы, фамилия

## РЕФЕРАТ

Выпускная квалификационная работа содержит 65 листов, 24 рисунка, 1 таблица, 15 источников.

**КЛЮЧЕВЫЕ СЛОВА:** КОДИРОВАНИЕ, АЛГОРИТМ, СЖАТИЕ, АНАЛИЗ, ПРОГРАММИРОВАНИЕ, КЛИЕНТ, СЕРВЕР.

Объект исследования – клиент-серверные технологии в удаленном администрировании.

Целью данной работы является разработка клиентского приложения для удаленного администрирования ПК.

Данная работа посвящена созданию и реализации клиентского приложения для удаленного администрирования ПК с использованием популярного языка программирования C#.

Были сформулированы необходимые требования, которым должна отвечать проектируемая система, на основе анализа существующих аналогичных программ для удаленного администрирования ПК. Была разработана архитектура приложения на основе клиент-сервер.

Разработка веб-ресурса была выполнена с использованием языка программирования C# (C Sharp), Windows Form, а также библиотеки AForge.Net, протокола TCP/IP.

					БР-02069964-11.03.02-05-18		
Изм.	Лист	№ докум.	Подпись	Дата			
Разраб.		Воробьев Ю.Ю.	<i>В.В.</i>	14.01.18	Лит.	Лист	Листов
Провер.		Шибайкин С.Д.	<i>С.Д.</i>	14.01.18		3	65
Реценз							
Н. Контр.		Кошечкина Е. А.	<i>Е.А.</i>	14.01.18	ИЭС, ИКТС, 431		
Утверд.		Никулин В. В.	<i>В.В.</i>	14.01.18			

Разработка клиентского приложения для удаленного администрирования ПК

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	5
1 Обзор существующих приложений для удаленного администрирования.....	7
1.1 TeamViewer.....	7
1.2 Radmin.....	9
1.3 Remote Manipulator System.....	12
2 Разработка проекта приложения.....	17
2.1 Общая структура приложения.....	17
2.2 Клиент-серверная архитектура.....	18
2.3 Этапы взаимодействия клиента и сервера.....	19
2.4 Способы сжатия графических данных.....	23
2.5 Анализ алгоритмов сжатия видео-данных.....	31
3 Реализация приложения для удаленного администрирования ПК.....	48
3.1 Серверная часть.....	48
3.2 Клиентская часть.....	48
3.3 Анализ и сравнение трафика.....	49
3.4 Выбор алгоритма шифрования.....	53
3.5 Сравнение методов сжатия изображения.....	54
ЗАКЛЮЧЕНИЕ.....	57
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	58
ПРИЛОЖЕНИЕ А (обязательное) СЕРВЕРНАЯ ЧАСТЬ.....	60
ПРИЛОЖЕНИЕ Б (обязательное) КЛИЕНТСКАЯ ЧАСТЬ.....	63

## ВВЕДЕНИЕ

В настоящее время компьютеры прочно вошли в жизнь современного производства. С их помощью выполняется большое количество задач и операций. Распространение многофункционального программного обеспечения с адаптацией к сетевым возможностям в большинстве приложений ведет к росту числа внедряемых в работу компьютеров. Они могут быть удалены друг от друга на внушительном расстоянии или стоять рядом в информационных центрах. Таким образом, использованные возможности обязаны быть эффективны для управления большим числом компьютеров как локально, так и удаленно. Но для бесперебойной работы этих сложных машин требуется профессиональное обслуживание квалифицированными специалистами, которые способны быстро и своевременно устранить возникающие неполадки. Особенно это касается офисных компьютеров, где время простоя может нанести значительные убытки производству. В связи с этим возникает необходимость в качественном техническом обслуживании.

Для оптимизации этого процесса многие системные администраторы используют незаменимый инструмент – удаленное администрирование. С его помощью можно не только уменьшить частоту посещения офисов и удаленных точек специалистом, но и увеличить скорость реагирования на заявки пользователей, что очень важно в современных реалиях.

Администрирование сети любой компании, как правило, выполняет собственный IT-отдел, специалисты которого, помимо обеспечения работы и безопасности сети, написания программ, закупки и ремонта оргтехники, занимаются технической поддержкой сотрудников не только головного офиса, но и филиалов.

Для эффективного и незамедлительного решения возникающих проблем необходимо не только использование эффективной и безопасной программы

					Лит	
Лит	Изм.	№ докум.	Подп.	Дата	БР-02069964-11.03.02-05-18	5

удаленного администрирования, но и руководство по работе в ней для решения основных задач администрирования.

					БР-02069964-11.03.02-05-18	Лист
Лит	Изм.	№ докум.	Подп.	Дата		6

# 1 Обзор существующих приложений для удаленного администрирования

## 1.1 TeamViewer

Существует немало приложений для организации удаленной работы с компьютерами. Большинство из них отличаются выраженной корпоративной направленностью, однако в ряде решений предлагаются варианты лицензирования, вполне приемлемые и для домашних пользователей, а программы для некоммерческого применения вообще бесплатны.

Разработчик – TeamViewer GmbH; размер дистрибутива – 3,43 Мбайт; работа под управлением – Windows 98/NT/2000/XP/Vista/7/8/8.1/10; имеются версии под Mac, Linux и мобильные системы (iOS, Android, Windows Phone, BlackBerry, Chrome OS); способ распространения – shareware; цена – определяется вариантом лицензии: Business – 29 900 руб. (неограниченное количество ПК), Premium – 59 200 руб., Corporate – 99 900 руб., для некоммерческого использования программа бесплатна. [1]

TeamViewer – одно из самых популярных в современном мире приложений для организации конференций и удаленного доступа к ПК, используя локальную сеть или Интернет. Программа дает возможность получить доступ к рабочему столу удаленного компьютера с полным управлением и записью сеанса подключения.

Другими функциями TeamViewer являются создание виртуальной частной сети, организация конференций и демонстраций, осуществляется поддержка голосового и видео общения между удаленными рабочими местами.

Для нормальной работы приложения не требуется настройка брандмауэра, маршрутизатора и прокси. Более того TeamViewer оптимизирует качество и скорость соединения в зависимости от вида подключения (мобильная сеть,

локальная сеть и так далее). Установленное соединение между узлами защищено с помощью обмена 1024 битными ключами RSA и 256 битными ключами сеанса подключения AES.

Программа состоит из одного инсталляционного модуля. Он устанавливается и на удаленном ПК, и на том, через который будет производиться администрирование. Также существует облегченная версия программы, которая не требует установки, а также особых прав администратора и мгновенно запускается, что ускоряет процесс установления соединения. Имеется «портативная» мобильная версия для смартфонов.

Процесс настройки удаленного доступа очень легкий и понятный: необходимо использовать идентификационные данные, присвоенные ПК в ходе установки приложения, для подключения к ПК администратора. Данная методика подключения устраняет проблемы с выяснением IP-адреса, перенаправлением портов и прочими вещами, неизвестными простому пользователю. Но это не означает, что нельзя устанавливать локальное соединение с использованием IP, необходимо лишь определенным образом настроить программу. Для идентификации также необязательно использовать длинный ID, вместо него можно использовать логин учетной записи Windows и соответствующий ему пароль. (рисунок 1.1). [2]

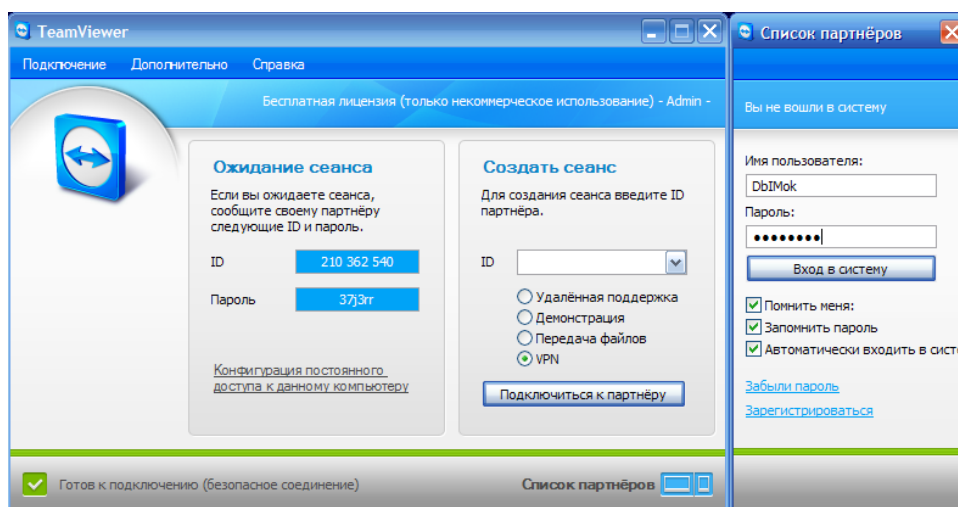


Рисунок 1.1 – Подключение к удаленному ПК с помощью TeamViewer



## 1.2 Radmin

Разработчик – «Фаматек»; способ распространения – shareware; цена – определяется вариантом лицензии: «Стандартная» – 1250 руб. за удаленный компьютер, «Корпоративная» (от 200 лицензий) – от 550 руб. и ниже за лицензию (зависит от количества удаленных ПК).

Radmin – довольно распространенный инструмент для проведения удаленного администрирования ПК. Он был признан многими специалистами, чей род деятельности напрямую связан с обеспечением работоспособности ПК, находящихся на удаленном расстоянии. Работа через Radmin возможна как через Интернет, так и через локальную сеть. Среди возможностей продукта можно отметить совместимость с ОС Linux, работу в полном доступе с правами администратора, режим просмотра, функция обмена данными, управление состоянием компьютера (выключение, перезагрузка, включение, доступ к BIOS), поддержка работы с буфером обмена в двустороннем порядке, командная строка, Telnet. (рисунок 1.2). [3]

В Radmin возможно подключение более 1 пользователя к демонстрации рабочего стола, поддерживает переключение между различными сессиями, а также оптимизирует передаваемый трафик. Безопасная работа осуществляется с помощью стандарта шифрования AES, аутентификация пользователей с помощью средств безопасности Windows, а также собственная система учетных записей с паролями. Существует фильтрация IP-адресов для ограничения доступа некоторым хостам.

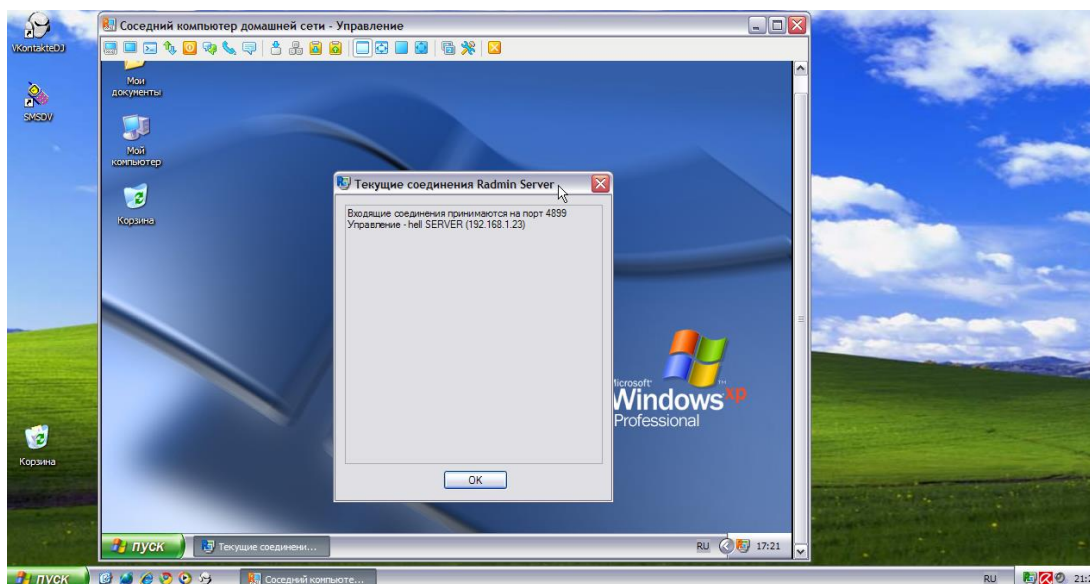


Рисунок 1.2 – Управление удаленным ПК в среде Radmin

Radmin состоит из двух модулей: клиентского модуля Radmin Viewer и серверного модуля Radmin Server. Модуль Radmin Viewer устанавливается на локальном компьютере, через который будет происходить управление удаленным рабочим столом. Radmin Server же устанавливается на удаленном компьютере. Обязательным условием для обоих компьютеров является соединение по протоколу TCP/IP в локальной сети, по модему или через Интернет. Установка приложения не является сложной, трудностей не с ней не возникает. В первую очередь устанавливается Radmin Server на удаленном компьютере. В процессе установки модуля потребуется задать систему безопасности, которая будет применена, затем выполнить настройку прав доступа для отдельных пользователей (рисунок 1.3).

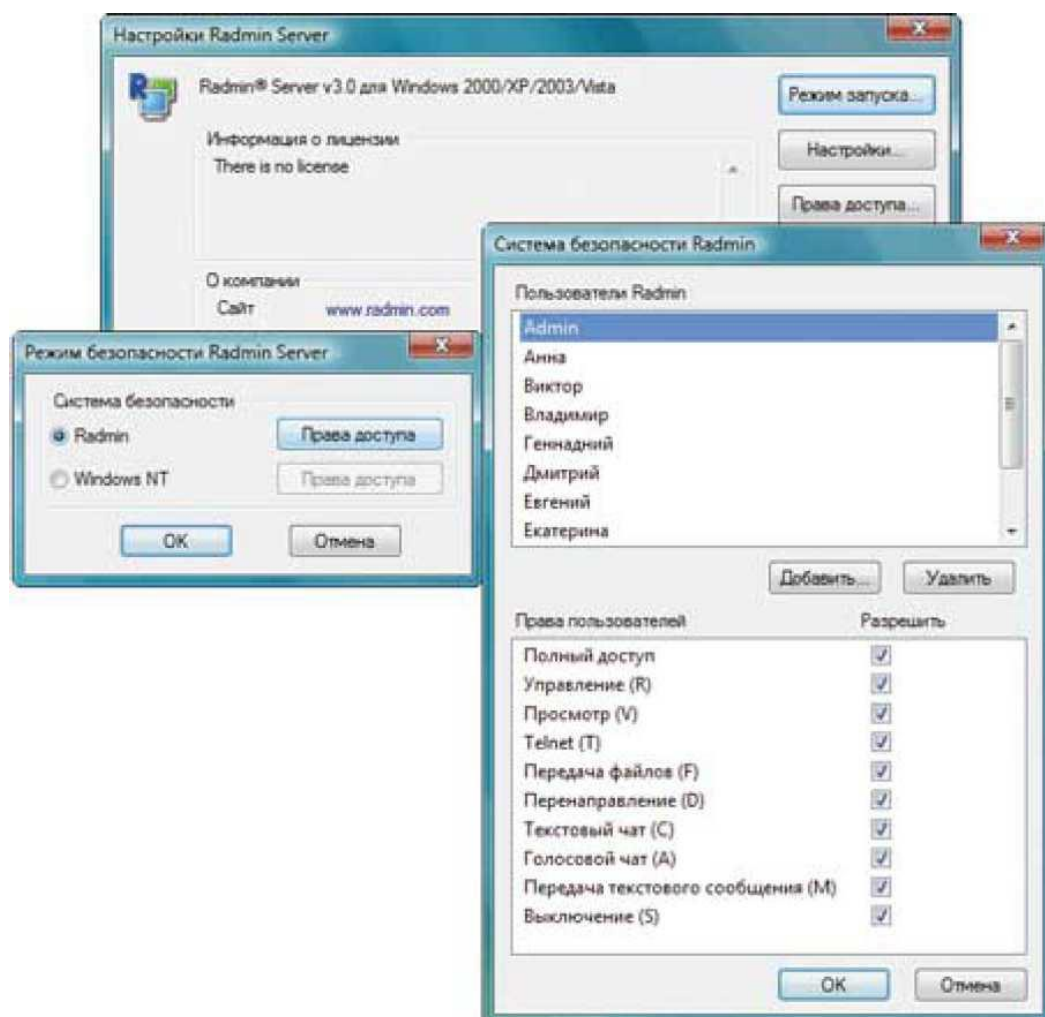


Рисунок 1.3 – Настройка прав пользователей в модуле Radmin Server

При необходимости установки программы Radmin Server на большое число компьютеров можно воспользоваться функционалом для удаленной установки. После этого запускается модуль Radmin Viewer на локальном компьютере и настраивается одно или более соединений с удаленными ПК (команда «Новое соединение» в меню «Соединение»), для чего необходимо указать IP-адрес удаленного ПК или его DNS-имя (рисунок 1.4).

Если удаленный компьютер имеет виртуальный IP-адрес или недоступен напрямую из Интернета, либо в рамках одного сегмента локальной сети, то все оказывается гораздо сложнее. Тогда дополнительно необходима настройка перенаправления портов на маршрутизаторе/прокси-сервере, который установлен на пути между удаленным компьютером и Интернетом.

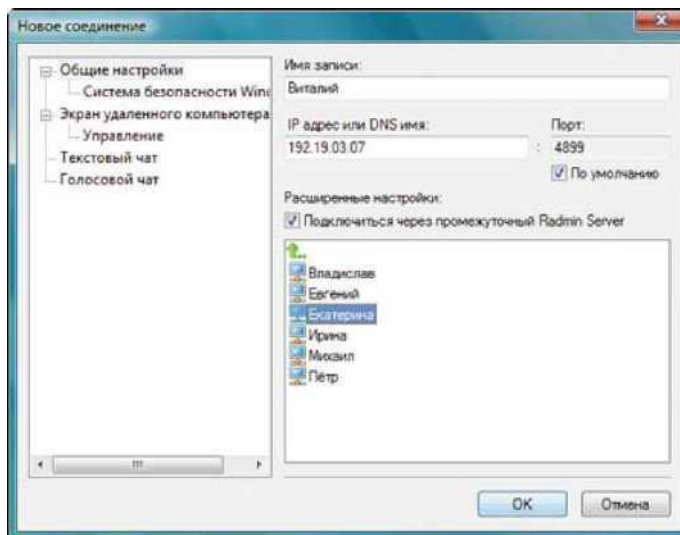


Рисунок 1.4 – Настройка нового подключения в Radmin

Другим вариантом является установка виртуального TCP/IP-соединения или туннеля между двумя компьютерами, которые не связаны напрямую, с применением программ сторонних разработчиков. Для установления соединения с удаленным ПК нужно дважды щелкнуть на нужном соединении. [4]

Программа позволяет подключаться к удаленному компьютеру в разных режимах:

- управление;
- просмотр;
- telnet;
- передача файлов;
- выключение;
- текстовый чат;
- голосовой чат.

### 1.3 Remote Manipulator System

Remote Manipulator System (RMS) – программа для удаленного администрирования через Интернет и в локальной сети.

Продукт совместим с операционной системой Linux. Решение предлагает более десяти режимов подключения к удаленному компьютеру и может использоваться для удаленного управления ПК, удаленного наблюдения, осуществления базовых операций с файлами между локальным и удаленным компьютерами, записи экрана по расписанию, подключения в режиме командной строки, управления питанием. [5]

RMS состоит из двух основных модулей: модуль-клиент, модуль-хост. Модуль «Клиент» предназначен для подсоединения к удаленным рабочим станциям, на которых установлен Хост. Клиент предоставляет удобный пользовательский интерфейс для управления списком соединений, построения карты сети, поиска удаленных рабочих станций и управления ими в различных режимах. «Хост» должен быть установлен на каждой удаленной рабочей станции, к которой нужно получить доступ. Этот модуль позволяет удаленно управлять компьютером, на котором он установлен. Возможна удаленная установка клиентских модулей.

Клиент (Viewer) – это приложение, используемое техническими специалистами или администраторами для установки сеанса удаленного доступа. Viewer позволяет управлять списком удаленных компьютеров и устанавливать с ними соединение в любом из доступных режимов (рисунок 1.5).

Хост необходимо устанавливать на каждый удаленный компьютер (возможна удаленная и автоматизированная массовая установка). Он более универсален, чем Агент (см. ниже), и обеспечивает как контролируемый удаленный доступ, так и не контролируемый (т.е. доступ, не требующий присутствия человека за удаленным компьютером).

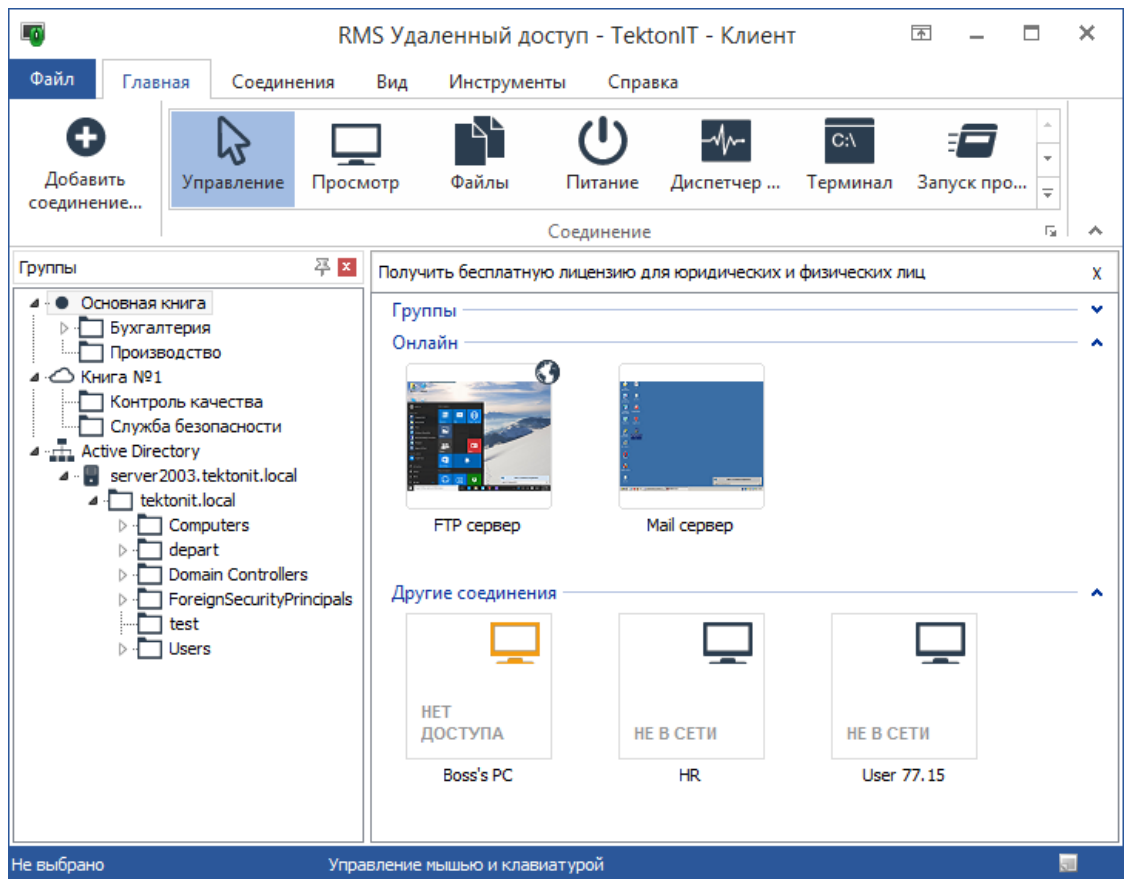


Рисунок 1.5 – Главное окно модуля «Клиент»

Так как Host работает как системная служба, то можно совершать любые действия с удаленным ПК без каких-либо дополнительных настроек, достаточно просто того, чтобы компьютер был включен и в сети (рисунок 1.6). [6]

Предусмотрены аудио и видеочат, обмен текстовыми сообщениями между компьютерами. Remote Manipulator System поддерживает работу с несколькими мониторами и может быть настроен на оптимальную производительность с учетом плюсов и минусов сети и рабочих станций, на которых будет установлен.

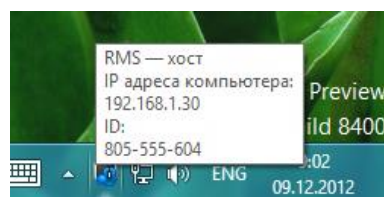


Рисунок 1.6 – Значок хоста в области уведомлений

Благодаря применению технологии Internet-ID, это решение может работать через сетевые экраны и NAT без дополнительных настроек маршрутизаторов и другого сетевого оборудования. Это означает, что для установки соединения необязательно знать IP-адрес удаленного компьютера – достаточно указать его ID.

Поддержка технологий «Обратное соединение» и «Соединение через» (так называемое каскадное соединение) дает возможность настроить RMS на работу в весьма сложной топологии сети с множеством маршрутизаторов и подсетей. Все данные, передаваемые компьютером по сети, шифруются с использованием алгоритмов шифрования RSA и AES, а аутентификация пользователей осуществляется по паролю либо посредством системы безопасности Windows с поддержкой Active Directory. Кроме того, реализована возможность IP-фильтрации, что позволяет ограничивать доступ определенных хостов и подсетей по стратегиям белого и черного списков. [7]

Технология подключения удаленного ПК зависит от того, известен IP-адрес удаленного компьютера или нет. Если IP-адрес известен, то после запуска R-Server на удаленном компьютере достаточно добавить новое соединение и указать его название и IP-адрес либо DNS-имя.

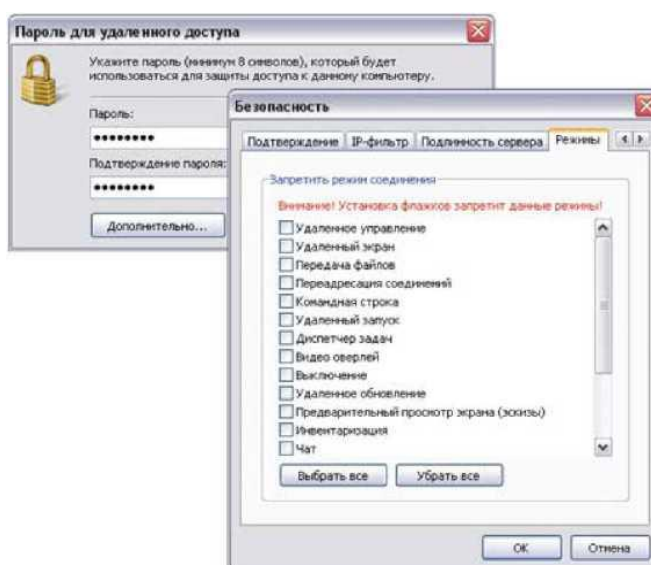


Рисунок 1.7 – Настройка модуля «Сервер» RMS

В противном случае на удаленном ПК после запуска R-Server потребуется выбрать команду «Настройка Internet-ID соединения», в открывшемся окне щелкнуть по кнопке «Получить новый ID» – после этого R-Server окажется готов принимать соединения от R-Viewer в режиме Internet-ID. В свою очередь, на администраторском компьютере, как и в предыдущем варианте, создается новое соединение, вместо IP-адреса для него указывается полученный на удаленном ПК идентификатор (рисунок 1.8).

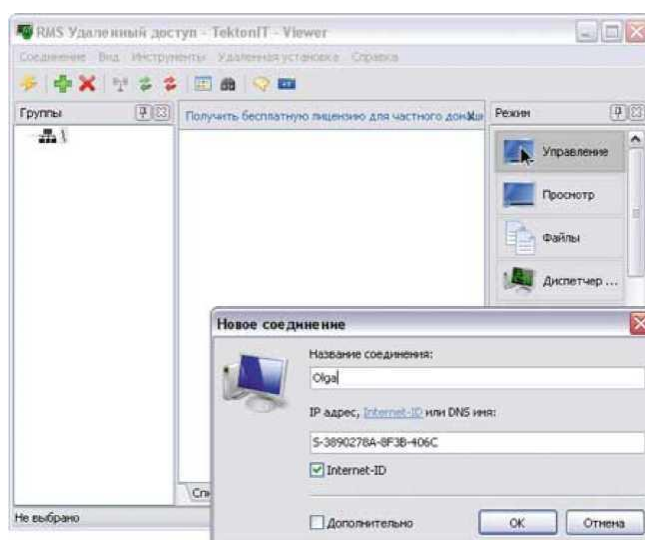


Рисунок 1.8 – Подключение к удаленному ПК в среде RMS



## 2 Разработка проекта приложения

### 2.1 Общая структура приложения

Клиент-серверная архитектура приложения подразумевает наличие у предприятия собственного сервера. Такое решение имеет ряд преимуществ:

- все передаваемые данные не покидают пределов предприятия;
- приложение будет работать даже при отсутствии подключения к сети Интернет;
- «прозрачность» – можно быть уверенным, что удаляемые данные действительно удалены, чего нельзя ожидать в случае, если разработчик предоставляет только клиентскую часть, а серверная остается подведомственной разработчику и/или другим сторонним лицам.

Обмен сообщениями производится с использованием протокола SSL, кроме того, поток данных между клиентом и сервером дополнительно шифруется на основе улучшенного алгоритма шифрования AES. Такое двойное шифрование обезопасит данные в случае успешной атаки злоумышленника на протокол SSL, или взлома сервера (на сервере история сообщений хранится в зашифрованном виде).

Новый клиент не имеет доступа к функциональности приложения, пока не будет одобрен на стороне сервера уполномоченным лицом. Такое решение исключает возможность получения доступа злоумышленника к данным пользователей.

Конечно, все вышеперечисленное не исключает полностью возможность взлома, но, по мнению разработчика, значительно его затрудняет. Руководство компании должно понимать, что нельзя полностью полагаться на средства криптографии. Необходимо регулярно проводить с сотрудниками профилактические беседы, в которых объясняется опасность хранения паролей на незащищенных носителях, к примеру, бумажных, в особенности, на рабочем месте.

К сожалению, человеческий фактор на сегодняшний день остается основной угрозой конфиденциальности информации, а главная тому причина - неосведомленность сотрудников.

## 2.2 Клиент-серверная архитектура

Приложение является многопользовательским и предназначено для использования в сети. Традиционно для такого рода приложений применяются два подхода: централизованное управление передачей сообщений между пользователями или распределение функций синхронизации сообщений между всеми участниками сетевого взаимодействия.

Первый подход является, по сути, применением технологии «клиент-сервер». При этом подходе создается выделенный сервер, который обслуживает запросы клиентов, хранит данные для управления состоянием клиентов и выполняет синхронизацию данных между ними. Он отвечает за прием данных от каждого клиента и распространение этих данных между другими. В этом случае все клиенты являются равноправными относительно распределения нагрузки и взаимодействуют исключительно с сервером.

Иногда серверная часть может включаться в клиентское приложение. В этом случае происходит взаимодействие типа «клиент-клиент». Тогда один из клиентов по предварительной договоренности несет на себе функции сервера, в результате являясь и клиентом, и сервером. Такая реализация в случае большого количества пользователей приводит к излишней нагрузке на одного из участников. Это, несомненно, является узким местом такого подхода. Еще одним из его недостатков является «утяжеление» клиентской части. В дополнение к сказанному заметим, что совмещение функций клиента и сервера в одном приложении может быть оправдано при реализации подключения «точка-точка», когда обе стороны в разное время могут меняться ролями.

Второй подход основан на сетевом взаимодействии без использования

сервера, как такового. В этом случае каждый участник коммуникационного процесса сам осуществляет рассылку данных всем остальным участникам, например, посредством широковещания. Это позволяет сэкономить требуемые на обеспечение поддержки сервера ресурсы. Но такая реализация, в основном, подходит для коммуникации в условиях локальной сети.

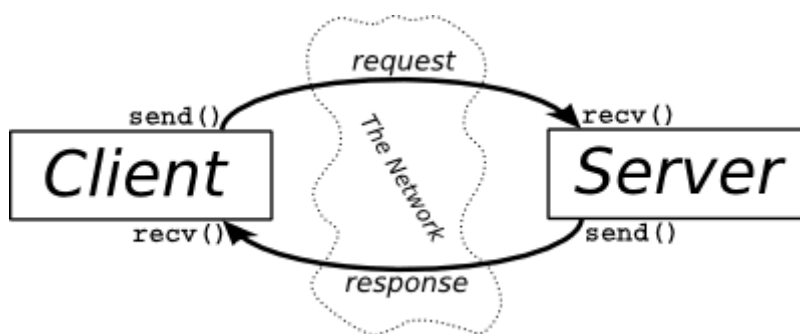


Рисунок 2.1 – Взаимодействие клиента и сервера

### 2.3 Этапы взаимодействия клиента и сервера

Взаимодействие между клиентом и сервером основано на вызовах методов сервера и обмене информационными сообщениями, которые организуются в очередь. В рассматриваемом случае клиент и сервер используют протокол TCP для обмена данными. Обмен данными, ориентированный на соединения, может использовать надежную связь, для обеспечения которой протокол уровня 4 посылает подтверждения о получении данных и запрашивает повторную передачу, если данные не получены или искажены. Протокол TCP использует именно такую надежную связь. TCP используется в таких прикладных протоколах, как HTTP, FTP, SMTP и Telnet.

Протокол TCP требует, чтобы перед отправкой сообщения было открыто соединение. Серверное приложение должно выполнить так называемое пассивное открытие (*passive open*), чтобы создать соединение с известным номером порта, и, вместо того, чтобы отправлять вызов в сеть, сервер переходит в ожидание поступления входящих запросов. Клиентское приложение должно

выполнить активное открытие (active open), отправив серверному приложению синхронизирующий порядковый номер (SYN), идентифицирующий соединение. Клиентское приложение может использовать динамический номер порта в качестве локального порта.

Сервер должен отправить клиенту подтверждение (ACK) вместе с порядковым номером (SYN) сервера. В свою очередь клиент отвечает ACK, и соединение устанавливается.

После этого может начаться процесс отправки и получения сообщений. При получении сообщения в ответ всегда отправляется сообщение ACK. Если до получения ACK отправителем истекает тайм-аут, сообщение помещается в очередь на повторную передачу. TCP – это сложный, требующий больших затрат времени протокол, что объясняется его механизмом установления соединения, но он берет на себя заботу о гарантированной доставке пакетов, избавляя нас от необходимости включать эту функциональную возможность в прикладной протокол.

Протокол TCP имеет встроенную возможность надежной доставки. Если сообщение не отправлено корректно, мы получим сообщение об ошибке. [8]

Перед тем, как установить соединение, сервер создает сокет и устанавливает локальную точку для прослушивания подключений. На следующем этапе запускается прослушивание входящих соединений и дальнейшее принятие необходимого входящего соединения. После установления соединения начинается отправка/получение данных. После завершения сессии сокет закрывается, а сервер переходит в режим прослушивания и принятия соединения (рисунок 2.2).

На сокете указывается порт и адрес, соединения с которых будут прослушиваться с помощью метода Listen. Как только на сокет будут поступать входящие соединения, они будут приниматься с помощью метода Accept, который начинает обрабатывать стоящий первым в очереди запрос. При помощи

метода `Receive` сервер начинает получать данные в виде массива байтов, а также возвращает количество полученных данных.

В случае, если отсутствуют данные для обработки, метод `Receive` блокирует входящий поток. После получения данных клиенту отсылается запрос при помощи метода `Send`, который содержит в себе количество принятых байтов.

В конце обработки запроса сокет закрывается для входящих соединений.



Рисунок 2.2 – Схема работы сервера на основе TCP протокола

Работа клиента на основе TCP протокола отличается от сервера тем, что после создания сокета вызывается метод `Connect`, в который передается адрес и порт сервера. (рисунок 2.3).[9]



Рисунок 2.3 – Работа клиента на основе TCP протокола

В общем виде взаимодействие клиент-сервер представляет собой принцип, по которому взаимодействие начинает клиент, сервер лишь отвечает клиенту и сообщает о том может ли он предоставить услугу клиенту и если может, то на каких условиях. Клиентское программное обеспечение и серверное программное обеспечение обычно установлено на разных машинах, но также они могут работать и на одном компьютере.

Данная концепция взаимодействия была разработана в первую очередь для того, чтобы разделить нагрузку между участниками процесса обмена информацией, а также для того, чтобы разделить программный код поставщика и заказчика. Ниже представлена упрощенная схема взаимодействия клиент-сервер (рисунок 2.4).

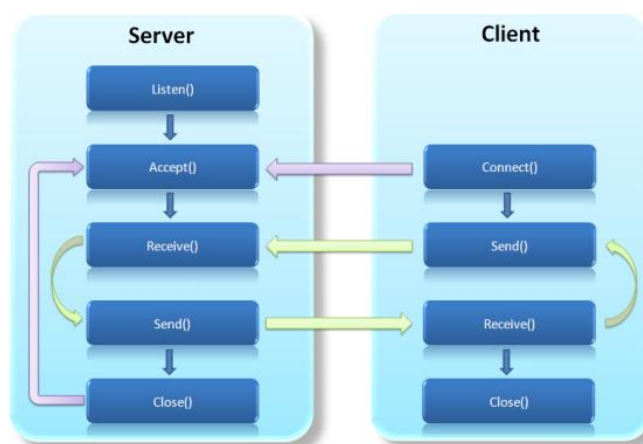


Рисунок 2.4 – Схема взаимодействия клиент-сервер

## 2.4 Способы сжатия графических данных

Существует две большие группы алгоритмов, которые делятся на:

- алгоритмы сжатия без потерь;
- алгоритмы сжатия с потерями.

Если говорить о сжатии без потерь, то необходимо иметь ввиду, что есть такой алгоритм, который является обратным алгоритму сжатия. Данный алгоритм позволяет очень точно восстановить исходное изображение из сжатого. При этом для алгоритмов сжатия с потерями не предусмотрен обратный алгоритм. В этом случае регенерирующий алгоритм восстанавливает изображение, которое не обязательно точно совпадает с исходным. В основном подбор алгоритмов сжатия и восстановления осуществляется таким образом, чтобы сжатие имело высокую степень, но при этом сохранялось хорошее визуальное качество изображения.

Процесс сжатия файла можно разделить на два этапа: запись количества видеопикселей, запись их цвета. В некоторых программах данные пункты могут быть представлены наоборот. Из-за этой причины появляются такие ситуации, что программа, считывающая файл, обрабатывает данные не в порядке,

отличающемся от программы, которая производила сжатие. Если при открытии файла появляется уведомление об ошибке или открывается полностью искаженное и нечитаемое изображение, нужно попробовать открыть этот файл или изображение с помощью другого софта или преобразовать его в другой формат.

Сжатие методом RLE (Run Length Encoding – кодирование с переменной длиной строки) наиболее эффективно в случае, когда изображения содержат большие области, закрашенные однотонно, и наименее эффективно в случае отсканированных фотографий, потому что в них может не быть однотонных длинных последовательностей видеопикселей.

Метод сжатия LZW (Lempel-Ziv-Welch) основан на поиске узоров в изображении, которые повторяются. Рисунки, в которых содержатся очень много узоров, могут быть сжаты до 0,1 от их первоначального объема. Метод сжатия LZW используется в файлах с такими форматами, как TIFF и GIF. Главное отличие форматов в том, что данные с форматом GIF сжимаются всегда и в любом случае, а TIFF оставляет право выбора сжатия пользователю. Существуют различные вариации формата TIFF, они используют разные методы компрессии. Из-за наличия большого разнообразия схем сжатия отдельные версии формата TIFF могут быть несовместимыми. Получается, что не все форматы TIFF одинаковы, поэтому может не получиться прочитать файл в формате TIFF в графической программе, которая должна поддерживать данный формат. Но, данная проблема не является критической, а TIFF является одним из самых популярных растровых форматов в современном мире.

Метод сжатия LZW придуман и реализован в 1978 году израильянами Лемпелом и Зивом, и доработан позднее в США. Процесс сжатия происходит следующим образом: происходит поиск совпадающих последовательностей (так называемые фразы фразы) по всему файлу; далее такие последовательности сохраняются в таблице, где каждой из них присваивают ключи, то есть



маркеры более короткого размера. Например, если в файле имеются последовательности оранжевого, розового и зеленого пикселей, которые повторяются 50 и более раз, метод сжатия определяет это, а затем присваивает каждому отдельному набору целое число (например, 8). Следующим этапом он сохраняет данные 50 раз в виде числа 8. Метод LZW лучше действует на однородных, свободных от шума цветов участках, он отлично действует при сжатии любых графических данных. Но вместе с этим кодирование и распаковка происходит медленнее по времени.

Пользователь может управлять уровнем допустимых потерь, указывая нужную степень сжатия. Благодаря этому, можно выбрать самый подходящий режим обработки каждого файла. Пользователю предоставляется возможность установки гибких настроек, исходя из которых можно сделать выбор между экономией места на носителе или качеством файла. Если сохраняемый файл будет фотографией, которая представлена для высокохудожественного издания, то в первую очередь внимание оказывается качеству, так как в этом случае рисунок должен быть воспроизведен, как можно точнее и качественнее. Если же изображением является фотографией, которая предназначена для размещения на любительской поздравительной открытке, то в случае потери некоторой части исходной информации не произойдет ничего страшного, ведь она не несет большого значения. Для отдельных изображений необходимо производить эксперименты, которые определяют наиболее допустимый уровень потерь.

BMP (Windows Device Independent Bitmap) является наиболее простым растровым форматом изображений. Другое его название – DIB, он является встроенным форматом Windows, который поддерживается абсолютно всеми редакторами графиками, работающими под управлением операционной системы. В BMP данные о цвете хранятся только в модели RGB, поддерживаются как индексированные цвета (до 256 цветов), так и полноцветные изобра-

жения, причем в режиме индексированных цветов возможна простейшая компрессия RLE. Без компрессии объем файла оказывается максимально близко к исходному. Благодаря простейшему алгоритму записи изображения при сжатии файлов формата BMP системные ресурсы расходуются очень немного. Данный формат очень часто используют для логотипов, иконок, экранных заставок и других небольших элементов оформления программ и сайтов.

PCX (Soft Publisher's Paintbrush) – схожие возможности, как и у BMP, есть и у формата PCX, который был разработан фирмой Z-Soft для собственного графического редактора PC PaintBrush. Цветовые возможности: 1, 2, 4, 8 или 24- битовый цвет, поддержка схемы RGB. Но нет возможности сохранения изображения монохромного в серых оттенках. Как и BMP, этот формат в современном мире устарел. Новейшие графические редакторы совместимы с данным форматом только для поддержания совместимости с антикварным и старым софтом.

GIF (Graphics Interchange Format) формат является самым популярным в современное время форматом на просторах сети. Формат был предложен фирмой CompuServe еще в 1987 году. Особенностью формата является то, что используется режим индексирования не более 256 цветов. Данный факт ограничивает сферы применения данного формата изображениями, у которых существуют резкие переходы цветов. Особыми любителями использования формата GIF являются веб-мастера, которые сохраняют с его помощью различные иконки и другие элементы для оформления страниц.

Благодаря применяющемуся алгоритму сжатия без потерь LZW качества файлы имеют небольшой размер. Их можно пересылать по каналам связи очень быстро и надежно. К числу отличий относятся: режим проявления постепенного изображения, при котором графические строки выводятся в определенном порядке, а не подряд. Таким образом, до полной загрузки изображения GIF можно понять суть и смысл изображения и в случае необходимости можно прервать его загрузку, а полное изображение показывается в четыре

этапа. В 1989 году формат был обновлен и получил наименование GIF89A. В новой версии присутствует дополнительный альфа-канала, который реализует эффект прозрачности (но не больше одной градации), а также возможность хранить в одном файле более одной картинки, которые могут быть показаны с нужным временем показа. Эта особенность понравилась создателям анимированных рекламных баннеров.

Так как в основе GIF формата лежит алгоритм сжатия LZW, патент на который принадлежит американской компании Unisys, которая требует выплаты за использования их технологии, то в последнее время формат уходит и постепенно устаревает. К тому же не способствует длительному жизненному циклу формата невозможность отображения более чем 256 цветов в изображении. На сегодняшний день самый вероятный преемник GIF – формат PNG.

Формат PNG (Portable Network Graphics) был разработан сообществом независимых программистов. Он появился после того, как популярнейший формата GIF стал коммерческим продуктом. Этот формат так же, как и GIF, TIFF сжимает изображения без потери качества, но имеет особую отличительную черту: графическая информация сжимается в обеих плоскостях (в горизонтальной и вертикальной). Это обеспечивает более высокую степень сжатия, а также возможность поддержки растровых изображений с цветовой гаммой до 48 битов включительно.

Единственным недостатком отмечается, что формат не имеет возможности создания анимационных изображений. Но данная проблема уже потеряла свою актуальность, так как для создания анимации используется технология Flash. При этом формат PNG позволяет создавать изображения, у которых 256 уровней прозрачности с применением дополнительного альфа-канала с 256 градациями серого. Благодаря этому формат заметно отличается от других конкурентов на «рынке» форматов сжатия. Другими особенностями технологии является, во-первых, двумерная чересстрочная развертка, из-за которой изображение появляется и по строкам, и по столбцам. Вторая отличительная

черта – встроенная коррекция гаммы, она позволяет сохранять и открывать изображения на различных устройствах и платформах без потери яркости.

Несмотря на все преимущества, формат PNG не подходит для печати из-за отсутствия места для типа разрешения в заголовке. Для данных целей целесообразнее использовать PSD или TIFF. Изначально PNG создавался как формат для Интернета, он хорошо подходит для хранения и передачи графической растровой графики высокого качества через глобальную сеть.

Но, как у других форматов, существует ряд недостатков, которые сдерживают его развитие. Так, формат PNG значительно уступает формату GIF в случае, если речь идет о мелких элементах графического оформления страниц в сети (кнопки, рамки и тому подобное). Проблема заключается в следующем: около 1 Кб информации в файле – описание палитры цветов. Иногда этот размер сопоставим с размером всего изображения.

JPEG (Joint Photographic Experts Group) является самым популярным форматом для хранения фото изображений, он является общепризнанным форматом в сети Интернет. JPEG может хранить только 24-битовые полноцветные изображения. В нем реализуется сложный алгоритм сжатия, который основан на особенностях восприятия человека его зрением (используются блоки пикселей 8x8, содержащие один цвет, а также информацию о яркости, метод Хаффмана и другие параметры, зависящие от степени). Сжатие JPEG происходит с потерями качества, но степень компрессии может быть легко настроена на минимум, что делает потери незаметными для глаза. Если предварительно размыть изображение, то можно усилить сжатие, при этом минимизировав потери качества. Однако формат JPEG не подходит для хранения изображений, которые в будущем будут подвергнуты обработке. Это связано с тем, что при каждом сохранении файла или документа в JPEG формате, ухудшение качества происходит с лавинообразной скоростью и характером.

Лучший вариант – редактирование изображения в любом другом формате, а по завершении всех работ – окончательно сохранять в JPEG. Таким

образом, сохраняется хорошее качество изображения при минимальном размере конечного файла.

Метод сжатия JPEG обеспечивает высокий коэффициент сжатия (вплоть до 100:1) для изображений качества фотографий. Формат JPEG был разработан объединенной группой экспертов по фотографии. Высокий коэффициент компрессии достигается благодаря сжатию с потерями. В таком случае часть исходной информации теряется из конечного полученного после сжатия файла.

Метод JPEG основан на явлении, при котором человеческий глаз чувствительнее к изменению яркости, чем к изменению цвета. Поэтому при сжатии этим методом запоминается информация о разнице между яркостями видеопикселей в большем объеме, а о разнице между их цветами – в меньшей степени. Так как минимальные различия в цвете соседних пикселей по теории вероятности сложно заметить, то изображение после декомпрессии выглядит почти таким же, как до сжатия.

Существует еще одна разновидность формата JPEG, которая называется "прогрессивный JPEG" (p-JPEG). Он отличается от обычного тем, что при выводе его на экран содержимое изображения появляется почти сразу, но в низком качестве. По мере загрузки качество улучшается до исходного, данный процесс похож на показ GIF. В формате JPEG не поддерживается анимация, а также прозрачный цвет. Используется в основном для публикации высококачественных полноцветных изображений (например, фотография) в глобальной сети.

Формат TIFF (Tag Image File Format) был разработан компанией Aldus для своего графического редактора PhotoStyler. Как универсальный формат для хранения растровых изображений, TIFF достаточно широко используется, в первую очередь, в издательских системах, требующих изображения наилучшего качества. Кстати, возможность записи изображений в формате TIFF яв-

ляется одним из признаков высокого класса современных цифровых фотокамер. Формат поддерживает множество алгоритмов сжатия (в том числе популярные LZW, Deflate или JPEG), типов изображений от битового (1-, 2-, 4-, 8-, 24- и 32-битные изображения) и индексированных цветов до LAB, CMYK и RGB (кроме дуплексов и многоканальных документов). Со сжатием LZW файл TIFF занимает почти столько же места, сколько и GIF, только, в отличие от последнего, TIFF поддерживает полноцветные изображения и хранит в своем теле подробную информацию об изображении – разрешение, тип принтера и другие детали, необходимые для профессиональной работы с изображениями. В этом формате поддерживаются такие чисто профессиональные возможности, как обтравочные контуры, альфа-каналы, возможность сохранять несколько копий изображения с разным разрешением и даже включать в файл слои. Формат TIFF очень удобен при переносе изображений между компьютерами различных типов. [10]

Формат PSD (Adobe Photoshop) является стандартным форматом в пакете приложения Adobe Photoshop. Он отличается возможностью хранения нескольких слоев. Он включает в себя много дополнительных переменных (и по количеству не уступает TIFF), также сжимает изображения при помощи алгоритма сжатия без потерь. В некоторых случаях даже сильнее, чем PNG (только при условии, что размер файла более десятка или даже сотни мегабайт). Глубина цвета в формате составляет до 16 бит на канал (48-битные цветные и 16-битные черно-белые), а также альфа-каналы, контуры, слои, прозрачность, векторные надписи и так далее. Прекрасно подходит для хранения и передачи изображений, которые содержат специальные, свойственные только Adobe Photoshop, элементы.

Большинство стандартных программ для просмотра легко открывают и поддерживают данный формат PSD. Но при открытии их не в Adobe PhotoShop можно потерять специфические элементы и возможности. [11]

## 2.5 Анализ алгоритмов сжатия видео-данных

Основной сложностью при работе с видеоданными являются большие объемы свободного места на носителях, на которых будет осуществляться их хранение. Даже короткие отрезки с видео занимают значительные объемы пространства, а использование современных способов и алгоритмов сжатия не позволяют исправить ситуацию полностью. К примеру, на обычных «компакт-диск» можно записать более тысячи фотографий, около 9-10 часов музыки и лишь 30 минут видео, а, говоря о потоках телевизионных записей с частотой в 25 кадров/сек и разрешением 720x576, необходима пропускная способность канала связи в 240 Мб/с. Использование известных алгоритмов компрессии изображений, позволяющих снизить требование по пропускной способности в 10 раз, не спасает, так как вес видео данных все равно остается большой величиной.

В итоге большинство используемых сегодня алгоритмов сжатия видео – алгоритмы с потерей данных. При сжатии обычно применяются следующие типы избыточности:

- когерентность участков изображения – незначительное изменение цвета картинка в ближних;
- избыточность в цветовых плоскостях – для восприятия большую роль играет яркость изображения;
- подобие между кадрами – используется свойство незначительного изменения кадров при скорости в 25 кадров в секунду.

Первые два пункта достаточно известны и популярны в алгоритмах сжатия графических данных. Этот факт означает, что происходит кодирование не нового кадра, а разности между предыдущим и нынешним. Более сложным методом является нахождение для каждого отдельного блока в кадре после сжатия самого менее отличающегося от него блока в исходном кадре, который используется в качестве основного.

Далее происходит кодирование разницы между этими блоками. Этот метод является более ресурсоемким по сравнению с перечисленными.

Как и в случае с алгоритмами кодирования, так и в случае с видео сжатием существуют определенные требования.

Произвольный доступ – означает наличие возможности за ограниченное время найти и показать определенный. Данное требование реализуется вхождением в потоке данных точек входа. Это кадры, сжатые независимо (в качестве статического изображения). 0,5 с – нормальное время поиска произвольного кадра.

Быстрый поиск вперед/назад – данная возможность применяется в проигрывателях. Она подразумевает под собой быстрый показ кадров, которые не стоят в соседних положениях в исходном потоке.

Показ кадров в обратном направлении – данная функция очень редко нужна в приложениях. При жестких временных ограничениях показа следующего кадра выполнение этой функции может значительно уменьшить степень компрессии.

Аудиовизуальная синхронизация – наиболее серьезное требование. Для того, чтобы добиться синхронности аудио и видео дорожек, необходимы специальные данные, которые существенно увеличивают размер видеофайла. Видеосистема воспринимает это таким образом, что если не успеть достать и показать в нужный момент времени определенный кадр, то необходимо уметь корректно воспроизвести следующий за ним кадр. Если показывать видео без звука, то допустим более медленный или быстрый показ.

Устойчивость к ошибкам – требование, которое обусловлено тем, что преимущественно каналы связи ненадежны. Искаженное изображение из-за помех должно быстро восстанавливаться. Требование достаточно просто удовлетворяется определенным числом специальных независимых кадров в потоке. При этом степень сжатия уменьшается. Это вызвано тем, что на экране 2–3 с (50–75 кадров) может показываться одно и то же изображение. Но



необходимо все равно нагружать поток независимыми кадрами для создания устойчивости к ошибкам.

Время кодирования и декодирования – во многих видеосистемах общая задержка на процесс кодирования, передачи и декодирования должна быть не более 150 мс. Кроме того, в приложениях, поддерживающих редактирование, время реакции системы для нормальной интерактивной работы должно составлять не более 1 с.

Редактируемость – под этим понимается возможность изменять все кадры. Это должно происходить также легко, как если бы они были записаны независимо.

Масштабируемость – простота реализации концепции «видео в окне». Необходимо быстро изменять высоту и ширину изображения в пикселах. Масштабирование способно вызвать неприятные эффекты в алгоритмах, основанных на дискретном косинусном преобразовании. Корректно реализовать эту возможность для MPEG можно только при условии достаточно сложных аппаратных реализаций. В таком случае алгоритмы масштабирования не будут заметно увеличивать время декодирования.

Масштабирование достаточно просто осуществляется во фрактальных алгоритмах. В них изображение не распадается на квадраты (пикселы), т. е. отсутствует эффект «зернистости» при увеличении изображения в несколько раз. Если необходимо уменьшать размер изображения, то хорошим вариантом становится использование алгоритмов, которые основаны на wavelet-преобразовании.

Небольшая стоимость аппаратной реализации – при разработке должна оцениваться и учитываться примерная конечная стоимость продукта. Если стоимость большая, то при использовании алгоритма, входящего в международные стандарты, производители будут предлагать собственные, более конкурентоспособные решения и алгоритмы. На практике данное требование

означает, что при реализации алгоритма должно быть использовано наименьшее число микросхем.

Рассмотренные выше требования к алгоритмам являются противоречивыми. Высокая степень сжатия подразумевает архивацию каждого следующего кадра, используя предыдущий. Но с другой стороны требования на аудиовизуальную синхронизацию и произвольный доступ к любому кадру за ограниченное время дают возможность вытянуть все кадры в цепочку. Несмотря на это существует некий компромисс. Реализация сбалансированной, учитывающей все противоречивые требования системы можно достичь за счет настройки компрессора при сжатии видеофайла.

Методы сжатия видеоданных можно разделить на три типа:

- Jpeg-подобные;
- wavelet-подобные;
- фрактальные.

Фрактальные методы сжатия не получили широкого распространения в связи со сложностью алгоритма, поэтому рассматриваться они не будут.

Процесс сжатия видео в MPEG можно описать двумя этапами: во-первых, уменьшение избыточности видеoinформации во временном измерении, оно основано на том, что ближайшие кадры мало отличаются, и, во-вторых, сжатие отдельных изображений.

Существует 4 типа кадров, которые удовлетворяют всем требованиям:

- I-кадры - кадры, сжатые независимо;
- P-кадры- при сжатии используются ссылки на изображения;
- B-кадры- сжатие с использованием ссылки на два отдельных изображения;
- DC-кадры – используются только для быстрого поиска, данные кадры, сжатые с большой потерей качества и независимо.

I-кадры дают возможность произвольного доступа к любому кадру. Они являются входными точками в поток видео данных для декодера. P-кадры используют при архивации ссылку на один I- или P-кадр, при этом повышается степень сжатия. B-кадры, используя ссылки на два кадра (следующий и предыдущий), обеспечивают самую высокую степень компрессии.

Одним из основных понятий при сжатии нескольких изображений является понятие макроблока. При сжатии кадр из цветового пространства RGB переводится в цветовое пространство YUV. Каждая из плоскостей сжимаемого изображения (Y, U, V) разделяется на блоки 8x8, с которыми работает ДКП. Причем плоскости U и V, соответствующие компоненте цветности, берутся с разрешением в 2 раза меньшим (по вертикали и горизонтали), чем исходное изображение. Таким образом, мы сразу получаем сжатие в 2 раза, пользуясь тем, что глаз человека хуже различает цвет отдельной точки изображения, чем ее яркость (подробнее об этих преобразованиях смотрите в описании алгоритма JPEG). Блоки 8x8 группируются в макроблоки.

Макроблок – это группа, которая состоит из четырех соседних блоков в плоскости компоненты яркости Y (матрица из пикселей элементов 16x16) и 2 соответствующих им по расположению блока из плоскостей цветности U и V. Таким образом, кадр разбивается на независимые единицы. Данные единицы несут полную информацию только о части изображения. При этом размер изображения должен быть кратен 16.

Отдельные макроблоки сжимаются независимо. Это означает, что в B-кадрах можно сжать конкретный макроблок как I-блок, P-блок со ссылкой на предыдущий кадр, P-блок со ссылкой на последующий кадр и B-блок.

Алгоритм сжатия кадров в MPEG сравним с уже существующим алгоритмом для изображений – JPEG. Если говорить коротко, то сам алгоритм сжатия представляет собой конвейер преобразований. Это дискретное косинусное преобразование исходной матрицы 8x8, квантование матрицы и вытягивание

ее в вектор (зигзаг – сканирование), сжатие вектора групповым кодированием и, наконец, сжатие по алгоритму Хаффмана.

Общую схему алгоритма можно представить следующим образом:

- 1) подготовка макроблоков – на начальном этапе каждому макроблоку присваивается метод сжатия;
- 2) перевод макроблока в цветовое пространство YUV. Получение нужного количества матриц 8x8;
- 3) для P- и B- блоков вычисляется разность с определенным макроблоком в опорном кадре;
- 4) дискретное кодовое преобразование (ДКП);
- 5) квантование;
- 6) зигзаг-сканирование;
- 7) групповое кодирование;
- 8) кодирование Хаффмана.

В процессе декодирования перечисленные шаги повторяются в обратном порядке.

Данный алгоритм основан на сжатии отдельных кадров с использованием JPEG-сжатия [12]. После выполнения сжатия из последовательности изображений формируется видеопоток.

Из-за того, что человеческий глаз обладает большей чувствительностью к изменению яркости, чем к изменению цвета, кодирование изображений Jpeg подобными алгоритмами происходит представлением картинки не в RGB-формате, а в формате Y<sub>C</sub>C<sub>b</sub>. При этом компоненты хроматического красного и хроматического синего усредняются. Это позволяет уменьшить объем данных более, чем в 1.5 раза.

Motion-JPEG является наиболее простым алгоритмом сжатия видеоданных. В нем каждый кадр сжимается независимо алгоритмом JPEG. Этот прием дает высокую скорость доступа к произвольным кадрам как в прямом, так и в обратном порядке следования. Соответственно легко реализуются плавные

«перемотки» в обоих направлениях, аудиовизуальная синхронизация и, что самое главное, редактирование.

Типичные операции JPEG сейчас поддерживаются на аппаратном уровне большинством видеокарт, и данный формат позволяет легко оперировать большими объемами данных при монтаже фильмов. Независимое сжатие отдельных кадров позволяет накладывать различные эффекты, не опасаясь, что взаимное влияние соседних кадров внесет дополнительные искажения в фильм. Среди характеристик M-JPEG можно отметить поток, разрешение и сжатие. Поток и разрешение являются произвольными, а сжатие доступно от 5 до 10 раз.

Среди плюсов алгоритма можно отметить быстрый доступ к отдельным кадрам видеоданных, простое редактирование потока, доступная и недорогая аппаратная реализация. Среди минусов – низкая степень сжатия.

При кодировании изображение разбивается на блоки размером  $8 \times 8$  пикселей, и дальнейшие операции выполняются над каждым из этих блоков. Кодирование производится в три фазы:

- 1) быстрое преобразование Фурье;
- 2) квантование;
- 3) сжатие.

Преобразование Фурье осуществляет перевод сигналов из пространственной области в частотную. Для блока размером  $8 \times 8$  пикселей оно задается выражением 2.1:

$$l(u, v) = \frac{a(u)a(v)}{4} \sum_{x=0}^7 \sum_{y=0}^7 i(x, y) \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right) \quad (2.1)$$

Квантование осуществляется путем деления амплитуд спектральных составляющих, полученных после преобразования Фурье, на заранее заданные

числа, взятые из таблицы квантования. Сжатие осуществляется в три стадии. Сначала производится зигзаг-сканирование, как показано на рисунке 2.5.

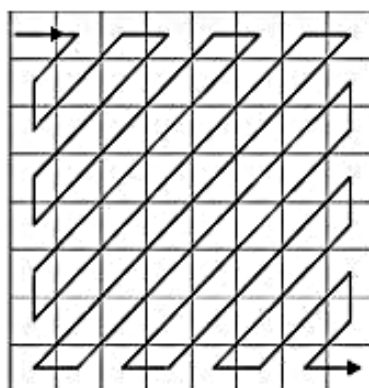


Рисунок 2.5 – Схема зигзаг-сканирования

Эта операция по упорядочению коэффициентов Фурье необходима для того, чтобы получить более длинные последовательности нулей, поскольку такие последовательности лучше сжимаются. Затем происходит RLE-сжатие, при котором каждый коэффициент кодируется двумя числами: первое число показывает, сколько нулевых элементов предшествует данному коэффициенту, а второе значение этого коэффициента. Затем производится либо сжатие по Хаффману, либо арифметическое кодирование. Ниже при сравнении данного метода кодирования с другими алгоритмами применено сжатие по Хаффману с использованием статических таблиц.

Стандарт Mpeg4 описывает методы кодирования различной мультимедиа информации, такой как аудио и видеопотоки, статические двумерные изображения и различные интерактивные типы информации. Поскольку целью данной работы является исследование различных методов сжатия видеопотоков, далее рассматривается только та часть стандарта, которая непосредственно касается сжатия видеопотоков.

В стандарте Mpeg4 так же, как и в стандарте JPEG, видеоданные представляются в формате  $Y_CrCb$ . Кодирование видеопотоков производится с использованием объектных видео плоскостей (VOP-плоскостей). При сжатии видеопотока каждое изображение можно представить в виде суперпозиции VOP-плоскостей, которые в дальнейшем кодируются независимо друг от друга. VOP-плоскость содержит определенную часть кадра, хранящую некоторую визуально различимую область исходного изображения.

Поскольку на данный момент не существует эффективных методов разделения изображений на отдельные объекты, чаще всего используется всего одна видео-плоскость, целиком содержащая исходное изображение. При сжатии видеоплоскость разделяется на последовательность кадров. В стандарте Mpeg4 используются три типа кадров: I, P и B.

I-кадры кодируются целиком, без использования информации о других кадрах. P-кадры кодируются с использованием информации о предыдущих I- и P-кадрах. B-кадры используют информацию как о предыдущих, так и о последующих I- и P-кадрах.

Теперь рассмотрим непосредственно кодирование кадров (рисунок 2.6). Кадры разбиваются на макроблоки.

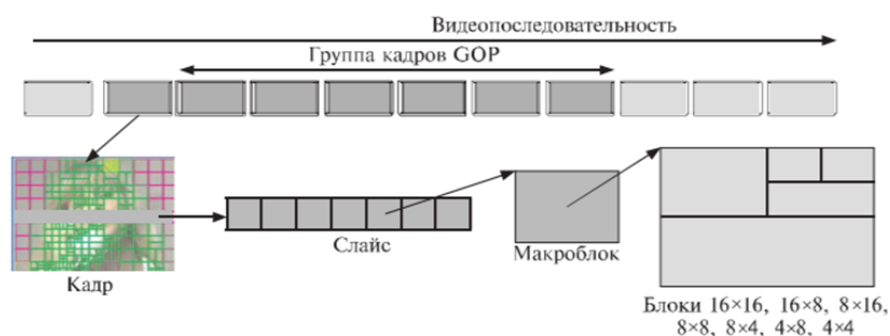


Рисунок 2.6 – Кодирование кадров

Для Y-компоненты макроблоки имеют размер  $16 \times 16$  пикселей, а для  $C_r$  и  $C_b$  компонент макроблоков равен  $8 \times 8$ . Макроблоки бывают двух типов: intra-

макроблоки и inter-макроблоки. Макроблоки разных типов кодируются по-разному. I-кадр состоит только из intra-макроблоков, а P-кадр содержит и inter- и intra-макроблоки.

Рассмотрим кодирование intra-макроблоков для  $C_r$ - и  $C_b$ -компонент и подблоков Y-компоненты. Сначала производится преобразование Фурье точно такое же, как описано выше в стандарте JPEG. После этого производится квантование коэффициентов Фурье. Старший из коэффициентов Фурье отражает среднее значение яркости в данном блоке, поэтому его называют DC-коэффициентом. Поскольку чаще всего яркость соседних блоков одинакова, кодирование DC-коэффициента целиком является избыточным, достаточно закодировать разность между текущим и одним из предыдущих блоков. Подобное предсказание производится и для старших AC-коэффициентов. Если при предсказании количество информации не уменьшается, то блок кодируется без учета предсказания. Если AC-предсказание осуществляется, то – в зависимости от того, по отношению к какому блоку оно производится, – выбирается один из трех возможных методов зигзаг-сканирования. После этого выполняется RLE-сжатие и сжатие по Хаффману так же, как и в стандарте JPEG.

Для inter-макроблоков сначала производится операция компенсации движения, и только после этого они кодируются. Для каждого макроблока используется один вектор скорости; при этом предсказание осуществляется с точностью до 0,125–0,5 пиксела. Для нахождения вектора скорости производится поиск наиболее похожего места на предыдущих I- и P- кадрах. Поиск векторов движения производится только для Y-компоненты, для  $C_r$  и  $C_b$ -компонент вектор движения получается путем деления пополам вектора для компоненты Y. После нахождения вектора движения из кодируемого макроблока вычитается наиболее похожий блок, и в дальнейшем кодируются разница между блоками, позиция фрейма, по отношению к которому закодирован данный макроблок, и вектор движения. Если кодирование разницы занимает больше бит, чем кодирование исходного блока, то блок маркируется как intra-



блок и после этого производится его кодирование по соответствующему алгоритму. В дальнейшем inter-блоки кодируются точно так же, как и intra-блоки, за исключением того, что для них не производятся AC/DC-предсказания.

Стандарт MPEG-4. Он кардинально отличается от принимаемых ранее стандартов. Рассмотрим наиболее интересные и полезные нововведения.

Расчет трехмерных сцен и работа с синтетическими объектами. Составной частью декодера MPEG-4 является блок визуализации трехмерных объектов. В MPEG-4 каждый накладываемый объект рассчитывается отдельно, а накладывается только потом. Также существует возможность использования видеопотока в качестве текстуры. Она накладывается на поверхности рассчитываемых объектов. Работа такого плана с трехмерными объектами значительно увеличивает степень сжатия при сохранении хорошего качества изображения. Размер синтетически созданного описания будет в несколько раз меньше, чем размер аналогичных фильмов, которые были сжаты как поток кадров. Отдельно в стандарте предусмотрена работа со «спрайтами» – статическими изображениями, накладываемыми на кадр. При этом размер спрайта варьируется от самого маленького до превышающего размер кадра в несколько раз. Это дает определенную гибкость при создании MPEG-4- фильмов и позволяет сильно уменьшать объем информации для кодирования. [13]

Объектно-ориентированная работа с потоком данных. Теперь работа с потоком данных становится объектно-ориентированной. Данные – это и живое видео, и звуковые данные, а также синтетические объекты и тому подобное. Из них можно создавать сцены, которыми можно управлять. Для обычных пользователей мало что изменится, но для разработчиков среда объектов кардинальное упрощает работу со сложными структурами.

Активная зрительская позиция. BIFS позволяет задавать реакцию объектов сцены на конкретные действия пользователя. Среди действий можно выделить удаление, добавление объектов, ввод команд с клавиатуры. Событийная модель заимствована из языка моделирования виртуальной реальности

VRML. Данная технология создает огромное пространство для создания программ для обучения и развлечений. В стандарте предусмотрена обработка команд на стороне сервера. Программа-просмотрщик отправляет данные на сервер, затем получает ответ в виде оценки.

Синтезатор лиц и фигур. В стандарте присутствует интерфейс к модулю синтеза лиц и фигур. Например, в файл записываются ключевая информация о профиле и текстуре лица, а при записи фильма сохраняются коэффициенты изменения формы. Для новостных передач данная функция позволяет в десятки раз уменьшить размер файла без ухудшения качества.

Улучшенные алгоритмы сжатия видео. В стандарте предусмотрены блоки, отвечающие за потоки 4,8–65 Кбит/с с прогрессивной и чересстрочной разверткой. При передаче через слабозащищенные каналы связи предусмотрены методы кодирования с устойчивостью к помехам. За счет незначительного увеличения размера передаваемых данных уменьшается вероятность потери части. При передаче видео с одновременным просмотром возможно огрубление изображения в случае, если из-за ограниченной пропускной способности канала связи декодер не успевает получить и обработать всю видео информацию. Заложенные в стандарте три уровня детализации изображения позволяют легко адаптировать компрессионный алгоритм для сетевой трансляции видео.

Поддержка профилей на уровне стандарта. Понятно, что реализация всех возможностей стандарта превращает декодер в весьма сложную и большую конструкцию. При этом далеко не для всех приложений необходимы какие-то сложные специфические функции (например, синтез речи). Создатели стандарта поступили просто: они оговорили наборы профилей, каждый из которых включает в себя набор обязательных функций. Если в фильме записано, что ему для проигрывания необходим такой-то профиль и декодер этот профиль поддерживает, то стандарт гарантирует, что фильм будет проигран правильно.

Выше кратко перечислены некоторые отличия MPEG-4 от предыдущих стандартов. Надо отметить, что на момент создания стандарта острой потребности в описанных выше вещах еще не было. Иначе говоря, мы имеем дело с хорошо продуманной работой по формированию стандарта, которая была закончена к тому времени, когда в нем возникла первая необходимость.

Создателями MPEG-4 учтен опыт предшественников (в частности, VRML), когда слишком раннее появление стандарта и отсутствие в нем механизма профилей серьезно подорвало его массовое применение. Будем надеяться, что массовому применению MPEG-4 такие проблемы не грозят.

Характеристики MPEG-4: поток, разрешение: 0,0048–20 Мбит/с, поддерживаются все основные стандарты видеопотоков.

Плюсы: поддержка достаточно прогрессивных звуковых стандартов, высокая степень универсальности, поддержка новых технологий (различные виды синтеза звука и изображения).

Минусы: высокая сложность реализации.

Стандарт H.261 специфицирует кодирование и декодирование видеопотока для передачи по каналу  $r \times 64$  Кбит, где  $r=1...30$ . В качестве канала может выступать, например, несколько телефонных линий.

Входной формат изображения – разрешения CIF или QCIF, частота кадров от 30 fps и ниже. Используется уменьшение разрешения в 2 раза для компонент цветности.

В выходной поток записываются два типа кадров: INTRA – сжатые независимо (соответствуют I-кадрам) и INTER – сжатые со ссылкой на предыдущий кадр (соответствуют P-кадрам). В передаваемом кадре не обязательно присутствуют все макроблоки изображения; если блок изменился незначительно передавать его обычно нет смысла. Сжатие в INTRA-кадрах осуществляется по схеме сжатия отдельного изображения. В INTER-кадрах производится аналогичное сжатие разности каждого передаваемого макроблока с

«наиболее похожим» макроблоком из предыдущего кадра, (компенсация движения). Для сглаживания артефактов ДКП предусмотрена возможность применения размытия внутри каждого блока 8x8 пикселей. Стандарт требует, чтобы INTRA-кадры встречались в потоке не реже чем через каждые 132 INTER-кадра (чтобы не накапливалась погрешность кодирования и была возможность восстановиться в случае ошибки в потоке).

Степень сжатия зависит в основном от метода нахождения «похожих» макроблоков в предыдущем кадре, алгоритма решения, передавать ли конкретный макроблок, выбора способа кодирования каждого макроблока (INTER/INTRA) и выбора коэффициентов квантования результатов ДКП. Ни один из перечисленных вопросы стандартом не регламентируются, оставляя свободу для построения собственных оптимальных алгоритмов.

Характеристики H.261: поток, разрешение: рх64 Кбит, р=1..30, CIF или QCIF. Среди плюсов выделяется простота реализации. Наиболее важным минусом стандарта является достаточно низкая степень сжатия.

Данный стандарт похож на H.261, но с дополнениями и усложнениями функций и характеристик. Он содержит «базовый» стандарт кодирования, который почти не отличается от алгоритма сжатия H.261, дополненного множеством опций и расширений. Ниже будут перечислены его основные отличия.

Вместо кодов Хаффмана используется арифметическое кодирование. Это дает повышение степени сжатия от пяти до десяти процентов.

Поддержка большого числа форматов входных видеоданных. Основное отличие от более универсальных форматов заключается в адаптации для нескольких фиксированных разрешений, что позволяет делать менее универсальные, но более быстрые процедуры обработки кадров. Построенный таким образом декодер работает несколько быстрее.

Возможность задания векторов смещения, указывающих за границы изображения. При этом граничные пиксели используются для предсказания

пикселей вне изображения. Данный прием усложняет алгоритм декодирования, но позволяет значительно улучшить изображение при резкой смене плана сцены.

Появление В-кадров, которые позволяют увеличить степень сжатия, за счет усложнения и увеличения времени работы декодера.

Компенсация движения с субпиксельной точностью. Возможность сдвинуть блок на полпиксела также увеличивает степень сжатия, но увеличивает время работы декодера.

Возможность задания вектора смещения для каждого блока 8x8 в макроблоке, что в ряде случаев существенно увеличивает сжатие и снижает блочность изображения.

Особый режим сжатия INTRA-макроблоков со ссылкой на соседние макроблоки в обрабатываемом кадре, особый режим квантования и специальная таблица Хаффмана для улучшения сжатия I-кадров в ряде случаев.

Изменение разрешения и деформирование базового кадра, использующегося в качестве базового при сжатии.

Улучшено сглаживание границ блоков изображения, прошедшего декодирование. Тем самым уменьшается эффект «блочности». При резком движении в кадре алгоритм сжатия вынужден повысить степень квантования блоков после ДКП. Это нужно для уместения в определенный битовый поток для передачи. При этом в кадре изображения возникают блоки размером 8x8. «Сращивание» границ крайних пикселей блоков сближают по яркости так, чтобы уменьшить заметную разницу. Это позволяет очень часто значительно улучшить визуальное качество видео.

Различные режимы квантования и кодирования по Хаффману.

Характеристики H.263: поток, разрешение: 0,04-20 Мбит/с, sub-QCIF, QCIF, CIF, 4CIF, 16CIF и отдельно настраиваемые разрешения.

Плюсы: алгоритм H.263, так же как H.261, допускает быструю аппаратную реализацию, однако при этом позволяет добиться большей степени сжатия при том же качестве. Поддерживает сжатие звука.

Минусы: по количеству заложенных идей находится между MPEG-2 и MPEG-4.

В основе Wavelet-подобных алгоритмов сжатия лежит теория Wavelet-преобразования. При Wavelet преобразовании производится разложение сигнала по базисным функциям. В отличие от преобразования Фурье в качестве базовых функций используются не косинусы, а вейвлеты. На рисунке 2.7 представлены четыре семейства Wavelet- функций. [14]

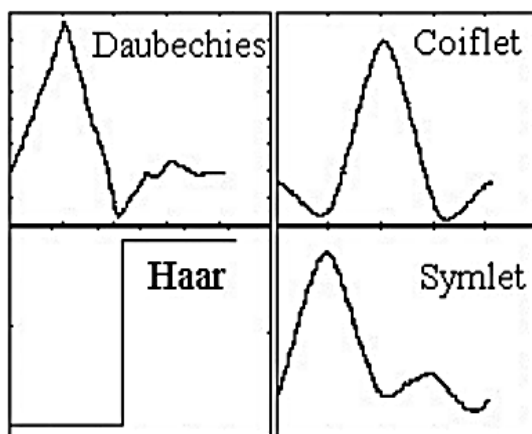


Рисунок 2.7 – Четыре наиболее известных семейства вейвлет-функций

Для сжатия изображений используются два вейвлет преобразования с разными базисными функциями. При этом базисные функции выбирают таким образом, чтобы одно преобразование осуществляло фильтрацию низких частот, а второе – высоких. Низкочастотную составляющую сигнала оставшуюся после фильтрации будем далее обозначать буквой – L, а высокочастотную – H. Поскольку обрабатываемая картинка двумерна, производится двумерное вейвлет преобразование, при этом сначала осуществляется фильтрация по строкам, а затем по столбцам, либо наоборот.

Рассмотрим операцию сжатия. Изображение представляется в виде компонент  $YCbCr$ , каждая из компонент обрабатывается отдельно. Кодирование каждой компоненты происходит в 4 стадии:

- 1) дискретное двумерное вейвлет-преобразование;
- 2) квантование;
- 3) для низкочастотных компонент – кодирование с использованием дискретной импульсно-кодовой модуляции (каждый элемент кодируется с использованием трех соседних), а для высокочастотных – сканирование нуль деревьев;
- 4) арифметическое кодирование.

Таким образом, подводя итоги, были рассмотрены различные алгоритмы сжатия, а в качестве основного метода сжатия был выбран PNG формат. Он имеет наиболее высокую степень сжатия, что позволяет производить передачу изображения с удаленного ПК с наименьшими нагрузками на сеть.

## 3 Реализация приложения для удаленного администрирования ПК

### 3.1 Серверная часть

После запуска приложения-сервера появляется следующее окно, изображенное на рисунке 3.1.

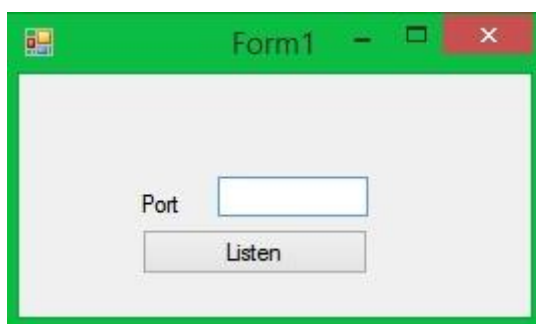


Рисунок 3.1 – Стартовое окно приложения-сервера

После заполнения поля Port и нажатия кнопки Listen появляется окно, на котором будет отображаться изображение рабочего стола удаленного ПК.

Листинг программного кода сервера представлен в Приложение А.

### 3.2 Клиентская часть

После запуска приложения-клиента появляется окно, изображенное на рисунке 3.2.

В поле IP необходимо ввести IP адрес, а в поле Port – порт сервера. После нажатия на кнопку Connect произойдет подключение. После нажатия на кнопку Share Screen начнется передача изображения рабочего стола от клиента к серверу.

Листинг программного кода приложения-клиента представлен в Приложение Б.



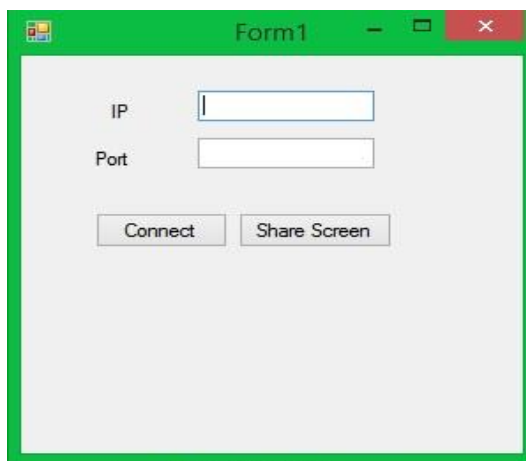


Рисунок 3.2 – Стартовое окно приложения-клиента

### 3.3 Анализ и сравнение трафика

Для анализа трафика приложения для удаленного управления рабочим столом используется программа CommView[15]. Разработчик – TamoSoft. Используется сетевыми администраторами для быстрого и качественного решения задач администрирования, связанных с нахождением уязвимостей, отслеживанием трафика, поиском и устранением вторжения в сети, а также связанных с утечкой информации.

Первым этапом проанализируем трафик, проходящий через различные продукты для RDC: TeamViewer, Radmin, RMS, а также разработанное в данной работе приложение.

За среднее время одного сеанса связи примем 10 минут. Пропускная способность сетевого канала 80 Мбит/с.

При помощи функционала приложения CommView измерим объем трафика в среднем, передающийся через все приложения по очереди.

Как видно на рисунке 3.3, среднее количество байт в секунду за 10 минут работы равняется 77644, что составляет 0,075 Мб/с. Загрузка сети за все время работы не превышает 0,1%.

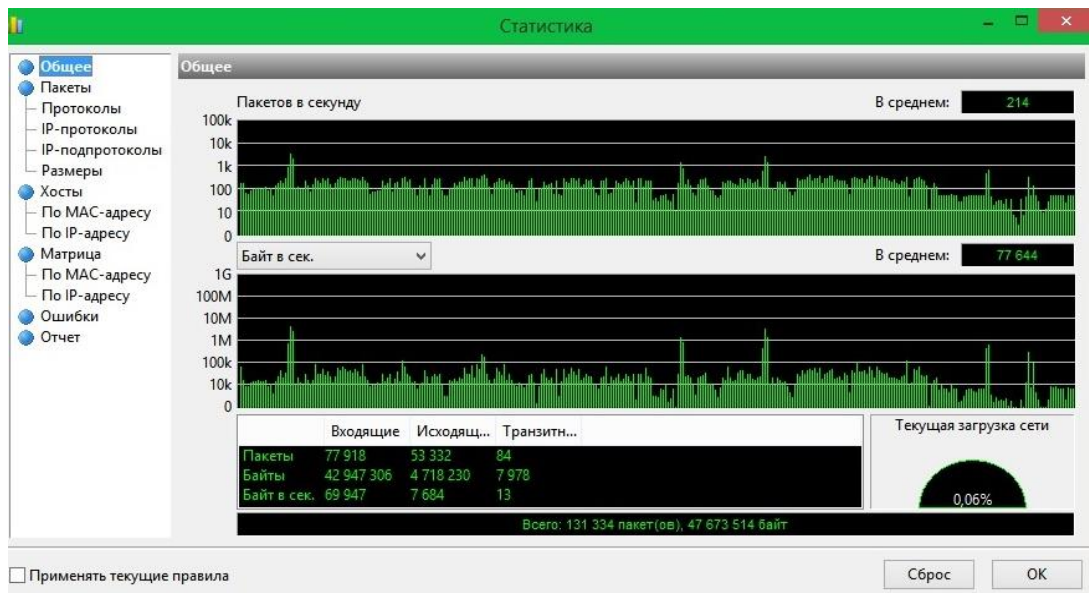


Рисунок 3.3 – Анализ TeamViewer

Из рисунка 3.4 видно, что Radmin, в отличие от TeamViewer, занимает большую часть сети. Если говорить о цифрах, то это 192275 байт/сек против 77644 байт/сек, что 2,5 раза больше.

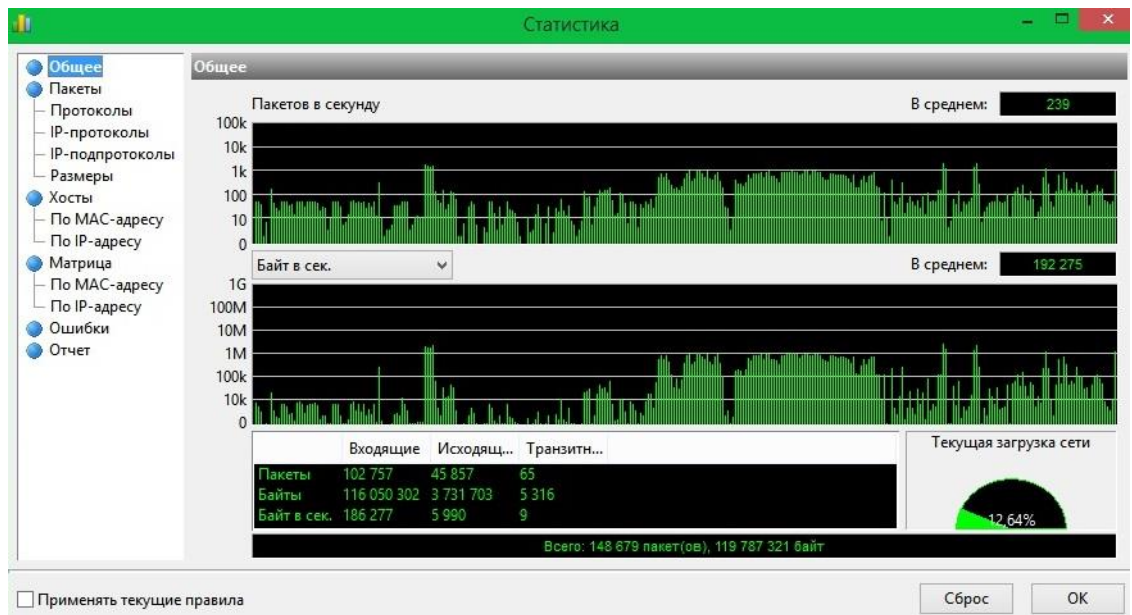


Рисунок 3.4 – Анализ Radmin

Рисунок 3.5 показывает, что среднее количество передаваемых байт в секунду составляет 131254, что в 1,7 раза больше передаваемых через TeamViewer.

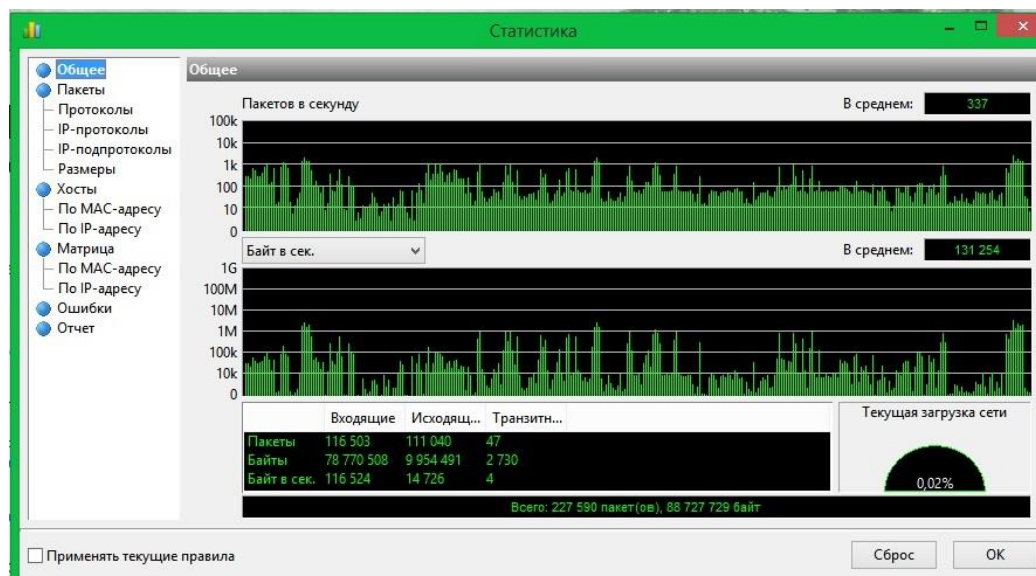


Рисунок 3.5 – Анализ RMS

И, наконец, проведем анализ самостоятельно разработанного приложения (Рисунок 3.6). Как видно по представленной статистике приложение имеет число в графе байт в секунду 125866, что составляет 0,96 Мбит/сек.

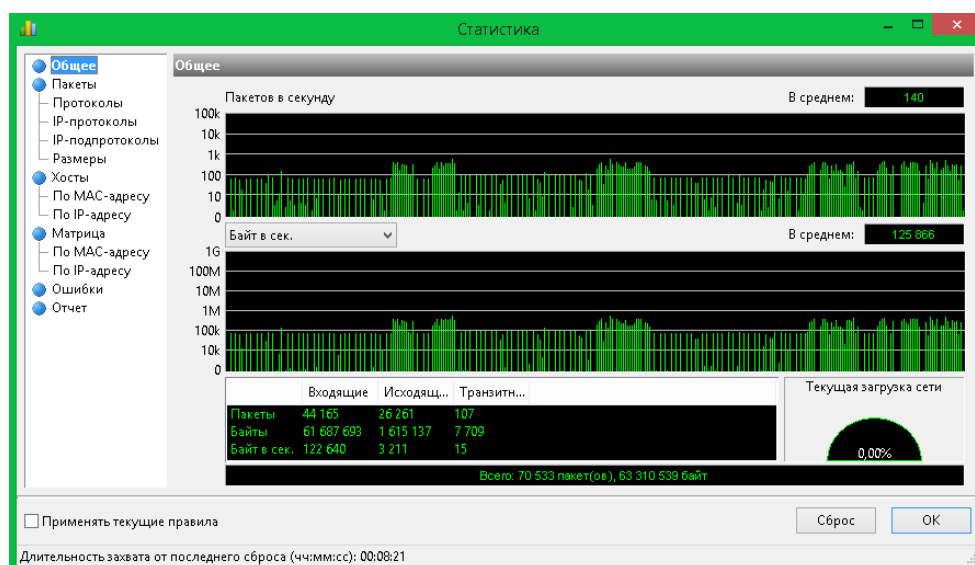


Рисунок 3.6 – Анализ приложения для удаленного доступа

Далее составим сравнительную таблицу (таблица 3.1), которая наглядно покажет объем потребляемой пропускной способности канала связи между удаленными ПК.

Таблица 3.1 – Сравнение трафика различных RDC - приложений

Название приложения	Объем передаваемого трафика, байт/сек	Процент от сетевого канала (80 Мбит/сек), %
TeamViewer	77644	0,73
Radmin	192275	1,83
RMS	131254	1,25
Приложение RDC	125866	1,2

На рисунке 3.7 представлен скриншот работы программы, разработанной в данной работе.

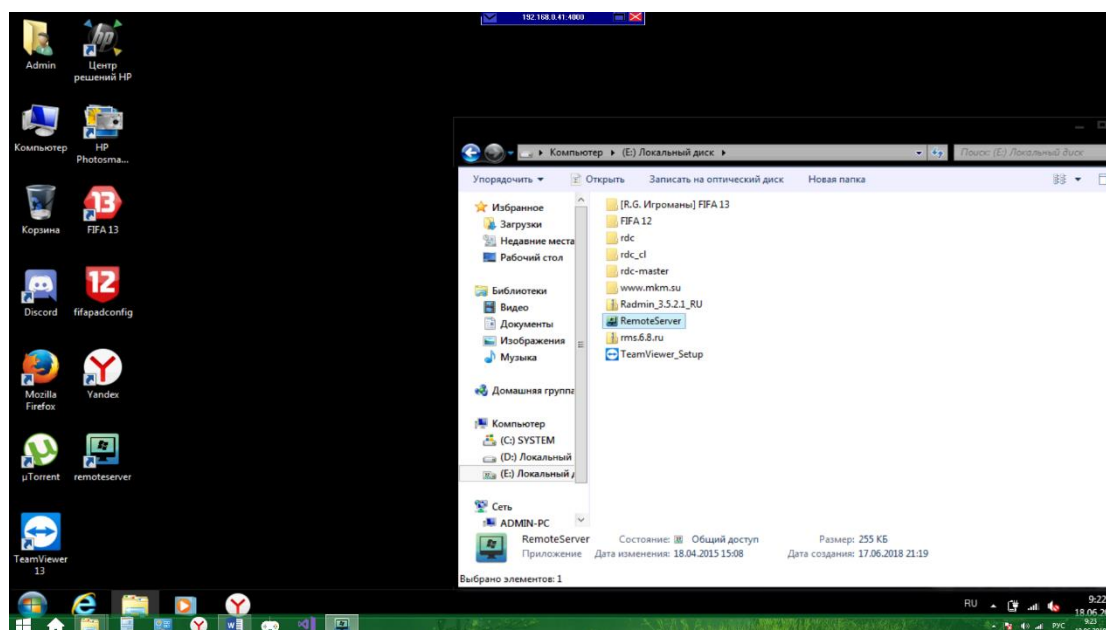


Рисунок 3.7 – Скриншот рабочего стола удаленного ПК

### 3.4 Выбор алгоритма шифрования

В качестве метода шифрования канала связи между удаленными ПК было выбрано сквозное шифрование. Данный способ передачи означает, что доступ к данным имеют только лишь пользователи конечных устройств. Таким образом сквозное шифрование не позволяет третьим лицам завладеть криптографическими ключами.

Существует несколько алгоритмов обмена ключами: симметричный (AES, DES, RC5 и другие), асимметричный (RSA, DSA, Диффи-Хелмана и прочие). Оконечное шифрование предполагает то, что ключи шифрования имеются только взаимодействующие между собой стороны.

Сквозное шифрование, которое используется в известных мессенджерах Telegram и WhatsApp, гарантирует доступ к исходным данным только отправителю и получателю. Таким образом информация пользователей становится недоступной даже серверам, через которые происходит передача.

Основными угрозами обхода шифрования являются: атака «человек посередине», безопасность конечных точек и бэкдоры.

Атака «человек посередине» подразумевает захват канала связи между получателем и отправителем, а затем выдача себя за получателя сообщения путем подмена открытого ключа. Для маскировки третья сторона может зашифровать сообщение и передать отправителю. Для предотвращения атаки принято использовать отпечаток открытого ключа, которые сравниваются на обеих сторонах перед началом общения.

Атака вида «безопасность конечных точек» означает, что под угрозой становятся конечные устройства, которые взламываются, что позволяет злоумышленникам захватить ключ шифрования или дешифрованное сообщение. В данном случае основным и самым эффективным методом повышения надежности является физическая изоляция устройств.

И, наконец, бэкдор, которые некоторые компании могут внедрять в программное обеспечение для нарушения или обхода шифрования. Примером является Skype, который передавал данные в Агентство Национальной безопасности США.

### 3.5 Сравнение методов сжатия изображения

В работе для обеспечения наименьшей загрузки сети было проведено сравнение основных методов сжатия изображения: JPEG, TIFF и PNG.

На рисунке 3.8 приведена статистика работы приложения с методом сжатия изображений JPEG.

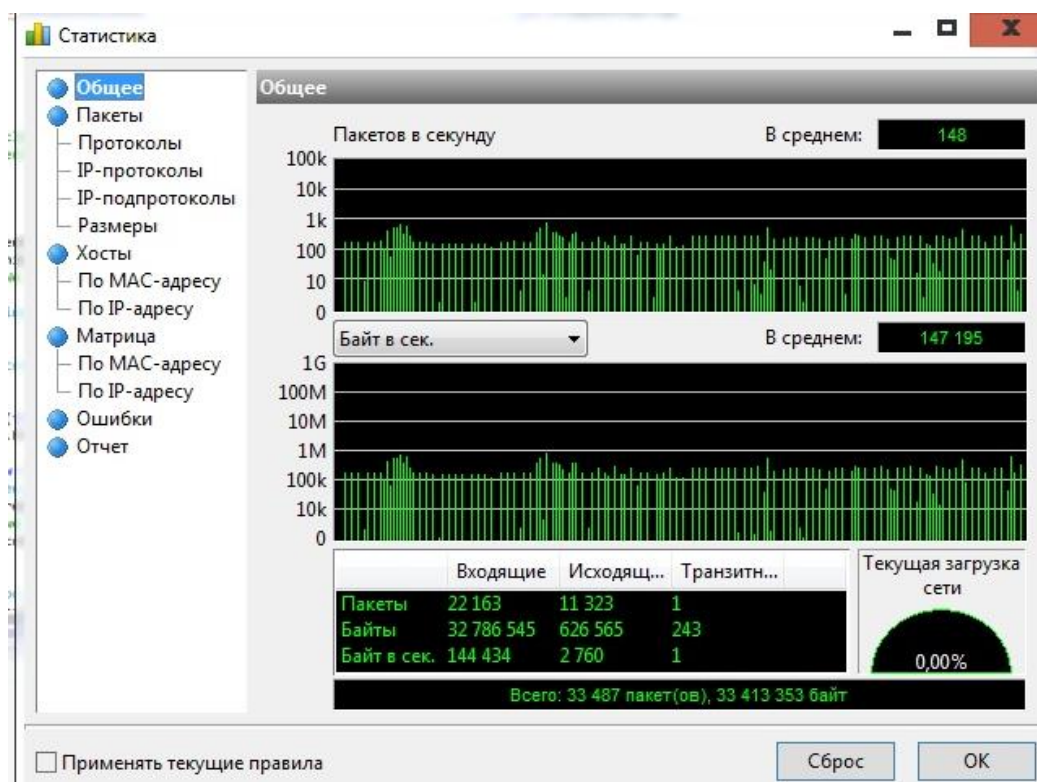


Рисунок 3.8 – Анализ JPEG

На рисунке 3.9 изображен анализ сжатия TIFF.

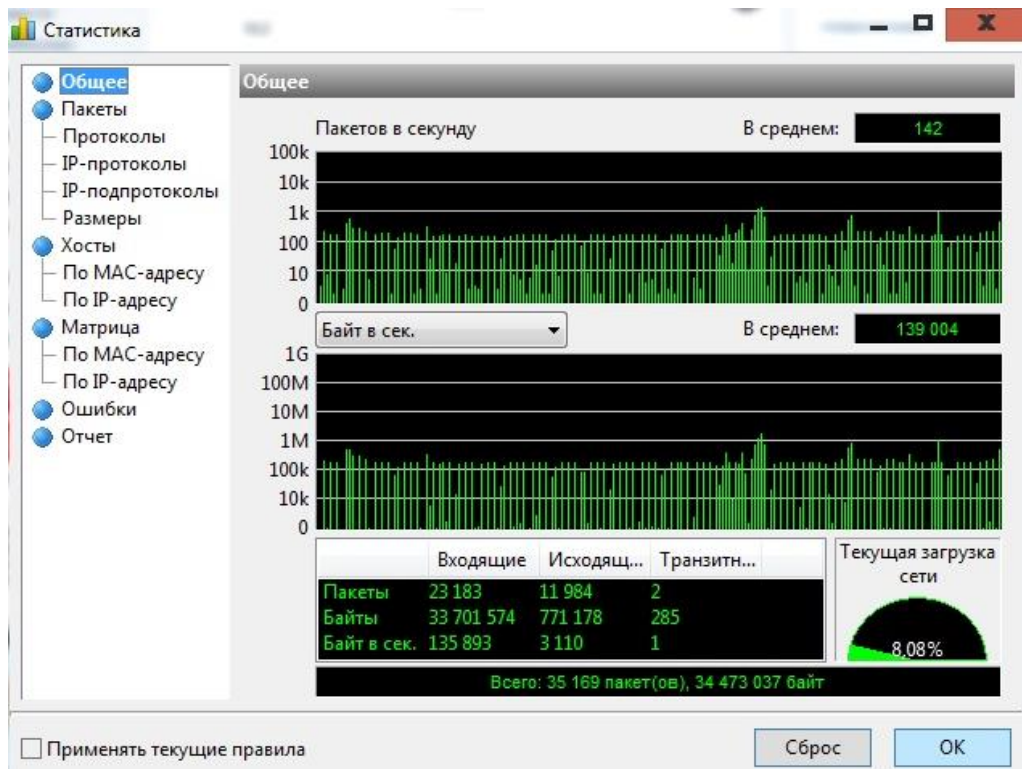


Рисунок 3.9 – Анализ TIFF

Наконец, на рисунке 3.10 показана статистика приложения за время работы с методом сжатия PNG.

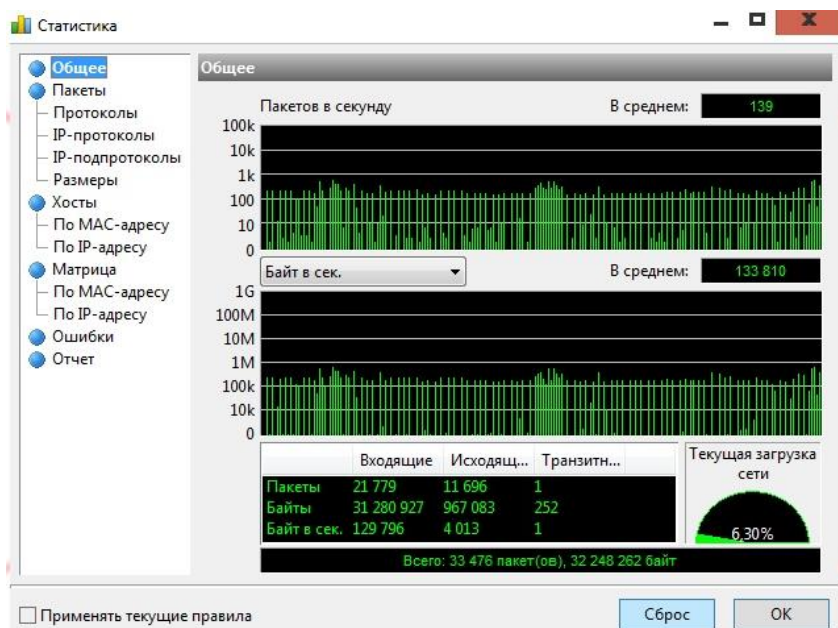


Рисунок 3.10 – Анализ PNG

Как видно из рисунков 3.8 – 3.10, наилучшим форматом для сжатия изображения является PNG. Он обеспечивает наибольшую степень сжатия передаваемых изображений, что благоприятно влияет на загрузку сетевого канала.

					БР-02069964-11.03.02-05-18	Лист
Лит	Изм.	№ докум.	Подп.	Дата		56



## ЗАКЛЮЧЕНИЕ

По результатам проделанной работы можно подвести следующие итоги:

- был произведен обзор существующих приложений для удаленного доступа и администрирования ПК, выявлены их преимущества и недостатки;
- сформированы необходимые требования для оптимальной работы приложения;
- изучена клиент-серверная архитектура и принцип взаимодействия хоста и клиента;
- изучены способы сжатия графических данных;
- выбран оптимальный алгоритм сжатия изображений для оптимизации нагрузки на сеть;
- реализовано приложение для удаленного доступа к ПК;
- был произведен анализ трафика приложений;
- выбран и реализован метод шифрования передаваемых данных между клиентом и сервером.

Таким образом, итогом проделанной работы является приложение, позволяющее в режиме реального времени производить администрирование удаленного ПК.

					Лист
Лит	Изм.	№ докум.	Подп.	Дата	БР-02069964-11.03.02-05-18

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Удаленный доступ к компьютеру. [Электронный ресурс]. – Режим доступа: <http://www.teamviewer.com/ru/>.
- 2 Руководство пользователя TeamViewer для удаленного управления. [Электронный ресурс]. – Режим доступа: <https://dl.tvcdn.de/docs/ru/v12/TeamViewer12-Manual-Remote-Control-ru.pdf>.
- 3 Radmin: удаленное управление ПК. [Электронный ресурс]. – Режим доступа: <http://www.radmin.ru/>.
- 4 Хилл Бр., Полный справочник по Cisco. [Текст]. Пер. с англ./ Бр. Хилл. – М.: Издательский дом «Вильямс», 2004. – 1078 с.
- 5 Удаленное администрирование RMS. [Электронный ресурс]. – Режим доступа: <https://rmansys.ru/>.
- 6 Удаленный доступ. [Электронный ресурс]. – Режим доступа: <https://rmansys.ru/remote-access/>.
- 7 Улучшенный стандарт шифрования (AES). [Текст]. Пер. с англ. – Национальный институт стандартов и технологий, 2001. – 38 с.
- 8 Нортрап Т., Основы разработки приложений на платформе Microsoft .NET Framework. [Текст]. Пер. с англ. / Нортрап Т., Вилдермьюс Ш., Райан Б. – М.: Издательство «Русская Редакция», 2007. – 864 с.
- 9 Сетевое программирование в C# и .NET. [Электронный ресурс]. – Режим доступа: <https://metanit.com/sharp/net/>.
- 10 Ватолин Д.С. Алгоритмы сжатия изображений. Методическое пособие. [Текст] / Д.С. Ватолин. - Издательский отдел факультета Вычислительной Математики и Кибернетики МГУ им. М.В.Ломоносова, 1999. – 76 с.
- 11 Алгоритмы сжатий изображений. [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/116697/>.
- 12 Wallace, G. "The JPEG algorithm for image compression standard" / G. Wallace. – Vol 34 No 4 Communications of the ACM, 1991. – pp 31-44.

13 Niedemayer, M., "DIVX3 / MS-MPEG4v1-v3 / WMV7-8" / M. Niedemayer. – GNU Free Documentation, 2003. – 98 p.

14 Ватолин Д. С. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео / Д.С. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин. – М.: Диалог-МИФИ, 2003. – с.67-70.

15 Официальный сайт компании TamoSoft [Электронный ресурс]. – Режим доступа: <https://www.tamos.ru>

					БР-02069964-11.03.02-05-18	Лист
Лит	Изм.	№ докум.	Подп.	Дата		59

**ПРИЛОЖЕНИЕ А**  
**(обязательное)**  
**СЕРВЕРНАЯ ЧАСТЬ**

```
using System;
using System.Drawing;
using System.IO;
using System.Net;
using System.Net.Sockets;
using System.Runtime.Serialization.Formatters.Binary;
using System.Security.Cryptography;
using System.Threading;
using System.Windows.Forms;

namespace rdc
{
    public partial class Form2 : Form
    {
        private readonly int port;
        private TcpClient client;
        private TcpListener server;
        private NetworkStream mainStream;

        private readonly Thread Listening;
        private readonly Thread GetImage;
        byte[] Key = { 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x10, 0x11, 0x12, 0x13, 0x14,
0x15, 0x16 };
        byte[] IV = { 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x10, 0x11, 0x12, 0x13, 0x14,
0x15, 0x16 };

        public Form2(int Port)
        {
            port = Port;
            client = new TcpClient();
            Listening = new Thread(StartListening);
            GetImage = new Thread(RecieveImage);
            InitializeComponent();
        }
        private void StartListening()
```

```

    {
        while (!client.Connected)
        {
            server.Start();
            client = server.AcceptTcpClient();
        }
        GetImage.Start();
    }

private void StopListening()
{
    server.Stop();
    client = null;
    if (Listening.IsAlive) Listening.Abort();
    if (GetImage.IsAlive) GetImage.Abort();
}

private void RecieveImage()
{
    BinaryFormatter binFormatter = new BinaryFormatter();
    while (client.Connected)
    {
        mainStream = client.GetStream();
        RijndaelManaged RMCrypto = new RijndaelManaged();
        CryptoStream CryptStream = new CryptoStream(mainStream, RMCrypto.CreateDecryptor(Key,
IV), CryptoStreamMode.Read);
        BinaryReader br = new BinaryReader(CryptStream);
        while (true)
        {
            var len = br.ReadInt32();
            var arr = br.ReadBytes(len);
            MemoryStream ms = new MemoryStream(arr);
            pictureBox1.Image = Image.FromStream(ms);
        }
    }
}

protected override void OnLoad(EventArgs e)
{
    base.OnLoad(e);
}

```

```
server = new TcpListener(IPAddress.Any, port);
Listening.Start();
}

protected override void OnFormClosing(FormClosingEventArgs e)
{
    base.OnFormClosing(e);
    StopListening();
}
}
}
```

					Лист
Лит	Изм.	№ докум.	Подп.	Дата	БР-02069964-11.03.02-05-18

**ПРИЛОЖЕНИЕ Б**  
**(обязательное)**  
**КЛИЕНТСКАЯ ЧАСТЬ**

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Imaging;
using System.IO;
using System.Linq;
using System.Net.Sockets;
using System.Runtime.Serialization.Formatters.Binary;
using System.Security.Cryptography;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace rdc_cl
{
    public partial class Form1 : Form
    {

        private readonly TcpClient client = new TcpClient();
        private NetworkStream mainStream;
        private CryptoStream CryptStream;
        private int portNumber;
        byte[] Key = { 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x10, 0x11, 0x12, 0x13, 0x14,
0x15, 0x16 };
        byte[] IV = { 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x10, 0x11, 0x12, 0x13, 0x14,
0x15, 0x16 };

        private Image GrabDesktop()
        {
            Rectangle bounds = Screen.PrimaryScreen.Bounds;
            Bitmap screenshot = new Bitmap(bounds.Width, bounds.Height, PixelFormat.Format32bppArgb);
            Graphics graphic = Graphics.FromImage(screenshot);
            mainStream = client.GetStream();
        }
    }
}
```

```

RijndaelManaged RMCrypto = new RijndaelManaged();
CryptoStream CryptStream = new CryptoStream(mainStream, RMCrypto.CreateEncryptor(Key, IV),
CryptoStreamMode.Write);
BinaryWriter bw = new BinaryWriter(CryptStream);
MemoryStream ms = new MemoryStream();

while (true)
{
    graphic.CopyFromScreen(bounds.X, bounds.Y, 0, 0, bounds.Size, CopyPixelOpera-
tion.SourceCopy);
    ms.Position = 0;
    screenshot.Save(ms, ImageFormat.Png);
    var arr = ms.ToArray();
    bw.Write(arr.Length);
    bw.Write(arr);
    bw.Flush();
}
}

private void SendDesktopImage()
{
    BinaryFormatter binaryFormatter = new BinaryFormatter();

    //StreamWriter streamWriter = new StreamWriter(CryptStream);
    //streamWriter.Write(mainStream);

    binaryFormatter.Serialize(CryptStream, GrabDesktop());
}

public Form1()
{
    InitializeComponent();
}

private void button1_Click(object sender, EventArgs e)
{

```



```

portNumber = int.Parse(textPort.Text);
try
{
    client.Connect(textIp.Text, portNumber);
    MessageBox.Show("Connected!");
}
catch (Exception)
{
    MessageBox.Show("Failed to connect!");
}

private void button2_Click(object sender, EventArgs e)
{
    if (button2.Text.StartsWith("Share"))
    {
        timer1.Start();
        button2.Text = "Stop sharing";
    }
    else
    {
        timer1.Stop();
        button2.Text = "Share screen";
    }
}

private void timer1_Tick(object sender, EventArgs e)
{
    SendDesktopImage();
}
}
}

```

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ им. Н.П. ОГАРЁВА»  
(ФГБОУ ВО «МГУ им. Н.П. Огарёва»)

**ОТЧЕТ**

*о результатах проверки бакалаврской работы обучающегося  
на наличие заимствований*

Автор работы Воробьев Юрий Юрьевич

Тема работы «Разработка клиентского приложения для удаленного  
администрирования ПК»

Руководитель Шибайкин Сергей Дмитриевич

Представленная работа прошла проверку на наличие заимствований в  
системе «Антиплагиат. ВУЗ»

Результаты автоматической проверки: оригинальность 69,06%  
цитирования 0,36%  
заимствования 30,58%

Результаты анализа полного отчета на наличие заимствований:

правомерные заимствования: 69,06%

корректные цитирования: 0,36%

неправомерные заимствования: нет

признаки обхода системы: нет

Общее заключение об итоговой оригинальности работы и возможности ее допуска к защите: система показала, что оригинальный текст в проверяемом документе составляет 69,06%, остальные 30,58% присутствуют в 15 источниках. Анализ источников показал, что в исследуемом тексте имеются корректные совпадения в виде наименований публикаций, конференций, терминологии и устойчивых словосочетаний, принятых в данной сфере науки, нормативных и правовых документах.

Таким образом, студент Воробьев Ю.Ю. допускается к сдаче ВКР.

Руководитель

С.Д. Шибайкин

Должность

канд. техн. наук, доцент

## ОТЗЫВ

*на бакалаврскую работу Воробьева Юрия Юрьевича студента 4 курса дневного отделения института электроники и светотехники направления подготовки 11.03.02 «Инфокоммуникационные технологии и системы связи» на тему «Разработка клиентского приложения для удаленного администрирования ПК»*

Бакалаврская работа Воробьева Ю.Ю. выполнена в полном соответствии с заданием и представлена к заданному сроку. Она состоит из трёх разделов и двух приложений А и Б.

Целью работы является разработка клиентского приложения для удаленного администрирования ПК.

В процессе работы Воробьев Ю.Ю. показал себя грамотным специалистом, способным самостоятельно решать поставленные перед ним задачи. Он умеет системно мыслить, уверенно ориентируется в основных понятиях связи, может чётко сформулировать задачи для достижения поставленной цели и успешно их решает.


В процессе работы проводился обзор существующих приложений для удаленного администрирования ПК, исследование сетевой структуры «клиент-сервер», создание приложения в среде разработки Visual Studio.

В целом выпускная работа логично построена, оформлена на высоком инженерно-техническом уровне.

Результаты разработки показывают, что поставленная перед студентом задача полностью выполнена.

Считаю, что выпускная квалификационная работа заслуживает оценки «отлично», а студент Воробьев Юрий Юрьевич – присвоения квалификации бакалавр по направлению подготовки «Инфокоммуникационные технологии и системы связи».

Руководитель работы \_\_\_\_\_

  
С.Д.Шибайкин