



На правах рукописи

САВЕЛЬЕВ
Антон Игоревич

**АРХИТЕКТУРЫ, АЛГОРИТМЫ И ПРОГРАММНЫЕ СРЕДСТВА ОБРАБОТКИ
ПОТОКОВ МНОГОМОДАЛЬНЫХ ДАННЫХ В ПИРИНГОВЫХ ВЕБ-
ПРИЛОЖЕНИЯХ ВИДЕОКОНФЕРЕНЦСВЯЗИ**

Специальность 05.13.11 – Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

АВТОРЕФЕРАТ

диссертации на соискание ученой степени
кандидата технических наук

Санкт-Петербург - 2016

Работа выполнена в Федеральном государственном бюджетном учреждении науки Санкт-Петербургском институте информатики и автоматизации Российской академии наук (СПИИРАН).

Научный руководитель:

доктор технических наук, профессор
Ронжин Андрей Леонидович

Официальные оппоненты:

доктор технических наук
Матвеев Юрий Николаевич,
Федеральное государственное автономное образовательное учреждение высшего образования Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, заведующий кафедрой речевых информационных систем;

доктор технических наук, профессор,
Одоевский Сергей Михайлович,
Федеральное государственное казенное военное образовательное учреждение высшего профессионального образования «Военная академия связи имени Маршала Советского Союза С.М. Буденного», профессор кафедры сетей связи и систем коммутации.

Ведущая организация:

Федеральное государственное бюджетное учреждение науки Институт проблем управления им. В.А. Трапезникова Российской академии наук.

Защита состоится «15» декабря 2016 г. в 12.00 часов на заседании диссертационного совета Д.002.199.01 при Федеральном государственном бюджетном учреждении науки Санкт-Петербургском институте информатики и автоматизации Российской академии наук по адресу: 199178, Санкт-Петербург, В.О., 14 линия, 39.

С диссертацией можно ознакомиться в библиотеке Федерального государственного бюджетного учреждения науки Санкт-Петербургского института информатики и автоматизации Российской академии наук

Автореферат разослан « » _____ 2016 г.

Ученый секретарь
диссертационного совета Д.002.199.01
кандидат технических наук



Фаткиева Р.Р.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы диссертации. При разработке систем видеоконференцсвязи основное внимание уделяется методам мультимедийной обработки данных, способам передачи данных, оптимизации архитектуры соединения клиентских приложений. Реализация систем видеоконференцсвязи в мобильных устройствах накладывает свои ограничения на качество и объем обрабатываемой информации вследствие недостаточных вычислительных и сетевых встроенных ресурсов мобильных гетерогенных устройств.

Недостаточная пропускная способность каналов связи для передачи стремительно растущих объемов передаваемых мультимедийных данных требует разработки новых методов их передачи. В многопользовательских системах видеоконференцсвязи между всеми участниками сеанса связи необходимо одновременно передавать несколько идентичных мультимедийных данных, что существенно увеличивает объем потоков и нагрузку на сервер.

Помимо перечисленных недостатков существует проблема встраиваемости приложений для использования их в более крупных системах. Данная проблема обусловлена тем, что все существующие приложения являются законченным программным продуктом и не располагают функционалом для интеграции в сторонние системы.

Поэтому актуальность разработки архитектур, алгоритмов и программных средств автоматической обработки мультимедийных потоков данных в пиринговых (peer-to-peer) веб-приложениях видеоконференцсвязи, обеспечивающих сокращение объема передаваемых данных и возможность построения речевых и многомодальных интерфейсов для инфокоммуникационных приложений, подтверждается отсутствием кроссплатформенного программно-аппаратного обеспечения гетерогенных клиентских приложений, поддерживающих многоканальную коммуникацию удаленных абонентов.

Степень разработанности темы. Проблеме анализа и синтеза цифровых сетей с интеграцией служб, обеспечивающих перенос в сессии различных типов информации единым образом в общей физической среде, посвящены многие научные и практические исследования отечественных и зарубежных ученых, которые проводятся с конца 60-х гг. прошлого столетия (Амосов А.А., Гольдштейн Б.С., Захаров Г.П., Ершов В.А., Кутузов О.И., Кучерявый А.Е., Голд Б., Джитман И., Франк Х., Клейнрок Л, и др.). Большой вклад в совершенствование инфокоммуникационных технологий внесли Полонников Р.И., Александров В.В., Гонсалес Р., Вуддс Р. И др. Проблемами совершенствования методов и алгоритмов маршрутизации в вычислительных сетях занимались такие ученые, как Д. Бертсекас, Д. Гарсиа-Диас, П. Гупта, А.Б. Гольдштейн, Б.С. Гольдштейн, Д. Кантор, О.Я. Кравец, Д.В. Куракин, И.П. Норенков, А. Филипс, С. Флloyd и др.

Цель и задачи исследования. Основной целью диссертационной работы является разработка архитектур, алгоритмов и программных средств автоматической обработки потоков данных в пиринговых веб-приложениях видеоконференцсвязи, обеспечивающих сокращение объема передаваемых данных и снижение потребляемых ресурсов сервера и распределенных гетерогенных клиентских устройств при многоканальной обработке многомодальных потоков данных во время сеансов связи. Для достижения указанной цели в работе сформулированы и решены следующие задачи:

1. Анализ современных архитектур систем видеоконференцсвязи, а также методов и программных средств многоканальной обработки потоков аудиовизуальных и служебных потоков данных.
2. Разработка архитектур клиентской и серверной частей в пиринговых системах видеоконференцсвязи, обеспечивающих при передаче данных снижение потребляемых ресурсов сервера и клиентских приложений.
3. Разработка алгоритмов взаимодействия клиентской и серверной частей системы видеоконференцсвязи и установления соединений между клиентами, обеспечивающих распределение и обработку их потоков данных на клиентах.

4. Разработка программных средств клиентской и серверной части веб-приложения видеоконференцсвязи, обеспечивающих кроссплатформенность и многоканальные сеансы связи между гетерогенными устройствами распределенных абонентов.
5. Разработка методики тестирования пиринговых систем видеоконференцсвязи, включающей алгоритмы функционального тестирования, набор тестов, оценивающих потребляемые ресурсы, и ручное тестирование.
6. Разработка приложения видеоконференцсвязи, его верификация и тестирование по предложенной методике и сравнение с существующими аналогами.

Научная новизна:

1. Разработаны архитектуры клиентской и серверной частей в пиринговых многопользовательских системах видеоконференцсвязи, отличающиеся применением «безсерверного» принципа коммуникации клиентов, обеспечивающие сокращение количества промежуточных узлов для потоков многомодальных данных с учетом ограниченных вычислительных и сетевых встроенных ресурсов гетерогенных устройств абонентов.

2. Разработаны алгоритмы взаимодействия клиентской и серверной частей системы видеоконференцсвязи, отличающиеся поддержкой взаимодействия групп клиентов по протоколу WebRTC, распределением передаваемых мультимедийных данных по категориям, обработкой служебных данных на сервере.

3. Разработана архитектура программных средств клиентской и серверной части веб-приложения видеоконференцсвязи, обеспечивающая кроссплатформенность за счет реализации веб-интерфейса с использованием языков HTML, CSS и JavaScript, а также сокращение количества промежуточных узлов для передаваемых мультимедийных данных по протоколу WebRTC за счет программной реализации «безсерверных» соединений клиентских частей приложения в процессе сеанса связи.

4. Разработана методика тестирования пиринговых систем видеоконференцсвязи, включающая функциональное, нагрузочное, ручное тестирование, которая позволяет определить потребляемые ресурсы устройств, улучшить эргономику пользовательского интерфейса, выявить отклонения и нарушения в работе системы, следить за использованием оперативной памяти, а также проводить сравнение с аналогичными системами видеоконференцсвязи.

Теоретическая и практическая значимость работы

Разработанные архитектуры, алгоритмы и программные средства предназначены для обработки мультимедийной информации между удаленными участниками видеоконференцсвязи напрямую, без использования промежуточного сервера. Кроссплатформенные веб-технологии позволяют адаптировать работу приложения под различные программно-аппаратные платформы и обеспечивать контроль над обычными и управляемыми аккаунтами в единообразной адаптируемой среде в режиме реального времени. Возможность встраивания разработанного приложения позволит организовывать сложные программно-архитектурные решения, которые обеспечат его работу в составе облачных систем. Сама система видеоконференцсвязи содержит модуль для создания контролируемых аккаунтов и принудительного соединения с ними. Данная особенность позволяет организовывать взаимодействие с пользователями, когда требуется построить систему временного удаленного аудиовизуального взаимодействия с принудительным соединением с абонентом. Управляемые аккаунты могут быть использованы в системах контроля и управления доступом, для обеспечения контроля за рабочими местами, производством и другими объектами, требующими периодического аудиовизуального контакта. Эффективное отображение и передача аудиовизуальной информации, набор дополнительных модулей и возможность встраивания в другие программные решения - одни из главных достоинств пиринговых веб-приложений видеоконференцсвязи, поэтому разрабатываемые технологии будут востребованы государственными министерствами и ведомствами, крупными корпорациями, общественно-политическими организациями, активно использующими удаленное аудиовизуальное взаимодействие.

Методология и методы исследования. Для решения поставленных задач в работе использовались методы теории передачи данных, методы цифровой обработки сигналов, теории информации, теории множеств, теории объектно-ориентированного проектирования и программирования.

Положения, выносимые на защиту:

1. Разработанные архитектуры клиентской и серверной частей в пиринговых многопользовательских системах видеоконференцсвязи обеспечивают сокращение количества узлов для передаваемых потоков многомодальных данных с учетом ограниченных вычислительных и сетевых встроенных ресурсов гетерогенных абонентских устройств.

2. Алгоритмы взаимодействия клиентской и серверной частей системы видеоконференцсвязи и установления пирингового соединения между абонентами обеспечивают поддержку взаимодействия групп абонентов, распределение и обработку мультимедийных потоков данных, передачу служебных данных для пирингового соединения.

3. Программные средства клиентской и серверной части веб-приложения видеоконференцсвязи обеспечивают кроссплатформенность за счет реализации веб-интерфейса с использованием языков HTML, CSS и JavaScript, а также сокращение передаваемых потоков многомодальных данных по протоколам WebRTC между клиентами.

4. Методика тестирования пиринговых систем видеоконференцсвязи, включающая алгоритмы функционального тестирования, набор нагрузочных тестов, ручное тестирование, которая позволяет оценить потребляемые ресурсы и эргономику пользовательского интерфейса, выявить отклонения и нарушения в работе системы, следить за использованием оперативной памяти, а также проводить сравнение систем видеоконференцсвязи друг с другом.

Степень достоверности и апробация результатов. Достоверность научных положений, основных выводов и результатов диссертации обеспечивается за счет анализа состояния исследований в данной области, согласованности теоретических выводов с результатами экспериментальной проверки алгоритмов, а также апробацией основных теоретических положений диссертации в печатных трудах и докладах на международных научных специализированных конференциях.

Исследования, отраженные в диссертации, проведены в рамках прикладных и поисковых научно-исследовательских работ: соглашение с Министерством образования и науки РФ в рамках ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2014-2020 годы» (проект «Исследование и разработка системы аудиовизуального распознавания речи на базе микрофона и высокоскоростной видеокамеры», соглашение № 14.616.21.0056) в 2015-2016 гг.; грант РФФИ № 15-07-06774 «Разработка методов обработки и обмена мультимедийными данными в пиринговых веб-приложениях многоточечной видеоконференцсвязи» в 2015-2017 гг.; грант РФФИ № 13-08-00741 «Разработка методов и кроссплатформенных программных средств аудиовизуального сопровождения мобильных мероприятий» в 2013-2015 гг.; Грант Президента РФ № МД-3035.2015.8 «Разработка математического и программного обеспечения многомодальной ассистивной технологии для помощи людям с ограниченными возможностями здоровья» в 2015-2016 гг.

Разработанные архитектуры, алгоритмы, программные средства были также использованы при выполнении поисковой НИР Результаты диссертационного исследования представлялись на международной конференции NEW2AN/ruSMART (Санкт-Петербург, 2012, 2014), международной конференции Pattern Recognition and Image Analysis: New Information Technologies, PRIA-2013 (Самара, 2013); международной конференции SPECOM (Греция, 2015; Венгрия 2016), международной конференции ICR 2016 (Венгрия, 2016), IEEE международной конференции ICARSC 2016 (Португалия, 2016), XXIX международной научной конференции «Математические методы в технике и технологиях (ММТТ-29)» (Санкт-Петербург, 2016).

Публикации. По материалам диссертации опубликовано 30 печатных работ, включая 6 публикаций в научных журналах, рекомендованных ВАК, 9 публикаций в изданиях, индексируемых в WoS/Scopus.

Личный вклад автора. Теоретические выводы и практические решения, результаты тестирования. Основные научные положения сформулированы и изложены автором самостоятельно.

Структура и объем работы. Диссертация объемом 135 машинописных страниц содержит введение, четыре главы и заключение, список литературы (116 наименований), 8 таблиц, 37 рисунков, 1 приложение с копиями актов внедрения.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована важность и актуальность темы диссертации, сформулированы цели диссертационной работы и решаемые задачи, определена научная новизна работы и ее практическая значимость.

В первой главе диссертации проведен анализ и описаны основные проблемы, возникающие при разработке систем видеоконференцсвязи. На текущий момент существует широкий спектр приложений видеоконференцсвязи широко используемых в различных сферах жизнедеятельности человека. С увеличением числа участников видеоконференцсвязи повышается объем передаваемых мультимедийных и служебных данных. Также возрастает нагрузка на само приложение: увеличивается количество входящих и исходящих потоков данных и возрастает число обрабатываемых и выводимых данных устройством пользователя. Приложения видеоконференцсвязи, работающие на настольных компьютерах, в большинстве случаев располагают необходимыми ресурсами для обработки данных, но при большом количестве входящих и исходящих потоков возможны проблемы из-за перегрузки центрального процессора, оперативной памяти и видеокарты устройства. На мобильных устройствах ситуацию усугубляет отсутствие ресурсов, необходимых для обработки больших объемов данных, и небольшие по размерам дисплеи, не способные корректно отображать более двух участников одновременно.

Специализированные системы совместной работы помимо передачи аудиовизуальных потоков данных, захватываемых микрофонами и видеокамерами, обеспечивают обмен дополнительной мультимедийной информацией, например, передачу текущего слайда презентации, а также возможность совместного редактирования документов и рукописных набросков. На сегодняшний день существует широкий спектр Интернет-приложений для видеоконференцсвязи и совместной работы в области обучения, бизнеса и индустрии развлечений. В зависимости от назначения системы Интернет-связи можно разделить на средства общения, проведения видеоконференций и совместной работы. Данные приложения Интернет-связи могут обладать множеством дополнительных возможностей, позволяющих наладить более продуктивную работу абонентов. Другим классификационным признаком систем связи является тип клиентского устройства, по которому можно разделить системы на мобильные приложения, приложения для настольных компьютеров/ноутбуков и кроссплатформенные приложения. К сожалению, лишь не многие системы веб-конференций обладают кроссплатформенностью, которая достигается за счет встраивания приложения веб-страницы. Малое количество систем видеоконференцсвязи, имеющих клиентские части, которые поддерживают возможность кроссплатформенной работы без предустановки приложения в систему, говорит об актуальности разработок в данном направлении.

Большинство приложений видеоконференцсвязи в основном имеют клиент-серверную архитектуру. Данная архитектура позволяет одновременно взаимодействовать большому количеству участников и проводить широковещательные видеоконференции. Преимуществом данной архитектуры является низкая нагрузка на клиентскую часть приложения, которая используется для получения и вывода данных. Основную нагрузку по обработке, распределению и передаче данных в такой архитектуре осуществляет серверная часть. Такие системы удобно использовать для совещаний сотрудников в крупных

компаниях, проведения масштабных мероприятий и других событий, где необходимо взаимодействие множества участников. Несмотря на все преимущества данной архитектуры, главным ее недостатком является серверное оборудование, стоимость которого изначально высока и растет пропорционально количеству возможных участников данного вида связи. Программное обеспечение данных систем не является бесплатным и финансово доступно только крупным организациям.

Главным преимуществом пиринговой архитектуры является передача данных напрямую между клиентскими частями приложения без использования промежуточного сервера. Поскольку обработка и распределение данных происходит в клиентской части приложения, приложение потребляет большое количество ресурсов устройства, и необходима дополнительная система, обеспечивающая хранение информации о других пользователях. В результате приложения, построенные на базе пиринговой архитектуры, позволяют осуществлять видеоконференцсвязь с небольшим количеством участников. Однако такие системы являются самыми востребованными среди обычных пользователей. Таким образом, при анализе архитектур, возможностей и количества пользователей существующих систем видеоконференцсвязи наиболее перспективными были выбраны «безсерверные» пиринговые системы, обеспечивающих сокращение количества узлов для передаваемых данных и возможность построения речевых и многомодальных интерфейсов для данного типа телекоммуникационных приложений.

Вторая глава посвящена формальному описанию задачи синтеза архитектур клиентской, серверной частей и обмена данными в пиринговых многопользовательских системах видеоконференцсвязи, а также разработанных архитектур и алгоритмов.

Для формального описания задачи синтеза архитектур пиринговых многопользовательских систем видеоконференцсвязи введено множество возможных типов архитектур $A = \{ A_\chi, \chi \in ST \}$, в качестве которых выделяются архитектура клиентской части $A_{\text{КЛИЕНТ}}$, архитектура серверной части $A_{\text{СЕРВЕР}}$ и архитектура обмена данными $A_{\text{ОБМЕН}}$. Для связи перечисленных множеств друг с другом введем в рассмотрение системный динамический альтернативный мультиграф следующего вида:

$$A_\chi^t = \langle X_\chi^t, F_\chi^t, Z_\chi^t \rangle, \quad (1)$$

где χ – индекс, характеризующий тип архитектуры приложения видеоконференцсвязи, $\chi = \{1, 2, 3\}$ – множество индексов, соответствующих архитектуре клиентской, серверной частей и обмена данными соответственно; $t \in T$ – множество моментов времени; $X_\chi^t = \{x_{\chi l}^t, l \in L_\chi\}$ – множество элементов, входящих в состав архитектуры A_χ^t (множество вершин графа) в момент времени t ; $F_\chi^t = \{f_{\langle \chi, l, l' \rangle}^t, l, l' \in L_\chi\}$ – множество дуг графа типа A_χ^t , отражающих взаимосвязи между его элементами в момент времени t ; $Z_\chi^t = \{z_{\langle \chi, l, l' \rangle}^t, l, l' \in L_\chi\}$ – множество значений параметров, количественно характеризующих взаимосвязь соответствующих элементов графа. Зададим множество допустимых операций отображения указанных выше графов друг на друга:

$$M_{\langle \chi, \chi' \rangle}^t : F_\chi^t \rightarrow F_{\chi'}^t, \quad (2)$$

а также операции композиции указанных отображений:

$$M_{\langle \chi, \chi' \rangle}^t = M_{\langle \chi, \chi_1 \rangle}^t \circ M_{\langle \chi_1, \chi_2 \rangle}^t \circ \dots \circ M_{\langle \chi_n, \chi' \rangle}^t.$$

Тогда многоархитектурное состояние можно определить как подмножество декартова произведения множеств элементов, на которых строятся соответствующие архитектуры приложения видеоконференцсвязи:

$$S_\delta \subseteq X_1^t \times X_2^t \times X_3^t, \quad \delta = 1, \dots, K_\Delta$$

Множество многоархитектурных состояний приложения видеоконференцсвязи запишется в следующем образом:

$$S = \{S_\delta\} = \{S_1, \dots, S_{K_\Delta}\}.$$

Введём множество допустимых операций отображения многоархитектурных состояний приложения видеоконференцсвязи друг на друга:

$$\Pi_{\langle \delta, \delta' \rangle}^t : S_\delta \rightarrow S_{\delta'}.$$

При этом будем полагать, что каждое многоархитектурное состояние приложения видеоконференцсвязи задаётся в результате операции композиции соответствующих графа, описывающих каждый тип архитектуры (2).

Тогда для синтеза архитектур пиринговых многопользовательских систем видеоконференцсвязи может быть предложено ее следующее *теоретико-множественное описание*: необходимо разработать модельно-алгоритмическое обеспечение, позволяющее находить такие $\langle U^t, S_{\delta^*}^{t_f} \rangle$, при которых выполняются следующие условия:

$$J_\theta \left(X_\chi^t, F_\chi^t, Z_\chi^t, M_{\langle \chi, \chi' \rangle}^t, \Pi_{\langle \delta, \delta' \rangle}^t, t \in (t_0, t_f] \right) \rightarrow_{\langle U^t, S_{\delta^*}^{t_f} \rangle \in \Delta_g} \text{extr}, \quad (3)$$

$$\Delta_g = \left\{ \langle U^t, S_{\delta^*}^{t_f} \rangle \mid R_\beta \left(X_\chi^t, F_\chi^t, Z_\chi^t, M_{\langle \chi, \chi' \rangle}^t, \Pi_{\langle \delta, \delta' \rangle}^t \right) \leq \tilde{R}_g; U^t = \Pi_{\langle \delta_1, \delta_2 \rangle}^t \circ \Pi_{\langle \delta_2, \delta_3 \rangle}^t \circ \Pi_{\langle \delta_3, \delta_4 \rangle}^t; \beta \in \mathbf{B} \right\},$$

где U^t – управляющие воздействия, позволяющие синтезировать как архитектуры приложения видеоконференцсвязи, так и алгоритмы и программные средства функционирования ее элементов; J_θ – стоимостные, временные, ресурсные показатели, характеризующие качество функционирования системы видеоконференцсвязи; Δ_g – множество динамических альтернатив (множество A_χ архитектур приложения видеоконференцсвязи, их параметров Z_χ^t); \mathbf{B} – множество номеров пространственно-временных, технических и технологических ограничений, определяющих процессы обработки и передачи информации в пиринговых приложениях; \tilde{R}_g – заданные величины; $T = (t_0, t_f]$ – интервал времени, на котором синтезируются как архитектуры, так и программно-алгоритмические решения для приложения видеоконференцсвязи.

В качестве показателей J_θ качества функционирования таких систем на этапах синтеза ее различных архитектур целесообразно использовать объемы передаваемых данных и потребляемых ресурсов сервера и клиентских приложений.

Архитектура клиентской части пирингового приложения видеоконференцсвязи, представленная на рисунке 1, содержит три основные составляющие: модуль логики приложения $M_{\text{лп}}$ видеоконференцсвязи, модуль графического интерфейса $M_{\text{ги}}$ и хранилище состояний $M_{\text{хс}}$. Модули логики $M_{\text{лп}}$ и графического интерфейса $M_{\text{ги}}$ приложения с помощью модуля действий $M_{\text{д}}$ и модуля изменения состояния хранилища $M_{\text{их}}$ осуществляют передачу новых данных $Data$ в хранилище. Модуль действий $M_{\text{д}}$ генерирует сообщение S , что произошло некоторое изменение данных в приложении.

Все возможные изменения данных описываются типами действий $Type_{\text{д}}$, которые заранее предопределяются разработчиком. Чтобы использовать действие, необходимо вызвать функцию $F_{\text{д}}$ данного действия, которая отправит сообщение S модулю изменения хранилища $M_{\text{их}}$. Отправленное сообщение S должно содержать тип действия $Type_{\text{д}}$ и новую информацию I , предназначенную для хранилища $M_{\text{хс}}$. Одним из главных алгоритмов данного модуля $M_{\text{хс}}$ является алгоритм изменения данных в хранилище состояний, который реализуется в модуле изменения хранилища $M_{\text{их}}$. Модуль изменения хранилища $M_{\text{их}}$ обеспечивает обработку информации I от модуля действий $M_{\text{д}}$ и создает

новое состояние $State_D$ данных для хранилища. Важно отметить, что модуль изменения состояния хранилища $M_{ИХ}$ должен создавать новый набор данных $Data$, а не модифицировать старые. Данное ограничение обусловлено тем, что приложение видеоконференцсвязи должно хранить именно промежуточные состояния, которые легко отслеживать и отлаживать при разработке и работе приложения видеоконференцсвязи. В итоге модуль изменения хранилища $M_{ИХ}$ указывает, как состояние $State_{П}$ приложения видеоконференцсвязи должно измениться в ответ на определенное действие, произошедшее в данном приложении.

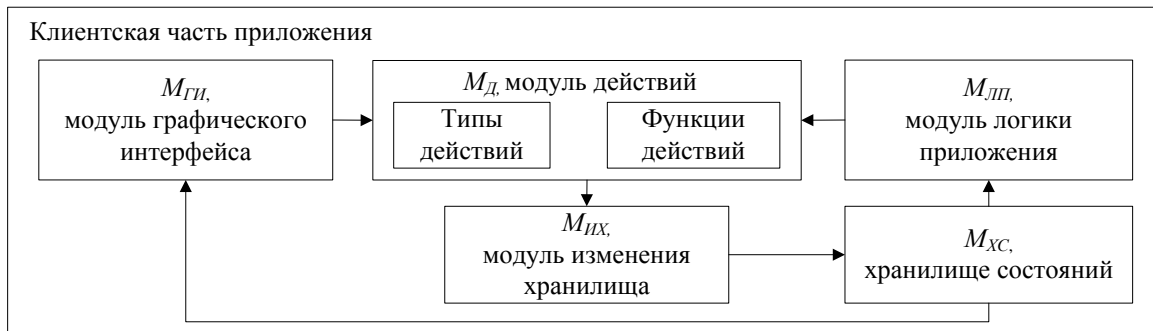


Рис. 1 Архитектура клиентской части пирингового приложения видеоконференцсвязи

Хранилище состояний $M_{ХС}$ содержит типы данных $Type_{Data}$, которые соответствуют определенному объекту управления O_y . При изменении одного из состояний $State_{П}$ в приложении видеоконференцсвязи генерируется оповещение для логики $Not_{Л}$ или графического интерфейса $Not_{ГИ}$ данной архитектуры. При получении такого оповещения, модуль логики $M_{ЛП}$ или графического интерфейса $M_{ГИ}$ сможет обработать новые данные $Data$ и обеспечить дальнейшее функционирование системы.

Для обеспечения хранения промежуточных данных $Data_{ПРОМ}$, данных пользователей $Data_{ПОЛ}$ и соединения с другими распределенными модулями в данной архитектуре используется серверная часть, архитектура которой показана на рисунке 2.



Рис. 2. Архитектура серверной части приложения видеоконференцсвязи

Для обеспечения авторизации и обмена сигнальными данными $Data_{СИГН}$ в серверной части приложения используется модуль взаимодействия с клиентами $M_{ВК}$. Алгоритмы, определяющие взаимосвязи между элементами графа $A_{СЕРВЕР}$, (рис. 3-5), описывают обработку «сигнальных» данных на клиентской и серверной части. В алгоритме подготовки клиентских частей перед формированием сигнальных данных (рис. 3) сначала 'вызывающий' клиент подает запрос на сервер для соединения с 'отвечающим' клиентом по протоколу WebSocket. Далее сервер производит поиск 'отвечающего' клиента среди подключенных. Если клиента нет, то сервер завершит звонок 'вызывающего' клиента, если 'отвечающий'

клиент найден, то ему отправляется запрос на соединение по протоколу WebSocket. 'Отвечающий' клиент формирует и отправляет ответ на запрос. Если ответ отрицательный, то сервер завершает звонок 'вызывающего' клиента, в случае положительного ответа: сервер получит id сокетов 'вызывающего' и 'отвечающего' клиентов, по id сокетов найдет 'комнату', в которых сокет находится в данный момент. После завершения данного алгоритма, происходит выполнение *алгоритма распределения сокетов и обработки их буферов на сервере* (рис. 4).

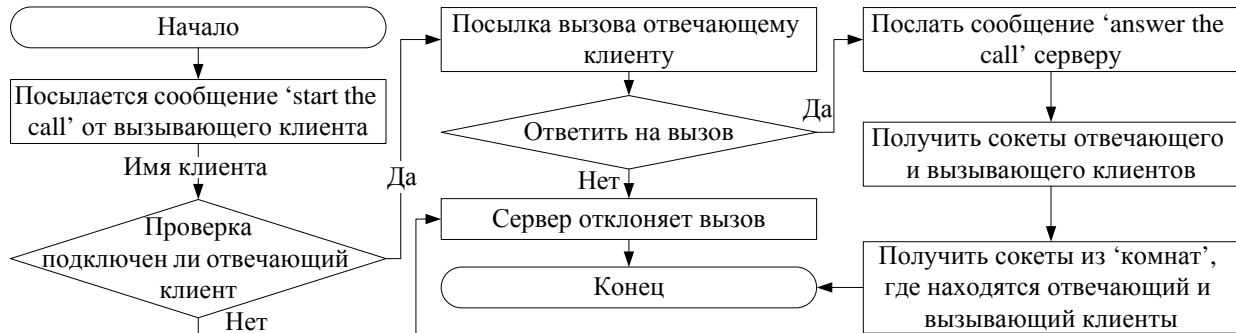


Рис. 3. Алгоритм подготовки клиентских частей перед формирования «сигнальных» данных

Сначала извлекается сокет из 'комнаты' 'отвечающего' клиента, для него создается буфер для хранения сокетов, ожидающих соединения по протоколу WebRTC. Затем сокет из 'комнаты' 'отвечающего' клиента добавляет в уже существующий буфер — сокет, взятый из 'комнаты' 'вызывающего' клиента. Если взятый сокет из 'комнаты' 'вызывающего' клиента был не последним, то операция извлечения сокета из 'комнаты' 'вызывающего' клиента и его добавление в буфер - повторяется со следующим сокетом из этой комнаты. После того как будет взят последний сокет из 'комнаты' 'вызывающего' клиента, сокет из 'комнаты' 'отвечающего' клиента отсоединяется от своей 'комнаты' и добавляется в 'комнату' 'вызывающего' клиента.

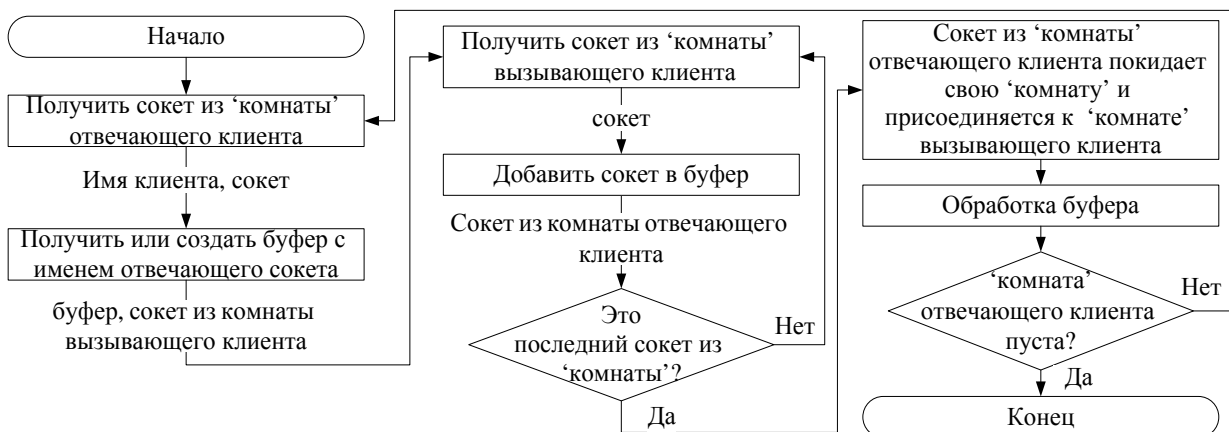


Рис. 4. Алгоритм распределения сокетов и обработки их буферов на сервере

Алгоритм работы с буфером данных на клиенте, представленный на рисунке 5, является общим для обработки запросов от сервера на формирование сигнальных данных 'call offer' или на обработку пришедших от другого клиента сигнальных данных 'call answer'. В клиентской части приложения видеоконференцсвязи существует два отдельных буфера для каждого типа данных.

Модуль обработки данных клиента $M_{\text{ОДК}}$ обеспечивает распределение, проверку и временное хранение данных, подключенных к серверу пользователей. Для работы с клиентом в серверной части создается копия его некоторых данных $Data_{\text{МОДК}}$. Для создания копии используется алгоритм обработки данных нового подключаемого клиента. После

создания копии клиента в серверной части для обеспечения передачи данных между клиентами одного пространства (находящимися в одной комнате), используется *алгоритм оповещения о новых данных в комнате*. Последний *алгоритм модуля обработки данных клиента* получает на вход данные от клиента, которые необходимо в дальнейшем сохранить.

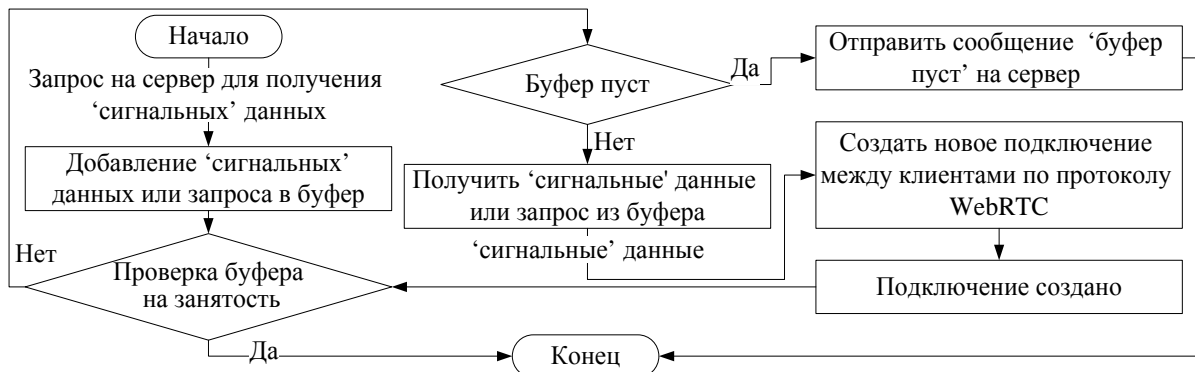


Рис. 5. Алгоритм работы с буфером данных на клиенте

Для сохранения, изменения и поиска информации в базе данных на сервере используется модуль взаимодействия с базой данных $M_{\text{БД}}$. Данное взаимодействие обеспечивает *алгоритм работы с базой данных*. Он обрабатывает ситуации взаимодействия с базой данных и случаи возникновения ошибок. Модуль сервера M_{API} обеспечивает внешнее взаимодействие сервера с другими серверами или приложениями. Данный модуль позволяет получить интересующую информацию другим устройствам, обеспечив формирование данных в заранее заданном виде. На данный момент модуль M_{API} облачного взаимодействия позволяет получать актуальную информацию о клиентах и некоторые их данные. Для сохранения, изменения и поиска информации в базе данных на сервере используется модуль взаимодействия с базой данных $M_{\text{БД}}$.

Для объединения серверной и клиентской частей приложения была разработана архитектура взаимодействия модулей веб-приложения видеоконференцсвязи, представленная на рисунке 6. Данная архитектура позволяет предотвратить потерю «сигнальных» данных при соединении трех и более участников видеоконференцсвязи. Основными структурными элементами архитектуры являются: 1 — клиентская часть приложения, 2 — серверная часть приложения, 3 — блок протоколов передачи данных. Клиентская часть подразделяется на две независимые составляющие — устройство пользователя и веб-страницу. Основным элементом архитектуры является серверная часть, выполняющая следующие функции: формирование клиентской части приложения; регистрация клиента; авторизация клиента; обмен «сигнальными» данными между клиентами; создание комнат чата и работу с базой данных.

Веб-страница клиентской части приложения включает классы, написанные на языке программирования JavaScript, необходимые для создания соединений с сервером и другими клиентами. Средства CSS и HTML служат для построения графического интерфейса, отображения данных и управления клиентской частью приложения. Язык JavaScript, используемый на веб-странице видеочата, включает в себя три различных типа инструкций, позволяющих организовать передачу данных по трем протоколам: WebRTC, WebSocket и HTTP. Также средства языка JavaScript служат для захвата и обработки потоков данных с микрофона и видекамеры.

Третий элемент архитектуры, представленной на рисунке 6, состоит из протоколов — HTTP, WebSocket и WebRTC. Эти протоколы обеспечивают обмен данными на различных этапах работы приложения, с их помощью осуществляется создание соединения клиентских частей по протоколу WebRTC для передачи потоковых аудио- и видеоданных между ними.

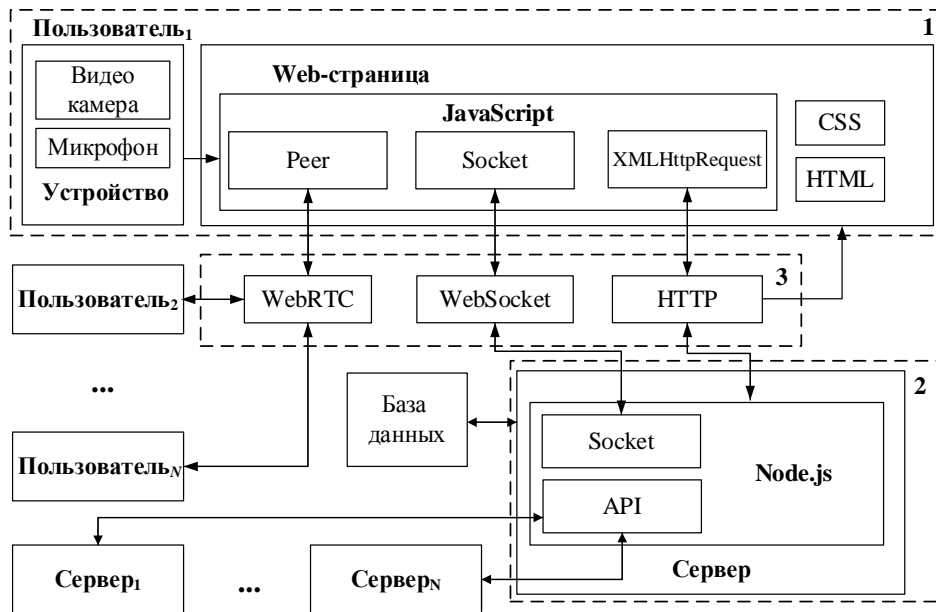


Рис. 6. Пиринговая архитектура обмена данными в приложении видеоконференцсвязи

Последний модуль сервера обеспечивает внешнее взаимодействие сервера с другими серверами, находящимися в единой облачной системе и позволяет получить интересующую информацию другим клиентским устройствам. Алгоритм, реализующий возможности модуля API облачного взаимодействия, представлен на рисунке 7.

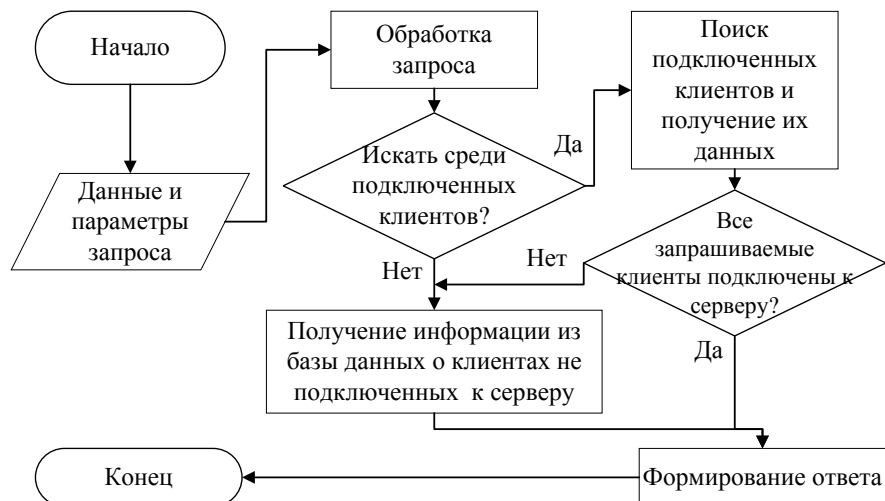


Рис. 7. Алгоритм работы API облачного взаимодействия

Работа данного алгоритма начинается с получения данных запроса и его параметров. Параметры запроса будут определять тип поиска данных клиентов. Обработчик запросов получает параметры и определяет нужно ли провести поиск среди подключенных клиентов. Поиск среди подключенных клиентов позволяет получить текущую актуальную информацию о данном клиенте, а также удостовериться в его соединении с сервером. Если искать среди подключенных клиентов нет необходимости, то модуль сформирует параметры и данные для запроса в базу данных. В случае если нужны данные о подключенных пользователях, модуль сначала попытается найти необходимую информацию среди подключенных пользователей. Затем если будет необходимо сформирует запрос в базу данных на получении информации о тех пользователях, которые не подключены. Последним шагом работы данного алгоритма будет формирование ответа на запрос с его последующей отправкой по сети.

Разработанные алгоритмы позволяют контролировать асинхронную архитектуру клиентской и серверной частей приложения, осуществляя обработку данных по

необходимости, препятствуя возникновению ситуаций перемешивания данных и прерывания выполняющихся соединений по протоколу WebRTC. Таким образом, в приложении достигается возможность создавать групповые видеоконференции и контролировать их состояние с помощью объединенных в «комнаты» клиентов на сервере.

В третьей главе описана архитектура разработанных программных средств приложения видеоконференцсвязи с использованием UML диаграмм прецедентов и классов. На рисунке 8 представлена диаграмма взаимодействия модулей в клиентской части приложения. Управляющий модулями класс Modules — это синглтон (класс существует в единственном экземпляре), который содержит в себе информацию об остальных модулях и умеет управлять данными, необходимыми для их работы. Внутри класса Modules происходит связь между другими модулями и данными, которыми они обмениваются.

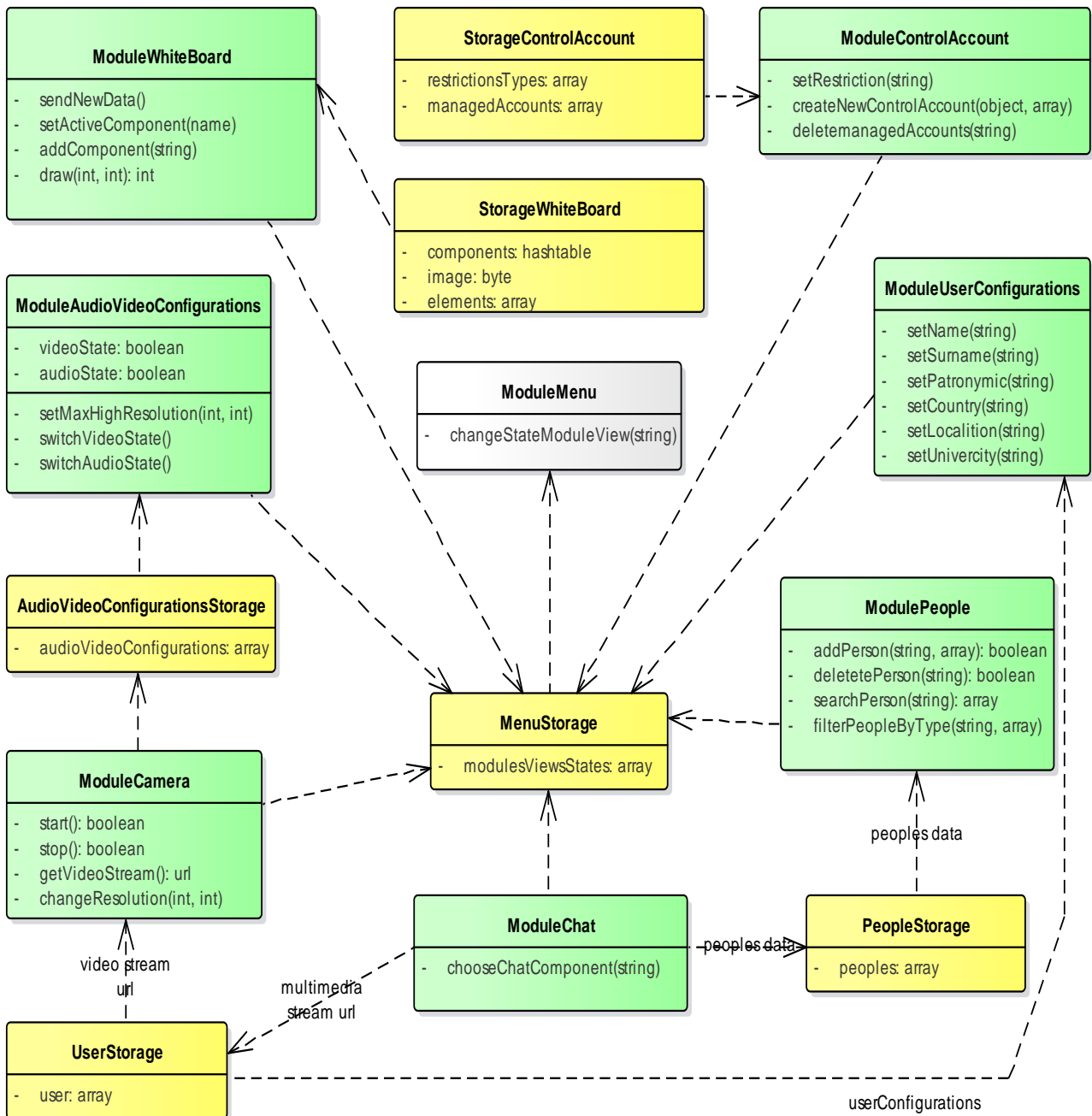


Рис. 8. Диаграмма взаимодействия модулей

Связь между модулями осуществляется посредством различных хранилищ состояний приложения. Одни модули инициируют изменения в данном хранилище, другие подписываются на данные изменения и получают оповещение при их изменении. Таким

образом осуществляется передача информации из одного модуля в другой, где прямые связи между модулями полностью отсутствуют и создаются абсолютно новые промежуточные состояния. Так как любой из модулей может быть видимым или скрытым в графическом интерфейсе, то все модули имеют связь с хранилищем состояний отображения. После запуска приложения в графическом интерфейсе программы по умолчанию отображается модуль камеры. Для изменения разрешения изображения, которое необходимо захватывать с камеры, используется функция `changeResolution()`, параметрами которой служат необходимые значения разрешения по ширине и высоте. Последняя функция в модуле камеры - `getVideoStream()`, которая обеспечивает получения url мультимедийного потока. Модуль настройки аудио/видео позволяет отключить захват видео- или аудиоданных с устройства при помощи функций `switchVideoState()` и `switchAudioState()` соответственно. Для изменения захватываемого разрешения с камеры служит функция `setMaxHighResolution()`.

Для настройки данных об аккаунте в приложении используется модуль настройки пользователя. Данный модуль обеспечивает изменение текущей информации об абоненте, которая необходима для его поиска и фильтрации. Набор функций присутствующих в данном модуле и начинающихся со словосочетания «set» имеют схожие задачи: считывают данные с введенных полей настроек пользователя и сохраняют их в пользовательском хранилище. При изменении определенных полей в пользовательском хранилище, новые данные пользователя отправятся на сервер, где будет произведено их сохранение в базе данных.

За создание управляемого аккаунта в приложении отвечает модуль нового управляемого аккаунта. Все ограничения, представленные в данном модуле, кроме ограничения на принудительное соединение, работают независимо от остальных. Пользователь, создавший аккаунт, может произвести удаление данного аккаунта с помощью функции `deletemanagedAccounts()`. Функция `deletemanagedAccounts()` удаляет все данные по аккаунту, в том числе его логин и пароль.

Последний модуль, связанный с графическим интерфейсом, используемым в клиентской части приложения, – модуль интерактивной доски. Данный модуль позволяет формировать изображения удаленным пользователем в едином пространстве в режиме реального времени. После изменения рабочей области модуль интерактивной доски, получит оповещение об изменении и новые данные от хранилища и вызовет функцию `sendNewData()`, которая отправит эти данные по пиринговому соединению всем подключенным в данный момент пользователям.

Программная реализация серверной части приложения содержит в себе четыре модуля: модуль взаимодействия с клиентами, модуль обработки данных клиента, модуль API для облачного взаимодействия и модуль взаимодействия с базой данных. Для совершения операций с базой данных в серверной части приложения используется модуль взаимодействия с базой данных. Данный модуль реализует запросы в базу данных при помощи ORM `Mongoose`. На основе `Mongoose` создана схема пользователя, которая повторяет поля пользователя в базе данных. Данная схема удобна тем, что легко преобразует структуру базы данных в объект, с которым удобно осуществлять взаимодействие на сервере.

Для обеспечения доступа к данным другими серверами используется модуль API для облачного взаимодействия. Данный модуль позволяет получить функционал страницы чата по протоколу `WebSocket`, чтобы отобразить данную страницу в составе другого клиентского приложения. За формирование и передачу страницы отвечает функция `getChatPage()`. Модуль API для облачного взаимодействия позволяет также получать данные о пользователях из базы данных и о подключенных в данный момент к серверу. Для получения данных из базы данных используется функция `getUserData()`. Функции `getConnectedUsers()` и `getUsersInVideoChat()`, возвращают массивы всех присоединенных пользователей и пользователей которые соединены в видеочате на данный момент.

В четвертой главе представлены результаты экспериментальной проверки разработанного приложения видеоконференцсвязи на основе предложенной методики,

включающей комплекс алгоритмов по нагрузочному тестированию и функциональному тестированию логики работы и корректности отклика графических элементов приложения.

Функциональное тестирование разработанного приложения было разделено на три части: тестирование графического интерфейса в клиентской части приложения, тестирование логики работы клиентской и серверной частей приложения. Алгоритм функционального тестирования графического интерфейса клиентской части приложения обеспечивает тестирование отклика и работоспособности отдельных компонентов различных веб-страниц и обеспечивает занесение результатов в отчет о тестировании. Для функционального тестирования логики работы клиентской и серверных частей был использован алгоритм, состоящий из следующих основных этапов работы: вызов тестируемой функции с определенными аргументами, сравнение итога работы функции с проверочными тестовыми данными и занесение результатов тестов в отчет. В ходе разработки приложения было проведено шесть итераций тестирования каждой из частей приложения. Каждая итерация тестирования была направлена на выявление проблем на различных этапах разработки приложения и их дальнейшего исправления. Благодаря функциональным тестам, удалось выявить некорректно функционирующие части приложения и обеспечить проверку работоспособности приложения видеоконференцсвязи.

Для оценки качества передаваемых аудиовизуальных данных и эргономики графического интерфейса было использовано ручное тестирование. Оценка, осуществляемая пользователями, показала, что качество передаваемых от одного клиента к другому аудиовизуальных данных не изменялось, в отличие от приложения Skype, где данные в некоторых временных промежутках были искажены у конечного пользователя в худшую сторону. Также на основе пользовательских оценок графического интерфейса были осуществлена модификация приложения, направленная на улучшение его эргономики.

Для анализа потребляемых ресурсов устройствами было проведено нагрузочное тестирование приложения с оцениванием объема потребляемой оперативной памяти и загруженности центрального процессора. Так как потоки данных для осуществления видеоконференцсвязи создаются только в клиентской части приложения, то для тестирования были выбраны именно клиентские устройства пользователей. С камер и микрофонов каждого из устройств был произведен захват аудио- и видеопотоков и соединение по протоколу WebRTC. Для организации работы в клиентской части приложения были использованы браузеры Firefox и Chrome. Тестирование проводилось в сравнении с наиболее популярной веб-версией многоканальной системой видеоконференцсвязи Skype, которая не поддерживается ни в одном мобильном браузере, в отличие от разработанного приложения, которое также имеет поддержку в браузере Chrome на Android платформах. Для работы Skype системы в браузерах на десктопных платформах необходимо было установить дополнительный плагин. Разработанная система работает без дополнительных средств и обеспечивает такой же функционал по передаче аудио- и видеоданных.

Усредненные результаты измерений, потребляемых ресурсов центральным процессором (ЦП) и оперативной памятью (ОП) в разработанном приложении и системе Skype на различных этапах работы приложений при одинаковых входных и выходных параметрах – одинаковы. Необходимо отметить, что приложения Skype на десктопных платформах работало только в браузере Firefox, разработанное приложение было работоспособно в трех браузерах: Chrome, Opera и Firefox. Таким образом при большей функциональности, отсутствии дополнительной установки плагинов и поддержки большего количества браузеров в том числе и на мобильной платформе Android, разработанное приложение видеоконференцсвязи обеспечивает такое же потребление ресурсов центрального процессора и оперативной памяти, как и система Skype.

В таблице 1 приведены оценки потребляемых ресурсов центрального процессора и оперативной памяти дополнительными модулями разработанной системы. Модуль контролируемого аккаунта не изменяет свои средние показатели потребления ресурсов на

всех режимах работы. Модуль интерактивной доски с присоединением новых участников видеоконференцсвязи использует большой объем ресурсов.

Для тестирования серверной части приложения был выбран обычный стационарный компьютер под управлением Linux Ubuntu 14.04 со следующими характеристиками: процессор Intel Core i5 6600 (4 ядра, 3.3 ГГц), видеокарта NVIDIA GeForce GTX 970 4 Гб, оперативная память 16 Гб DDR4, жесткий диск емкостью 2 Тб. Создан специальный набор тестов, который позволил осуществлять процессы соединения с сервером по различным протоколам, отправлять данные на сервер для их передачи другим пользователям и сохранения в базе данных, использовать возможности API облачного взаимодействия и осуществлять поиск и фильтрацию данных на сервере и в базе данных созданных тестовых клиентов, каждые 10 миллисекунд. Тестирование проходило в 3 этапа: запуск сервера, работа с четырьмя клиентами, работа со 100 клиентами. Оценки максимальных показателей потребления ресурсов на всех трех этапах отображены в таблице 2.

Таблица 1. Ресурсы, потребляемые дополнительными модулями

Режим работы	Потребляемые ресурсы			
	Модуль интерактивной доски		Модуль контролируемого аккаунта	
	ОП	ЦП	ОП	ЦП
Запуск приложения	15-20 кб	1-2%	2-10 кб	1%
1 соединение	3000-5000 кб	2-3%	2-10 кб	1%
3 соединения	10000-50000 кб	5-10%	2-10 кб	1%

Таблица 2. Ресурсы, потребляемые серверной частью

Устройства	Данные нагрузки		
	Запуск сервера	Минимальная нагрузка	Пиковая нагрузка
Центральный процессор(ЦП)	5%	12%	47%
Оперативная память(ОЗУ)	10 МВ	307МВ	1202МВ

Первый этап тестирования проходил на протяжении 10 минут, в ходе которого были выявлены незначительные отклонения от результатов, представленных в таблице 1, которые подтверждают стабильную работу сервера без нагрузки и корректную работу с памятью. Второй и третий этап проходили на протяжении 30 минут, каждые 20 секунд производилось снятие показаний о нагрузке на сервер, которые отображены на графиках, представленных на рисунке 9.

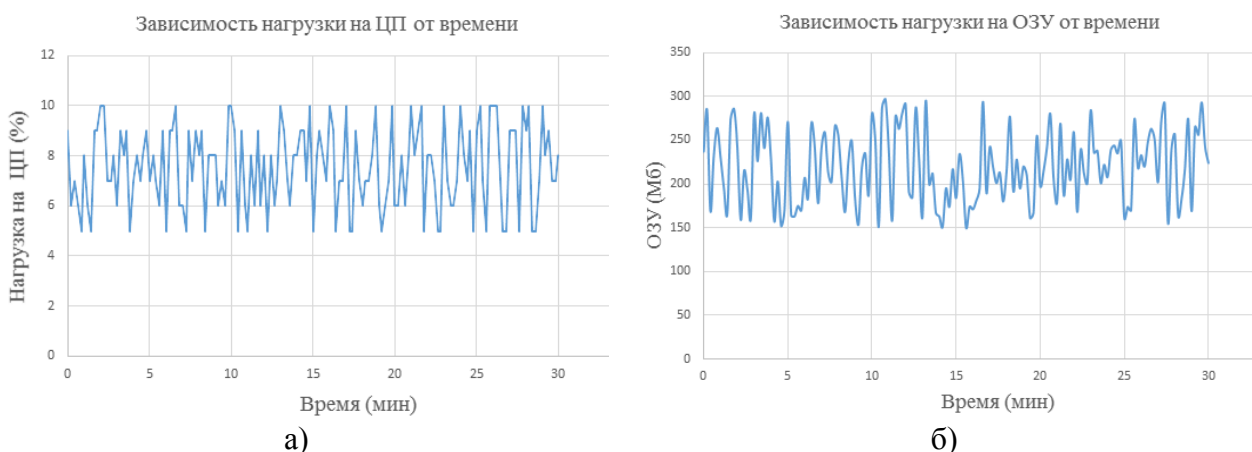


Рис. 9. Зависимость потребляемых ресурсов: а) центрального процессора; б) оперативной памяти при связи пяти клиентов

Нагрузочное тестирование серверной части позволило выявить, что работа сервера при отсутствии клиентов и во время различных нагрузок является стабильной, ресурсов обычного стационарного компьютера достаточно для одновременного обслуживания клиентов в режиме реального времени.

ЗАКЛЮЧЕНИЕ

Совокупность предложенных архитектур, алгоритмов и программных средств автоматической обработки потоков данных, а также их практическая реализация представляют собой решение актуальной научно-технической задачи поддержки многоканальной коммуникации удаленных абонентов на основе веб-приложений видеоконференцсвязи с пиринговой архитектурой, исполняемых на гетерогенных клиентских устройствах. Их внедрение вносит значительный вклад в развитие инфокоммуникационных технологий. При решении данной задачи были получены следующие результаты.

1. Архитектуры клиентской и серверной частей в системах видеоконференцсвязи, отличающиеся применением пирингового принципа коммуникации, обеспечивающего сокращение количества узлов для передаваемых данных, снижение потребляемых ресурсов сервера и распределенных гетерогенных клиентских устройств, а также возможность построения речевых и многомодальных интерфейсов для данного типа телекоммуникационных приложений.

2. Алгоритмы взаимодействия клиентской и серверной частей системы видеоконференцсвязи и установления соединений между клиентами по протоколу WebRTC, обеспечивающие распределение и обработку данных на сервере и клиенте, обработку «сигнальных» данных, поддержку взаимодействия между группами клиентов.

3. Программные средства клиентской и серверной частей веб-приложения видеоконференцсвязи обеспечивают кроссплатформенность за счет реализации веб-интерфейса с использованием языков HTML, CSS и JavaScript, а также сокращение передаваемых потоков мультимедийных данных по протоколам HTTP и WebSocket между клиентом и сервером, упакованных в формат JSON, за счет пиринговой передачи данных между клиентскими приложениями в процессе сеанса связи.

4. Методика тестирования пиринговых систем видеоконференцсвязи, включающая алгоритмы функционального тестирования, набор тестов, оценивающих потребляемые ресурсы и ручное тестирование, позволившая в ходе тестирования разработанного программного обеспечения улучшить эргономику пользовательского интерфейса выявить отклонения и нарушения в работе системы, подтвердить стабильную работу сервера и корректное использование памяти, а также провести сравнение с наиболее популярной системой видеоконференцсвязи «Skype», которое показало, что разработанная система потребляет меньшее количество оперативной памяти на всех этапах работы.

Дальнейшим перспективным направлением исследования является интеграция системы видеоконференцсвязи и интеллектуальных методов распознавания звуковых и графических образов, биометрической идентификации, цифровых программируемых технологий инфокоммуникационных систем, позволяющих существенно снизить объем передаваемых данных по пиринговым связям для распределения нагрузки на вычислительные и сетевые встроенные ресурсы мобильных гетерогенных клиентских устройств.

Полученные результаты соответствуют п. 3 «Модели, методы, алгоритмы, языки и программные инструменты для организации взаимодействия программ и программных систем» п. 7 «Человеко-машинные интерфейсы, модели, методы, алгоритмы и программные средства машинной графики, визуализации, обработки изображений, систем виртуальной реальности, мультимедийного общения» и п. 8 «Модели и методы создания программ и программных средств для параллельной и распределенной обработки данных, языки и инструментальные средства параллельного программирования» паспорта специальности 05.13.11 – «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей».

ОСНОВНЫЕ ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

В рецензируемых журналах из списка ВАК:

1. Савельев, А.И. Алгоритмы обработки данных в контролируемых аккаунтах системы видеоконференцсвязи / А.И. Савельев // Информационно-управляющие системы. — 2016. — № 3. — С 15-23.
2. Савельев, А.И. Модели кодирования полимодальной информации / О.О. Басов, И.С. Кипяткова, И.А. Саитов, А.И. Савельев // Информационно-управляющие системы. — 2016. — № 2. — С. 68–73.
3. Савельев, А.И. Межличностные аспекты полимодальности при построении коммуникационных систем / О.О. Басов, Д.А. Щербаков, А.И. Савельев, А.Л. Ронжин // Пилотируемые полеты в космос. — 2015. — № 4(17). — С. 28–48.
4. Савельев, А.И. Архитектура обмена данными без потерь в пиринговом веб-приложении видеоконференцсвязи / А.И. Савельев, М.В. Прищепа // Доклады Томского государственного университета систем управления и радиоэлектроники. — 2014. — № 2(32). — С. 238–245.
5. Савельев, А.И. Архитектура и программные средства системы сопровождения распределенных мероприятий / В.Ю. Будков, М.В. Прищепа, Ал.Л. Ронжин, А.И. Савельев // Научный вестник НГТУ. — 2014. — № 3(56). — С. 96–107.
6. Савельев, А.И. Оптимизация алгоритмов распределения потоков мультимедийных данных между сервером и клиентом в приложениях видеоконференцсвязи / А.И. Савельев // Труды СПИИРАН. — 2013. — № 8(31). — С. 61–79.

В зарубежных изданиях, индексируемых в WoS/Scopus:

7. Saveliev, A.I. Algorithms and Software Tools for Distribution of Multimedia Data Streams in Client Server Videoconferencing Applications / A.I. Saveliev, A.L. Ronzhin // Pattern Recognition and Image Analysis. — Springer, 2015. — vol. 25. — no. 3. — pp. 517–525.
8. Saveliev, A. Bimodal Speech Recognition Fusing Audio-Visual Modalities / A. Karpov, Al. Ronzhin, I. Kipyatkova, A. Ronzhin, V. Verkhodanova, A. Saveliev, M. Zelezny // In Proc. HCI-2016, Toronto, Canada, Springer LNCS. — 2016 — vol. 9732 — pp. 170–179.
9. Saveliev, A.I. Algorithms for Low Bit-Rate Coding with Adaptation to Statistical Characteristics of Speech Signal / O.O. Basov, A.L. Ronzhin, A.I. Saveliev, Al.L. Ronzhin // Speech and Computer. Springer, Proc. SPECOM-2015, LNAI 9319. — 2015. — pp. 65–72.
10. Saveliev, A.I. Architecture of data exchange with minimal client-server interaction at multipoint video conferencing / A.I. Saveliev, I.V. Vatamaniuk, An.L. Ronzhin // NEW2AN/ruSMART 2014, LNCS 8638. — 2014. — pp. 164–176.
11. Saveliev A.I. Elimination of distorted images using the blur estimation at the automatic registration of meeting participants / I.V. Vatamaniuk, Al.L. Ronzhin, A.I. Saveliev, An.L. Ronzhin // NEW2AN/ruSMART 2014, LNCS 8638. — 2014. — pp. 133–143.
12. Saveliev A.I. Optimization of multimedia server and client streams distribution in videoconference applications / A.I. Saveliev, An.L. Ronzhin // In Proc. of 11-th International Conference on Pattern Recognition and Image Analysis: New Information technologies, PRIA-2013. — 2013. vol. 2. — pp. 463–466.
13. Saveliev, A.I. . Context-Aware Mobile Applications for Communication in Intelligent Environment / V.Yu Budkov, A.I Saveliev, A.L. Ronzhin // Springer-Verlag Berlin Heidelberg, NEW2AN/ruSMART 2012, LNCS 7469. — 2012. — pp. 307–315.
14. Saveliev, A.I. Conceptual Model of Cyberphysical Environment Based on Collaborative Work of Distributed Means and Mobile Robots / Anton Saveliev, Andrey Ronzhin, Oleg Basov, Sergey Solyonyj // Springer, ICR 2016, LNAI 9812. — 2016. — pp. 32–39.
15. Saveliev, A.I. Scenarios of Multimodal Information Navigation Services for Users in Cyberphysical Environment / Irina Vatamaniuk, Anton Saveliev, Dmitriy Levonevskiy, Alexander Denisov // Springer, SPECOM 2016, LNAI 9811. — 2016. — pp. 588–595.

Автореферат диссертации

САВЕЛЬЕВ
Антон Игоревич

АРХИТЕКТУРЫ, АЛГОРИТМЫ И ПРОГРАММНЫЕ СРЕДСТВА ОБРАБОТКИ ПОТОКОВ
МНОГОМОДАЛЬНЫХ ДАННЫХ В ПИРИНГОВЫХ ВЕБ-ПРИЛОЖЕНИЯХ
ВИДЕОКОНФЕРЕНЦСВЯЗИ

Текст автореферата размещен на сайтах:

Высшей аттестационной комиссии Министерства образования и науки Российской
Федерации

<http://vak2.ed.gov.ru/catalogue>

Федерального государственного бюджетного учреждения науки Санкт-Петербургского
института информатики и автоматизации Российской академии наук (СПИИРАН)

<http://www.spiiras.nw.ru/DissSovet/Templates/PhDSchedule.htm>

Подписано в печать __.__.2016г.

Формат 60x84 1/16. Бумага офсетная. Печать офсетная.

Усл.печ.л. 1,0. Тираж 100 экз.

Заказ № 3101

Отпечатано в ООО «Издательство «ЛЕМА»»

199004, Россия, Санкт-Петербург, В.О., Средний пр., д. 24

тел.: 323-30-50, тел./факс: 323-67-74

e-mail: izd_lemma@mail.ru

<http://www.lemaprint.ru>