

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий



ВЫПУСКНАЯ РАБОТА БАКАЛАВРА

Тема: Модуль SIP-телефонии для веб-браузера

Студент гр. 43501/4 В.С. Филиппов

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

Работа допущена к защите

зав. кафедрой

_____ В.М. Ицыксон

«_____» _____ 2016 г.

ВЫПУСКНАЯ РАБОТА БАКАЛАВРА

Тема: Модуль SIP-телефонии для веб-браузера

Направление: 230100 – Информатика и вычислительная техника

Выполнил студент гр. 43501/4

_____ В.С. Филиппов

Научный руководитель,

ст. преп.

_____ А.В. Зозуля

Эта страница специально оставлена пустой.

РЕФЕРАТ

Отчет, 21 стр., 4 ил.

СОФТФОН, IP-ТЕЛЕФОНИЯ, WEB-ПРИЛОЖЕНИЕ

Бакалаврская работа посвящена осуществлению телефонных звонков из браузера. Рассмотрены существующие решения в данной области. Сформулированы требования к программному модулю, осуществляющего звонки. TODO

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	7
1. ОБЗОР СУЩЕСТВУЮЩИХ ПОДХОДОВ РЕАЛИЗАЦИИ SIP-ТЕЛЕФОНИИ ДЛЯ WEB-БРАУЗЕРА	8
1.1. Подход к реализации телефонии для web-браузера на Java	8
1.2. Подход к реализации телефонии для web-браузера на Flash	10
1.3. Подход к реализации телефонии для web-браузера на WebRTC	12
2. ПОСТАНОВКА ЗАДАЧИ, ВЫБОР СПОСОБА РЕШЕНИЯ	15
3. ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ МОДУЛЯ	17
4. РАЗРАБОТКА	18
5. ТЕСТИРОВАНИЕ, АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ	19
ЗАКЛЮЧЕНИЕ	20
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ . . .	21

СПИСОК ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ

AEC	Acoustic Echo Cancellation, эхоподавление
AES	Advanced Encryption Standard, один из симметричных алгоритмов блочного шифрования
AGC	Automatic Gain Control, автоматическая регулировка усиления
AJB	Adaptive Jitter Buffer, буфер выравнивания задержек передачи
AS3	Action Script 3
DTLS	Datagram Transport Layer Security, протокол датаграмм безопасности транспортного уровня
FCS	Flash Communication Server
JRE	Java Runtime Environment
RTC	Real-Time Communications, коммуникации в реальном времени
SRTP	Secure Real-time Transport Protocol, безопасный протокол передачи данных в реальном времени
VoIP	Voice over IP, IP-телефония
ПО	программное обеспечение

ВВЕДЕНИЕ

В последнее время очень популярными стали web-приложения, например социальные сети, игры, онлайн-редакторы (документов, изображений и видео), прямые видео-трансляции и многие-многие другие.

Около 15 лет назад клиент просматривал web-страницы, только переходя с одной на другую. Примерно в 2005 году, появился способ сделать страницы динамичными с помощью AJAX. С тех пор, почти весь обмен по HTTP инициировался клиентом разными способами, например каким-нибудь действием, или периодическим опросом сервера на получение новых данных. Однако при таком обмене появляется задержка на установление HTTP-соединения каждый раз при получении новых данных от сервера. Это создавало проблемы для создания web-приложений реального времени. (<http://www.html5rocks.com/en/tutorials/websockets/basics/>)

Около 5 лет назад появилась новая технология, которая позволила обмениваться двум сторонам асинхронно и симметрично. Это полнодуплексный протокол WebSocket, который работает поверх TCP. Уже в 2009 году вышла первая версия браузера, поддерживающая стандарт. (<http://blog.chromium.org/2009/12/web-sockets-now-available-in-google.html>)
//TODO

1. ОБЗОР СУЩЕСТВУЮЩИХ ПОДХОДОВ РЕАЛИЗАЦИИ SIP-ТЕЛЕФОНИИ ДЛЯ WEB-БРАУЗЕРА

Для разработки модуля телефонии для web-браузера необходимо сначала проанализировать существующие способы. Рассмотрим их в хронологическом порядке.

1.1. Подход к реализации телефонии для web-браузера на Java

Временем появления телефонии для браузера можно считать момент, когда в Java апплетах появилась поддержка захвата аудио с микрофона. JRE широко распространена и обычно уже установлена в Windows и Linux системах.[1] Java код выполняется на JRE установленной на компьютере или в расширении браузера, захватывает аудио с микрофона и отправляет его на сервер по протоколу RTP. Такой апплет должен быть подписан, и при его запуске пользователя спросят, желает ли он запустить подписанный апплет от данного производителя, который имеет доступ к функциям сетевого обмена, доступ к микрофону и т.п.

Преимущества данного подхода:

1. поддерживается большинством браузеров
2. возможность прямого взаимодействия с сервером по RTP

3. доступность JRE для конечного пользователя

К сожалению, в Java есть проблемы с обработкой звука в реальном времени. А это почти всё алгоритмы, которые должны быть у каждого VoIP-телефона: АЕС, АРС, АЖВ и Noise suppression (подавление шума).

Эхоподавление позволяет использовать динамики так, чтобы собеседник не слышал собственных слов, которые предаются обратно с динамиков на микрофон. АРС регулирует громкость так, чтобы не было слишком тихо или слишком громко. АЖВ устраняет большую задержку в передаче и "choppy audio"— прерывистый неразборчивый звук.

Все эти алгоритмы теоретически можно реализовать на Java, но это проблемно. Во-первых, реализовать универсальные и производительные алгоритмы (например, АЕС) достаточно сложно. Во-вторых, реализация таких алгоритмов на Java может работать в несколько раз медленнее, чем на C/C++, а это может сказаться с большим расходом ресурсов клиентского CPU.

Производители Java апплетов с функцией звонков реализуют собственные обработчики звука или используют уже существующие решения на C/C++. Они используют в апплете библиотеки, которые берут на себя обработку вышеописанных алгоритмов. В результате Java апплет имеет стандартные VoIP функции для обеспечения качественного звонка со всеми VoIP алгоритмами.

Таким образом, подход к реализации VoIP-телефонии на Java имеет два недостатка:

1. сложность реализации алгоритмов обработки звука для каждой платформы
2. отсутствие кроссплатформенности - алгоритмы обработки звука должны быть реализованы на всех платформах, или используемые библиотеки должны быть кроссплатформенными
3. необходимо устанавливать JRE

Довести DSP до отличного качества или купить соответствующие разработки может позволить себе не каждый вендор. То же касается поддержки различных кодеков для аудио и видео.

1.2. Подход к реализации телефонии для web-браузера на Flash

Начиная с 6 версии Flash Player умел взаимодействовать с FCS MX 1.0 и обмениваться с сервером потоками аудио данных. Он умел захватывать аудио и кодировать его с помощью кодека NellyMoser, и видео и кодировать его с помощью кодека Sorenson Spark. В качестве транспорта для аудио и видео в Flash Player 6 использовался протокол RTMP, который сегодня имеет открытую спецификацию, опубликованную Adobe. До полноценной VoIP-телефонии тогда было еще очень далеко. Но платформа делала свое дело и передавала звук и видео от одного плеера к другому через сервер.

Однако в связке Flash Player 6 + FCS MX 1.0 была задержка звука, она также осталась в следующих версиях сервера, включая последнюю Adobe Media Server. Причина в том, что RTMP протокол

работает поверх TCP, а потому не приспособлен для полноценного VoIP. Для приложений реального времени лучше использовать UDP.

Проблему с UDP в Flash Player решили в 10 версии: ввели поддержку нового протокола RTMFP и функцию AEC. В 11 версии Flash Player добавили поддержку кодеков G.711 и H.264. В AS3 API так же имеются AJB для кодеков G.711 и Speex.

Итак, VoIP алгоритмы, которые поддерживает Flash Player 11: AEC, AJB, AES шифрование. Шифрование AES защищает трафик между браузером и сервером от посторонних.

Но у Flash Player есть небольшая проблема. В документации Adobe AS3 сказано, что RTMFP поддерживает три режима: надежная доставка, частично-надежная доставка, ненадежная доставка. Но есть только два флага для аудио и видео которые принимают либо "true" либо "false". "False" описывается как режим частичной доставки. В итоге, получается, что ненадежную доставку включить не удаётся, а при передаче звука она наиболее важна. Частичная доставка – это TCP ретрансмиты, которые происходят очень ограниченное время, но этого хватает, чтобы испортить звук в нестабильной сети. Такие ретрансмиты вызывают дрожание, которые портит поток. AJB на принимающей стороне не может справиться с таким большим разбросом. Решением может оказаться добавление ненадёжной доставки на уровне протокола на серверной стороне.

Таким образом, у подхода к реализации VoIP-телефонии на Flash есть следующие преимущества:

1. поддерживается большинством браузеров

2. привычная технология для разработчиков – AS3
3. качественная передача аудио и видео

Однако имеются и недостатки:

1. требует промежуточного сервера (не поддерживает открытые UDP протоколы, такие как RTP/SRTP)
2. отсутствие AGC
3. необходимо устанавливать Flash Player

1.3. Подход к реализации телефонии для web-браузера на WebRTC

WebRTC - проект с открытым исходным кодом, предназначенный для организации передачи потоковых данных между браузером или другими поддерживающими его приложениями по технологии точка-точка.[2]

Технология WebRTC имеет продуманную архитектуру, избавленную от ошибок и недостатков, выявленных в плагинах браузера, которые существовали до неё. Технологические возможности WebRTC: SRTP, DTLS, ICE, STUN, AEC, AGC, AJB, Opus, VP8.

//TODO nomenclature ICE STUN Opus VP8 SDK

Набор используемых в WebRTC технологий больше похож на VoIP SDK. SRTP и DTLS обеспечивает защиту трафика между WebRTC узлами. ICE и STUN помогают преодолеть NAT.[3] AEC, AGC и AJB работают для того чтобы сделать аудио и видео качественным – без лагов и задержек. Кодеки Opus и VP8 хорошо подходят для

глобального Интернета, где скорость соединения может неожиданно падать.

Однако надо отметить, что подходы к реализации VoIP-телефонии в браузере, рассмотренные ранее (Java и Flash) требуют дополнительной установки ПО. WebRTC — это единственная технология, которая является родной для браузера.

Преимущества технологии WebRTC:

1. все алгоритмы обработки звука
2. технология встроена в браузер
3. совместимость с традиционными VoIP
4. реализован на популярном среди web-разработчиков языке JavaScript

Недостатки технологии WebRTC:

1. RFC ещё не разработан, на сегодняшний день существует черновик[4]
2. Поддерживается не всеми браузерами
3. Необходима поддержка со стороны сервера (имеется поддержка известными Asterisk, FreeSWITCH)

Как мы видим преимуществ у подхода для разработки модуля SIP-телефонии на основе технологии WebRTC больше. Недостатки же в ближайшее время будут устраняться.

//TODO В браузере имеется возможность осуществлять обмен не по HTTP-протоколу, а с помощью WebSockets. Уже в 2009 году

(<http://blog.chromium.org/2009/12/web-sockets-now-available-in-google.html>)

они стали доступны

2. ПОСТАНОВКА ЗАДАЧИ, ВЫБОР СПОСОБА РЕШЕНИЯ

В современном мире коммуникации через интернет становятся очень удобными, такие коммуникации экономят время. Такие коммуникации особенно будут экономить время, если их встраивать в CRM-системы. Поэтому задача будет следующая. Написать модуль SIP-телефона для веб-браузера, поддерживаемого как можно большим количеством браузеров, и легко в будущем встраиваемый в любой сайт, например, CRM-систему. В данной работе ограничимся только передачей аудио потока.

Для реализации данной задачи, будем использовать технологию WebRTC. Существуют две реализации технологии WebRTC по протоколу SIP на JavaScript. Это библиотеки sipML5 и JsSIP.

Преимущества sipML5:

1. наличие документации
2. поддержка трансфера звонка

Недостатки sipML5:

1. большой размер 1 Mb
2. документация не очень подробная

Преимущества JsSIP:

1. легковесная (130kb)

2. подробная документация

Недостатки JsSIP:

1. нет поддержки трансфера звонка
2. необходима установка NodeJS

3. ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ МОДУЛЯ

Модуль будет состоять из нескольких файлов:

1. `module.js` - подгружаемый файл
2. `module.html` - главный файл
3. `corePhone.js` - файл, обрабатывающий события звонков
4. `phonePopupUI.js` - файл обработки перемещения плавающего окна (графический интерфейс)
5. `phonePopupUI.css` - файл стилей плавающего окна (графический интерфейс)

4. РАЗРАБОТКА

5. ТЕСТИРОВАНИЕ, АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. WebRTC, Flash RTMFP, Java Applet – три ведущих технологии для браузерных VoIP звонков [Электронный ресурс], CNews Клуб. — URL: http://club.cnews.ru/blogs/entry/webrtc_flash_java_ (дата обращения: 16.06.2016).
2. WebRTCElectronic resource, WebRTC. — URL: <https://sites.google.com/site/webrtc/home> (online; accessed: 16.06.2016).
3. Всё, что вы хотели знать о протоколе SIP [Электронный ресурс], Системный администратор. — URL: <http://samag.ru/archive/article/2017> (дата обращения: 16.06.2016).
4. Bergkvist Adam, Burnett Daniel C., Jennings Cullen et al. WebRTC 1.0: Real-time Communication Between BrowsersElectronic resource // Internet Requests for Comments, W3C. — 2016. — May. — URL: <https://www.w3.org/TR/2016/WD-webrtc-20160531/> (online; accessed: 16.06.2016).