

Министерство образования Республики Беларусь  
Учреждение образования  
Белорусский государственный университет  
Информатики и радиоэлектроники

УДК 62010

Рымарчик  
Иван, Сергеевич

Система push-нотификаций

**АВТОРЕФЕРАТ**

на соискание степени магистра информатики и вычислительной техники  
по специальности 1-40 81 02  
«Интеллектуальные вычислительные комплексы, системы и компьютерные  
сети»

---

Научный руководитель  
Иванов Николай Николаевич  
доцент, кандидат физико-математических наук

---

МИНСК 2016

## КРАТКОЕ ВВЕДЕНИЕ

В данной магистерской диссертации проектируется и разрабатывается система push-уведомлений для передачи сообщений на зарегистрированные в системе абонентские устройства. Для поддержки нового устройства достаточно включить в прошивку устройства агента разрабатываемой системы.

Использование классического pull-подхода на современных абонентских устройствах малоэффективно, так как на неактуальные для абонента запросы растрачиваются системные и сетевые ресурсы как абонентских устройств, так и серверов приложений. Именно уменьшение потребления ресурсов является целью разработанной системы push-уведомлений. Поэтому тема диссертации актуальна.

В связи с этим целью диссертационной работы является проектирование и разработка системы push-уведомлений. Для достижения поставленной цели решались следующие задачи:

1. Изучение архитектуры и функционирования существующих аналогов систем push-уведомлений.
2. Проектирование системы push-уведомлений, выбор технологий разработки и стеков протоколов связи между компонентами системы.
3. Разработка программного обеспечения системы.

В ходе разработки магистерской диссертации рассматриваются современные аналоги, представленные на рынке:

- система Apple Push Notification Service от компании Apple, которая функционирует под управлением операционной системы iOS;
- система Google Cloud Messaging от компании Google, которая работает в операционной системе Android.

Проводится сравнительный анализ этих аналогов между собой и сравнение с нашей системой. В диссертации получены следующие результаты: разработана архитектура системы push-уведомлений, выбраны протоколы и способы взаимодействия абонентских устройств с системой push-уведомлений, а также формат сообщений. В завершении записки проводится анализ достоинств и недостатков предложенной схемы функционирования системы, а также намечены цели дальнейших исследований и возможные модификации системы.

Представленная к защите система push-уведомлений разработана под абонентское устройство с операционной системой Linux. Инструмент разработки – язык программирования C++ и открытые кроссплатформенные библиотеки boost, http-parser, libcurl.

Серверная часть разрабатывалась под архитектуру компьютеров x86, использовался язык программирования C++. Разработка программного

обеспечения серверной части имела целью использование небольшого объема ресурсов на каждое соединение, это обусловлено потенциально большим, порядка сотен тысяч, подключений абонентских устройств к одному серверу.

Для связи абонентского устройства и системы используется постоянное TCP соединение с нестандартными параметрами, которые задаются самой системой push-уведомлений при установлении соединения.

Во введении обоснована актуальность темы, сформулирована цель работы и перечислены задачи, вытекающие из поставленной цели.

В первой главе рассмотрены существующие аналоги, их архитектура, дан краткий сравнительный анализ.

Во второй главе предложен детальный проект системы с обоснованием выбора протоколов и решаются вопросы согласования компонент системы.

В третьей главе обоснован выбор средств для разработки системы, выбран инструмент разработки, то есть язык программирования и необходимые вспомогательные библиотеки. Кратко рассмотрен вопрос тестирования системы.

В заключении диссертации изложены возможные области применения проекта, описана конкретное приложение в бизнесе и обсуждаются дальнейшие направления развития темы.

## ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Объектом исследования данной магистерской работы является система для передачи push-уведомлений на абонентские устройства.

Программное обеспечение предназначена для встраивания в абонентские устройства и позволяют передавать сторонним приложениям данные на них по запросу, при этом самим приложениям не нужно поддерживать соединение со своими серверами. Данный подход позволяет уменьшить сетевые и системные нагрузки на конечные устройства и на сервера приложений в случаях, когда между приложениями и серверами не нужна частая передача данных.

Предмет исследования – клиентская часть системы и один из компонентов серверной системы, с которым общаются агенты (клиентские части).

Цель работы – разработка масштабируемой системы, способной одновременно работать с большим количеством устройств, которая облегчает написание приложений, которым необходима связь сервера с клиентом.

Система состоит из серверной и клиентской частей. Серверная часть работает на серверах с архитектурой x86/x64. Клиентская часть должна быть кроссплатформенной, так как она должна работать на множестве различных архитектур, которые включают в себя семейства архитектур ARM, MIPS и x86.

Система состоит из следующих компонент:

1. Сервис связи с конечными устройствами. Данный сервис может состоять из множества удаленных серверов, которые не связаны между собой. Агент подключается к системе, используя именно этот сервис. В ходе данной работы сервер этой системы называется брокером. Состоит из одного главного сервера, который знает обо всех запущенных экземплярах данного сервиса, и множества экземпляры, которые непосредственно поддерживают соединение с агентами.

2. Сервис регистрации. Данный сервис ответственен за обработку запросов на регистрацию от агентов и добавлению новых агентов в базу данных.

3. Сервис сообщений. Данный сервис отвечает за верификацию запросов от сторонних приложений. В случае успешной верификации запроса, данный сервис формирует бинарное сообщение в определенном формате, которое он передает агенту через сервис связи.

4. База данных. Данный сервис является хранилищем для всей информации, которая необходима для функционирования системой push-уведомлений. База данных не имеет выхода в сеть Internet, и доступна только во внутренней сети системы.

5. Клиентская часть. Данная часть системы будет работать на абонентских устройствах и будет включена в прошивку устройств.

Агенты поддерживают соединение с серверной частью, адрес которой может динамически меняться в зависимости от текущих правил распределения нагрузки.

Результаты работы изложены в статье к V международной заочной научно-технической конференции ITRT-2015.

Библиотека БГУИР

## КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

В начале работы рассмотрены существующие аналоги систем push-уведомлений от компаний Apple, Google и Amazon. Рассматривается их архитектура, рассматривается порядок работы при отправке уведомлений, используя данные системы. Приводятся наиболее важные функциональные параметры систем. Приводится сравнение со стандартной схемой работы приложения в случае использования pull-запросов, приводятся плюсы и минусы при использовании push-системы.

Во второй главе рассмотрены вопросы создания архитектуры системы, разбиение системы push-уведомлений на отдельные компоненты. Рассмотрены роли и обязанности каждого компонента, выбраны и описаны протоколы взаимодействия между компонентами. Приведен пример работы и взаимодействия компонентов при отправке сторонним приложениям push-уведомления. Проанализированы функциональные требования к некоторым компонентам системы, в частности к брокеру и к агенту.

В третьей главе рассматривается выбор языка программирования, стека технологий для разработки программного обеспечения системы. Рассматривается выбор кроссплатформенных библиотек для разработки брокера и агента. Приводится методика тестирования, рассматривается тестирование отдельных модулей как в процессе разработки, так и при окончательной проверке. Рассматриваются средства и программное обеспечение, которые использовались для тестирования. Рассматривается средство мониторинга для получения данных о работе системы.

## ЗАКЛЮЧЕНИЕ

В процессе работы над магистерской диссертацией были рассмотрены основные принципы работы систем push-уведомлений, приведены преимущества систем push-уведомлений над pull-системами. Были рассмотрены сценарии работы, при которых использование push-уведомлений имеет смысл. Были рассмотрена архитектура и функционирование существующих на рынке аналогов Apple Push Notification System, Google Cloud Messaging и Amazon Simple Notification Service. Была спроектирована система push-уведомлений, был выбран стек технологий для разработки (C++ и библиотеки boost, libcurl). Были разработаны протоколы общения не только между системой и абонентским устройством (агентом), но также между компонентами самой системы.

В результате диссертации был создан комплекс программных средств, состоящий из брокеров и агентов, которые позволяют использовать преимущества систем push-уведомлений. Благодаря данной разработке, у пользователей появится возможность для работы с приложениями на своих устройствах через сеть интернет, при этом разработчикам системы не надо заботиться о наличии постоянного соединения с серверами.

Разработанная система имеет следующие достоинства:

- минимальное количество необходимых ресурсов для работы системы на устройстве;
- возможность запускать клиентов и передавать им данные, даже если они не находятся в данный момент в памяти (не запущены);
- возможность передавать данные клиентам не только на локальной машине, но также в локальной сети – таким образом, появляется возможность использовать устройство в системе в качестве шлюза;
- кроссплатформенный агент – есть возможность запуска практически на любой системе, для которой есть компилятор языка C++;
- легковесный внешний интерфейс для работы с системой;
- в отличие от аналогов, размер максимального уведомления составляет  $2^{32}$  байт (у рассмотренных аналогов максимум 4Кб);
- наличие уведомлений об изменении текущего состояния агента (такой возможности нету у аналогов). Наличие такого типа уведомлений позволяет сторонним приложениям менять своё состояние в зависимости от текущего статуса (например, если видеочасть использует разрабатываемую систему, то она может менять цвет светодиодов).

В качестве недостатков разработанной системы можно отметить следующие пункты:

- необходимость иметь готовую конфигурацию на устройстве в момент запуска;
- отсутствие механизма добавления новых сторонних приложений в систему – то есть использовать её возможности могут только разработчики, у которых есть доступ к основной базе данных и к прошивкам устройств;
- необходимость запустить клиентское приложение как минимум 1 раз для того, что серверная часть клиентского приложения знала уникальный идентификатор устройства.

Первый недостаток возможно устранить. Для этого необходимо расширить протокол между системой и агентом системным типом уведомлений, то есть чтобы такие уведомления обрабатывал сам агент. Тогда он сможет по запросу системы добавлять какие-либо настройки (тем самым расширяя конфигурацию). Второй способ решения проблемы – сделать специального клиента, который будет добавлять конфигурацию (но это может повлечь некоторые проблемы безопасности).

Второй недостаток не рассматривался в рамках данной работы, поскольку использование системы планируется в рамках одной компании с собственными устройствами, поэтому вопрос добавления новых приложений для данных условий не стоял.

Третий недостаток является следствием используемой архитектуры. Его можно частично решить, добавив в обработчики уведомлений изменения состояния от агента клиента, чтобы он мог сразу после регистрации (если им было получено уведомление о регистрации) послать данные на свой сервер. Проблема тут возникает следующего характера – в текущем приложении используется только один способ для передачи всех типов уведомлений (и регистрация, и подключение/отключение, что может вызвать ненужный запуск клиента в случае установления соединения/потери связи. Решаются проблемы с ненужными запусками только разделением каналов передачи уведомлений (то есть дополнительными настройками) о регистрации и о статусе подключения.

Разработанная система push-уведомлений используется в устройствах фирмы NETGEAR (например, маршрутизатор R7000). Агент системы включен в прошивку устройств и используется для работы некоторых приложений, таких как получение информации об устройстве, приложении для работы с файлами на USB накопителях, удаленной настройки параметров маршрутизатора.

Среди возможных улучшений и дальнейшей работы над системой можно выделить добавление возможности динамически добавлять новые приложения в систему, используя специально API. Также нужно рассмотреть возможность добавления новых протоколов для передачи уведомлений системы (например, XMPP, как это сделано у GCM).



Библиотека БГУИР

### Список публикаций соискателя

[1–А] Рымарчик, И.С. Система push-уведомлений / И. С. Рымарчик // 51 конференция магистрантов БГУИР: тезисы докл. – Минск, 2015 – С. 94 – 95.

[2–А] Рымарчик, И.С. Система push-уведомлений / И. С. Рымарчик // V международная заочная научно-технической конференция ITRT-2015: сборник статей – Тольятти, 2015 – С. 201 – 204.

Библиотека БГУИР