

Решения JRRA для P2P систем

Андрей Пептюк

1. Предисловие

Jrra — это open source проект, направленный на популяризацию технологий P2P и создание средств разработки для кросс-платформенных, масштабируемых, гибких приложений.

2. Введение

RockyRoad (<http://www.jrra.org>) — это набор открытых масштабируемых API и их реализация на платформах J2SE and J2ME. Этот протокол был изначально задуман для того, чтобы позволить разработчикам создавать системы, способные работать как на мобильных беспроводных устройствах, так и на стандартных ПК.

RockyRoad был задуман для подключения мобильных устройств к P2P системам, и в особенности для разработчиков сервисов, основанных на физическом положении пользователя (location-specific services). Многие P2P платформы и протоколы — Jabber, Gnutella, ICQ и другие — хорошо работают по стабильным каналам, но разработчикам не всегда удается сохранить их достоинства при работе с мобильными устройствами.

Из-за растущей популярности языка Java и его удобства для создания кросс-платформенных решений участники open source проекта JRRA выбрали для реализации протокола платформу Java. RockyRoad реализованна Java 2 Standard Edition и Java 2 Micro Edition (Connected Limited Device Configuration). Это даст возможность устанавливать систему, использующую RR, как на ПК, так и на PDA устройствах и Java-совместимых сотовых телефонах. В настоящее время RockyRoad использует потоки байтов для передачи данных, но ведется работа по переходу на XML-пакеты. Это также обеспечит способность к взаимодействию — любая система, поддерживающая XML, сможет обрабатывать пакеты RockyRoad при наличии

соответствующего DTD.

Разные устройства, разные платформы, разные протоколы — минимум усилий. Протокол легко масштабируется — новые виды транспорта или форматы пакетов могут быть добавлены без переработки кода или изменения модели. А встроенная поддержка сервисов, основанных на физическом положении пользователя, обеспечит P2P место в мире повсеместно проникающих сетей.

RockyRoad — пакетно-ориентированный, а не потоко-ориентированный протокол. Такой подход обусловлен спецификой коммуникационных процессов на беспроводных устройствах. Протокол поддерживает различные транспортные механизмы, и сам может рассматриваться как транспортный механизм высокого уровня с точки зрения программиста, использующего API. Данные могут доставляться с помощью TCP или UDP в IP-сетях, SMS/USSD в сетях GSM или GRPS в сетях GSM или TDMA.

RockyRoad состоит из двух частей. Первая — это его транспортная часть. В этой части мы реализуем разные транспортные протоколы, наборы транспортных пакетов, механизмы сжатия и разбиения пакетов при отправке, динамический выбор транспорта и многое др. Вторая часть — сервисы (JRRR Services). Это подсистемы, которые являются необязательными либо имеют разную реализацию для всех прикладных систем, созданных на основе JRRR. Наиболее важными для P2P систем являются сервисы для поиска и индексации реер и ресурсов в сети, а также сервисы, обеспечивающие безопасную передачу данных. RockyRoad также обеспечивает другие категории сервисов, включая сервисы доступа к данным (на платформах с ограниченными ресурсами) и сервисы безопасности.

3. Архитектура

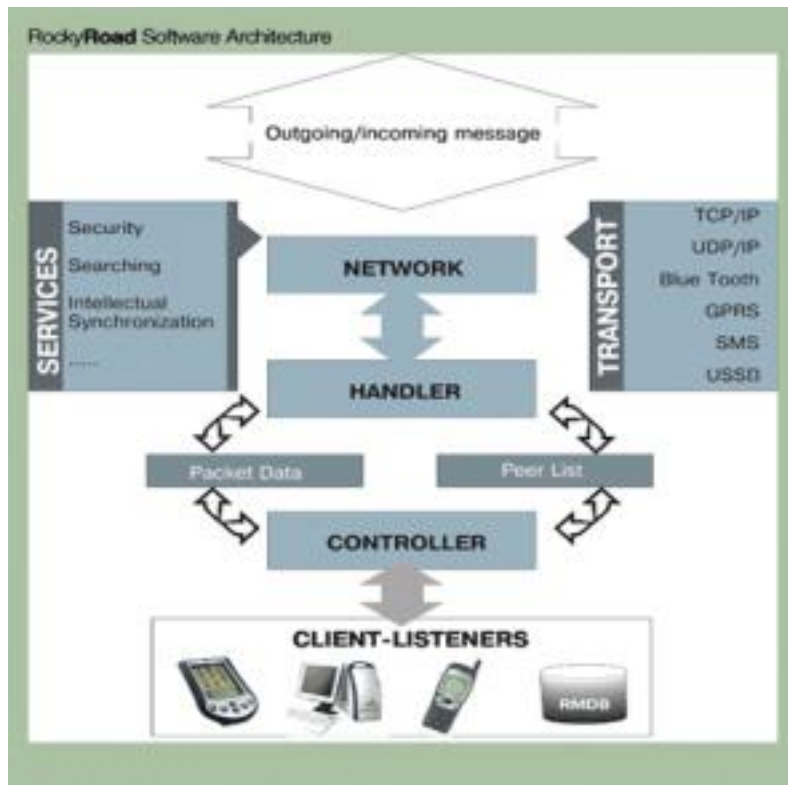


Рисунок 1. Архитектура системы.

На рисунке 1 показана общая архитектура RockyRoad. Ниже приводится очень упрощенное описание протокола.

Получение пакета из сети происходит с помощью класса Network, который рассчитан для работы с произвольным сетевым протоколом (TCP, UDP, SMS). Далее полученная последовательность байт передается классу Handler, который занимается сборкой сообщения, если оно было фрагментировано, его распаковкой, декодированием. Затем сам пакет и Peer его отправителя передаются классу Controller, который оповещает о пришедшем пакете пользовательские Listener'ы.

Отправка пакета происходит следующим образом: пользователь вызывает один из методов класса Controller и передает ему пакет для передачи и список Peer'ов (Peer), которым его отсылать. Далее пакет передается классу Handler, где он сериализуется и при необходимости разбивается, пакуется, кодируется и т.д. Затем уже готовый сериализованный поток передается классу Network для его отсылки по нужному

сетевому протоколу.

Более подробное описание доступно по адресу <http://www.jrra.org/bin/view/Main/Architecture>

4. Сервисы

Сервисами мы называем ПО, которое не вписывается в наше виденье PURE P2P технологий, и которое не будет обязательным для всех прикладных систем либо его реализация для разных систем будет разной. Общим признаком сервисов будет наличие привилегированного реер, имеющего специальные полномочия, отличающиеся от возможностей остальных реер.

Все сервисы протокола RR можно разделить на несколько групп. Каждая группа может иметь любое количество реализаций, и лишь разработчики прикладных систем на основе RR могут сделать выбор в пользу той или иной реализации.

4.1. Сервисы, основанные на физическом положении пользователя

Это новая идея в P2P системах, где мобильные приложения могут использовать информацию о физическом положении пользователя по отношению к другим людям и устройствам. Этот подход используется в таких системах, как Cricket [3] и WebSign [4]. Эти сервисы для мобильных устройств могут быть расширены с помощью P2P систем. API RockyRoad — первый интерфейс, предлагающий подобные функции разработчикам P2P приложений.

4.2. Сервисы поиска и индексации

Основной задачей этой группы сервисов является поиск нужных peers или ресурсов в сети, их индексация и построение иерархии участников и ресурсов на основе персональной информации от каждого участника сети. P2P системы условно можно разделить на два типа: реер-ориентированные и ресурсоориентированные системы.

Ярким примером р2р систем первого типа являются системы мгновенной передачи сообщений. Пользователи этих систем организуют соединения внутри сети между

Решения JRRR для P2P систем

интересующими их участниками. При выборе реер для взаимодействия пользователи системы исходят не из того, каким ресурсом (мультимедийные файлы, базы данных и т.п.) обладает на данный момент пользователь, а из того, кем является этот реер. Такие системы используются для проведения регулярных переговоров/консультаций, обсуждения текущих вопросов в интерактивном режиме.

Организация процесса поиска в сети для P2P систем первого типа достаточно проста. Типичным решением является наличие привилегированного реер, который содержит информацию обо всех участниках сети, их текущее состояние, способы доступа к реер и их текущие адреса в сети. При этом функции привилегированного реер могут переходить от одного реер к другому или быть распределенными по какому-либо принципу между разными реер.

В системах такого типа не имеет большого значения удобство доступа к реер. Самым важным является сам реер — поскольку его информация является уникальной. Задача оптимизации возникает только при выборе оптимального маршрута обмена сообщениями между этими двумя peers.

В качестве примера систем второго уровня можно рассматривать системы разделения файлов (Gnutella, FreeNet, Napster, Free Haven, ...). Для пользователей этого типа систем не важно кто является хозяином этой системы, а важно, какими ресурсами обладает на данный момент тот или иной реер.

Второй тип P2P систем имеет более сложную систему поиска и навигации. Для этих систем не важно где взять искомый ресурс, важен сам ресурс и оптимальный вариант для получения ресурса, с точки зрения скорости закачки ресурса. В зависимости от размера (в байтах) ресурса, система может усложнять алгоритмы расчетов на поиск оптимального решения. Если ресурс имеет большой объем, то имеет смысл потратить больше времени на поиск оптимального реер, содержащего данный ресурс.

Существуют самые разнообразные алгоритмы для поиска оптимального решения, специфика каждого связана со спецификой конкретной сети. Конечный пользователь — разработчик прикладной P2P системы сможет выбрать оптимальное решение для своей системы.

Особым интересом для команды RockyRoad обладает сервис обнаружения реер, использующий очень быстрый алгоритм, называемый “beaconing” (от англ. beacon —

маяк). Он основан на триангуляции. Подробности алгоритма выходят за рамки этой статьи, для справки смотрите ссылку в конце текста.

4.3. Сервисы безопасности и аутентификации

Этот раздел описывает подход RockyRoad к проблеме безопасного и аутентифицированного обмена сообщениями в распределенной системе со многими участниками (peers).

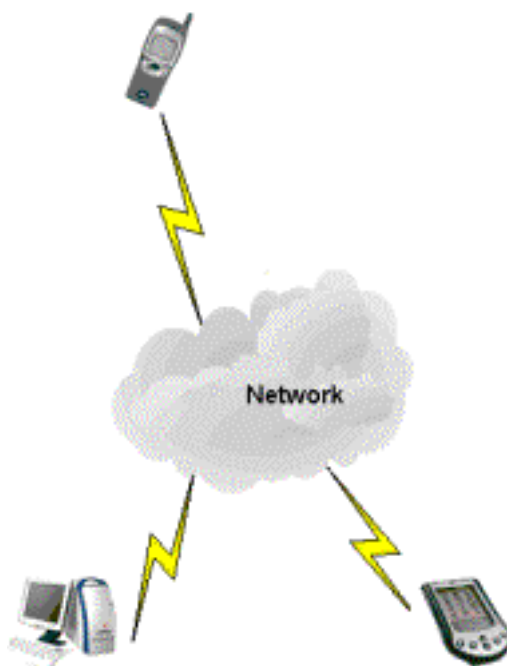


Рисунок 2. Проблемы безопасности.

Рассмотрим процесс взаимодействия между peers в сети. Данная сеть считается «небезопасной». Атакующая сторона (на рисунке 2 скрытая в облаке) обладает следующими возможностями:

- 1) перехватывать любые сообщения (попытки соединения) между данным peer(A) и любым peer B, а также ответные сообщения B к A.
- 2) Просматривать содержимое перехваченного сообщения и совершать одно из следующих действий:

- a. Переслать неизменное сообщение оригинальному адресату;
- b. Удалить сообщение;
- c. Переслать измененное любым образом сообщение оригинальному адресату.

3) Разговаривать с реер В, притворяясь реер А, и наоборот (у атакующей стороны есть возможность не только модифицировать существующие сообщения, но и создавать новые).

Такое описание/определение атакующей стороны может показаться несколько преувеличенным по сравнению с реальным положением вещей, но на самом деле это ничто большее, чем обобщенная картина пользователя, общающегося с миром через прозрачный прокси-сервер своего провайдера. Идентичная картина возникает в процессе доставки электронной почты, когда атакующий захватывает транзитный сервер. IP-сети, в которых перехват и/или изменение пакетов относительно легки, — это хороший пример среды, в которой у атакующей стороны при определенных обстоятельствах могут быть все приведенные выше возможности.

Вышеописанная проблема сводится к проблеме безопасной коммуникации групп. Решение подробно описано в [2], и здесь обсуждаться не будет. RockyRoad решает проблему безопасных групп с помощью немного измененной версии ключевых графов, описанных в [2].

4.4. Сервисы доступа к данным

Обычная проблема для разработчиков мобильных приложений — обеспечить пользователям доступ к большим базам данных, учитывая ограниченные ресурсы таких устройств, как мобильные телефоны. Традиционный клиент-серверный подход в этом случае по ряду причин далек от оптимального. Рассмотрим, например, PDA под управлением PalmOS.

Типичный PalmOS PDA имеет от 4 до 8 МВ энергонезависимой памяти для программ и данных, модемное подключение к Интернет со скоростью от 9600 до 14400 бит/с и не имеет внешних устройств хранения данных. Большинство клиент-серверных приложений требуют быстрой и надежной связи для потоковой передачи данных. Кроме того, хранение данных на клиентском устройстве затруднено из-за

ограниченного объема памяти.

Одно из возможных решений — интеллектуальное кэширование данных на стороне клиента. Этот сервис использует асинхронный протокол для обмена сообщениями между сервером и клиентом. «Интеллектуальность» относится к способности минимизировать количество сообщений, посылаемых для синхронизации данных. Создается иллюзия, что клиентское приложение работает с полной базой данных независимо от состояния соединения. На самом же деле клиент хранит только наиболее часто используемые данные, а неиспользованные порции постепенно стираются (с сохранением целостности на уровне ссылок).

Изменение данных на сервере вызывает посылку всем клиентам уведомления. Клиенты, находящиеся в этот момент в оффлайне, получают уведомление при подключении к сети. Термин «сервер» употребляется здесь не в традиционном смысле (поскольку все устройства, обменивающиеся информацией по сети, считаются равноправными), а относится к участнику сети (peer), который может хранить данные, не используемые мобильными peers и не кэшируемые в них. Сервер может быть частью уже существующей системы (например, SQL сервер) и при этом принимать равноправное участие в обмене данными.

Gateway

Gateway обеспечивают взаимодействие между peers, находящимися в разных сетях (IP, GSM и т.д.) В RockyRoad есть некоторые встроенные шлюзы. Они также необходимы для общения с peers, не поддерживающими протокол RockyRoad, но использующими протокол другого производителя (Jabber, Gnutella, Freenet и т.д.)

<http://www.jrra.org/bin/view/Main/Services>

5. Flexible networking

RockyRoad поддерживает гибкую настройку сетевого взаимодействия между peer-ами (flexiblenetworking). В будущем, когда возможность подключиться к сети будет повсеместной, мобильный пользователь, перемещающийся с места на место, будет получать доступ к различным видам сетевых и транспортных протоколов и прочим сервисам.

Имея возможность выбора сетевых и транспортных механизмов и сервисов более высокого уровня, мобильный пользователь должен иметь возможность настроить свои предпочтения желаемым образом. Например, он может предпочесть транспортный механизм, предоставляющий сервис с установлением соединения. Все такие предпочтения отражаются в политике пользователя (policy). RockyRoad содержит общие интерфейсы и несколько базовых реализаций. Для разработчиков прикладных систем RockyRoad предоставляет механизм указания политик, используя который, разработчик может получить предпочтения пользователя и передать их RockyRoad. Протокол использует механизм динамического связывания, чтобы предоставить пользователю наиболее подходящий сетевой, транспортный или сервисный механизм.

5.1. Динамическое привязывание

Мы считаем, что RockyRoad — первый API, полностью поддерживающий динамическое привязывание к сервисам на основе политики пользователя. Динамическое привязывание — простая концепция, для использования которой разработчик должен использовать RockyRoad с опцией блока политики. Блок политики позволяет разработчику предоставить конечному пользователю механизм указания политики и передать эти данные протоколу RockyRoad. Протокол динамически транслирует политику, чтобы найти наиболее подходящий сетевой транспорт или сервисный механизм. Термин «динамическое привязывание» позаимствован из языков программирования типа C++, где позднее связывание — одна из особенностей языка. Термин «динамическое» требует пояснения. Почему так назван простой механизм трансляции политики? Ответ заключается в том факте, что, когда RockyRoad используется с опцией `flexiblenetworking`, в нем запускается активный поток, который ищет транспортные, сетевые или сервисные механизмы, соответствующие предпочтениям пользователя. Другими словами, RockyRoad может динамически находить подходящие механизмы в зависимости от среды, в которой находится пользователь. Конечно, успешность динамического привязывания зависит от способности четко описывать политику и наличия простого механизма трансляции политики в желаемый сетевой механизм. Это может быть удобно реализовано с использованием XML.

6. Похожие проекты

Сейчас RockyRoad — открытая структура для создания P2P приложений, чем-то напоминающая JXTA фирмы Sun [5]. По сравнению с последним, RockyRoad был с самого начала нацелен на мобильные устройства. В последнее время Sun начала свой собственный открытый проект JXME, направленный на рынок мобильных устройств. Кроме того, RockyRoad — пакетно-ориентированный протокол, использующий в качестве основного транспорта UDP (в нашей базовой реализации) из-за нестабильных соединений и ограниченных ресурсов, обычных в мобильной среде. Этим он отличается от реализации фирмой Sun протокола JXTA, основанной на «поточно-ориентированном» TCP. Тем не менее, RockyRoad не стремится быть конкурентом JXTA, но скорее служит дополнительным API для JXTA. С другой стороны, JXTA можно рассматривать как один из дополнительных сервисов, используемых RockyRoad.

JINI [6] — еще одна технология, поддерживающая flexible networking. JINI позволяет гибкое обнаружение сервисов. Но RockyRoad идет дальше, вводя механизм политик, используя которые, разработчик может указывать индивидуальные характеристики предпочтительных сетевых, транспортных и сервисных механизмов. Кроме того, RockyRoad действительно поддерживает динамическое привязывание для мобильных пользователей в том смысле, что на основании предпочтений пользователя выбирается наиболее подходящий сетевой механизм, и это происходит при каждом изменении контекста, в котором находится пользователь. Это происходит потому, что RockyRoad активно исследует нижележащие уровни протоколов, чтобы найти наилучший вариант.

Интерес к беспроводным P2P системам растет. Больше информации вы можете найти по адресу

<http://www.wireless-p2p.com/index.html>

7. Ссылки

1. JRRR. <http://www.jrra.org>
2. Christopher Kommareddy, Narendar Shankar, Samrat Bhattacharjee. Finding close friends on the Internet — To appear in proceedings of IEEE International conference on Network Protocols-ICNP 2001"
3. C. Wong, M. Gouda, and S. Lam. Secure group communication using key graphs

- . In Proceedings of the ACM SIGCOMM'98, pages 68-79, September 1998.
- 4. N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In Proc. of the Sixth Annual ACM International Conference on Mobile Computing and Networking (MOBICOM), August 2000.
- 5. Websigns: Hyperlinking Physical Locations to the Web.
www.computer.org/computer/homepage/august/cover2
- 6. JXTA. <http://www.sun.com/software/jxta/> , <http://www.jxta.org>
- 7. JINI. <http://www.sun.com/jini> , <http://www.jini.org>