

Протоколы сетевых радиотрансляций Iccast/Shoutcast

Соколов Р.А., МФТИ
E-mail: romanso@rt.mipt.ru

Введение - Несмотря на то, что был создан ряд протоколов для трансляции медиаданных по сети (UDP/RTP, RTSP, MMS, SDP и др.), в настоящий момент для радио-трансляций по сети преимущественно используются протоколы TCP/HTTP с программными протоколами Shoutcast/Iccast (официально не задокументированы). HTTP, поддерживаемый протоколами IP и TCP (таким образом, HTTP сам не обрабатывает передачу пакетов данных) – это протокол, который позволяет браузерам загружать HTML-страницы. Этот протокол позволяет не только связывать документы между клиентом и сервером, но также он допускает потоковую трансляцию аудиоданных посредством передачи пакетов и исправляет ошибки (в случае, если пакет не будет доставлен с первого раза) посредством повторной передачи. Далее подробнее остановимся на протоколе Iccast/Shoutcast и на схеме создания соединения и передачи mp3-данных при прослушивании трансляции (в частности, нас интересует взаимодействие сервер-клиент). Особенностью сетевых трансляций является то, что для прослушивания аудиопотока не нужно скачивать аудиофайлы целиком, а сама трансляция не сохраняется на диске.

I. Протоколы Iccast/Shoutcast.

Система трансляции (сервер) Shoutcast (и соответствующий протокол) разрабатывается компанией Nullsoft и является полупромышленной, тогда как Iccast разрабатывается под лицензией GPL (является проектом с открытым исходным кодом). Обе системы в целом совместимы.

Оба протокола в большой степени основаны на HTTP/1.0. Основное различие – это группа новых заголовков: icy-заголовков в Shoutcast и x-audiocast-заголовков в Iccast.

URL типичного Iccast- или Shoutcast-потока имеет вид:

`http://Server[/path] [/file]:port`

или

`http://Server/path/file.pls`

Примечание: номер порта, как правило, лежит в диапазоне 8000-8999, в любом случае, он назначается сервером.

Многие Shoutcast- и Iccast-серверы не имеют собственных доменных имен. Таким образом, URL обычно имеет вид:

`http://nnn.nnn.nnn.nnn:XXXX`

где nnn.nnn.nnn.nnn - это IP-адрес сервера, а XXXX – номер порта.

A. Основные инструменты при транслировании:

- Источник (как правило, dsp-модуль в плеере)
- Сервер (поставляет mp3-поток источника клиенту)
- Клиент (используется для прослушивания аудиопотока, идущего с сервера)

В. Источник-сервер (для нашей задачи взаимодействие “источник-сервер” можно пропустить)

Чтобы сервер мог связываться с клиентом, ему нужен источник. Когда соединение с источником установлено, сервер будет передавать данные клиентам, когда они будут подключаться.

Диалог происходит так (рассмотрим на примере Shoutcast, далее при подробном рассмотрении установления соединения будут описаны особенности и Shoutcast, и Icecast)

1. Источник создает соединение с портом сервера (служебным)
2. Затем источник посылает пароль: password\r\n
3. Если пароль правильный, сервер посылает в ответ

```
OK2\r\n
icy-caps:11\r\n\r\n
```

что информирует источник о том, что сервер авторизовал dsp-модуль в качестве источника и готов принимать данные. Если пароль неправильный, сервер отправит в ответ неправильный пароль password\r\n.

4. Если источник получает в ответ OK2, он начинает посылать информацию о потоке серверу. Как правило, в форме:

```
icy-name:Unnamed
Server\r\n
icy-genre:Unknown
Genre\r\n
icy-pub:1\r\n
icy-br:56\r\n
icy-url:http://www.shoutcast.com\r\n
icy-irc:%23shoutcast\r\n
icy-icq:0\r\n
icy-aim:N%2FA\r\n
\r\n
```

Здесь для передачи информации и потоке используются заголовки:

icy-name – название станции
icy-genre – музыкальный жанр станции
icy-pub - указывает допускает ли сервер публикацию себя в публичной директории (1 – да, 0 -нет)
icy-br – битрейт потока
icy-url - homepage потока
icy-irc, icy-icq, icy-aim – контактная информация для публикации в публичной директории

5. Затем источник начинает отправлять mp3-поток.

С. Передача названия (песни) от источника серверу

Сервер получает название песни и URL страницы когда источник делает вызов URL.

```
http://www.host.com:portnumber/admin.cgi?pass=Server%20Password&mode=upinfo&song=Song%20Goes%20here&url=http://someurl.com
```

Когда источник делает этот вызов, название песни в клиенте, который поддерживает Shoutcast-стиль передачи названий, изменяется. Это взаимодействие всегда совершается через публичный порт (по умолчанию 8000), ни в коем случае не через служебный, так как он используется строго для передачи потока серверу.

D. Клиент-сервер

Взаимодействие клиент-сервер происходит способом, аналогичным тому, как взаимодействуют браузер и веб-сервер - по протоколу HTTP. Однако Shoutcast и Icecast имеют дополнительные заголовки.

1. Клиент подключается к серверу и, в добавок к обычному HTTP-заголовку, отправляет ему дополнительное поле:

```
icy-metadata:val\r\n
```

Этот тэг указывает на то, что если val=1, то клиент может обрабатывать названия песен (метаданные), передаваемые в потоке, и, таким образом, сервер будет посылать дополнительную информацию о названии. Если val=0, то метаданные передаваться не будут.

2. Затем сервер отправляет ответ:

```
ICY 200 OK\r\n (означает, что сервер принял запрос)
```

```
icy-notice1:<BR>This stream requires <a  
href="http://www.winamp.com/">Winamp</a><BR> (избыточное замечание)
```

```
icy-notice2:SHOUTcast Distributed Network Audio Server/posix  
v1.x.x<BR> (сообщает клиенту, какой это сервер и его версию)
```

```
icy-name:Unnamed Server\r\n (имя сервера)
```

```
icy-genre:Unknown Genre\r\n (жанр сервера)
```

```
icy-url:http://www.shoutcast.com\r\n (homepage сервера)
```

```
icy-pub:1\r\n (публичный или непубличный сервер)
```

```
icy-br:56\r\n (битрейт сервера)
```

```
icy-metaint:8192\r\n (см. далее)
```

```
\r\n (конец заголовка)
```

3. С этого момента сервер начинает посылать аудио-данные.

E. Передача метаданных в протоколе Shoutcast

Ранее мы рассмотрели, как сервер получает название песни от источника, теперь рассмотрим, как его получает клиент.

Когда клиент сообщает о том, что он может обрабатывать названия, Shoutcast-сервер добавляет следующий тэг заголовка:

```
icy-metaint:8192\r\n
```

который сообщает клиенту, сколько байт данных из потока нужно прочитать, прежде чем начнутся метаданные (в которых и содержится название). Они всегда начинаются в начале потока (а не в заголовке).



После этого клиент считывает один байт, который сообщает ему размер метаданных, деленный на 16, то есть если этот байт равнялся 4, то длина тэга метаданных - 64 байта. Если метаданные не равны в точности 64 байтам (например), Shoutcast помещает пробелы или "\0" в неиспользуемом пространстве.

Итак, процедура выделения метаданных (названия песни) из потока выглядит так:

1. Запрос метаданных:

Это просто добавление нового поля в HTTP-запрос:

```
Icy-MetaData:1
```

То есть, весь запрос будет выглядеть так:

```
GET path HTTP/1.0  
Icy-MetaData:1
```

Если будут запрашиваться метаданные, нужно уметь извлекать их из потока, иначе звуковой поток будет прерываться каждую секунду (хотя это хороший способ узнать, получаем ли мы метаданные вообще :).

2. Получение интервала метаданных:

Один из заголовков, которые вернутся на ваш запрос, будет сообщать о том, как часто метаданные будут посылаться в потоке. В частности, сколько байт MP3-данных будет между блоками метаданных. Этот заголовок выглядит так:

```
icy-metaint: number
```

Возможно, нужно будет хранить это число.

3. Получение данных:

Считываем поток данных и считаем байты. Когда число байт стало равно `number`, мы дошли до блока метаданных. Первая часть блока – это указатель длины. Как уже говорилось, он равен (*длина метаданных / 16*). Умножаем его на 16, чтобы получить длину метаданных (максимальная длина метаданных = 4080). Теперь считываем это количество байт – и мы имеем строку, содержащую метаданные. Обнуляем счетчик данных и повторяем все заново.

Следует заметить, что чаще всего длина метаданных равна 0, то есть их просто нет в потоке. Метаданные, как правило, посылаются в двух местах: сразу после соединения и когда сменяются песни.

4. Разбор метаданных:

Часть строки метаданных должна выглядеть так:

```
StreamTitle='title of the song';
```

что нам и нужно было.

II. Установка соединения с сервером, передача mp3-данных (пример реализации)

В этом разделе более подробно рассмотрим процесс соединения с сервером и передачу пакетов mp3-данных в Shoutcast/Icecast-трансляциях на примере программной реализации клиента, совместимого с обоими протоколами.

Заметим, что перед передачей данных mp3-плееру они буферизуются для того, чтобы избежать искажения аудиопотока. Рассматриваемый клиент не обрабатывает названия песен.

Одно наиболее существенное различие между протоколами Icecast и Shoutcast состоит в том, что Icecast-клиент использует дополнительный UDP-канал для обновления метаданных, тогда как по протоколу Shoutcast метаданные (как было рассмотрено ранее) вставляются в общий поток между mp3-пакетами. Для метаданных используются HTTP-заголовки icy (в Shoutcast) и x-audiocast (в Icecast).

Итак, процедура со стороны клиента выглядит следующим образом:

1. Получаем и разбираем адрес трансляции на имя хоста и порт.
2. В случае работы с Shoutcast-сервером создаем TCP-сокеты и соединяем его с сервером. В случае icecast-сервера создаем два сокета: один (TCP) для получения mp3-потока, другой (UDP) для передачи пользовательских датаграмм и метаданных.
3. Отправляем в сокет сообщение вида (в случае Icecast обмен сообщениями идет через UDP-сокеты):

```
GET / HTTP/1.0
```

в случае Shoutcast-сервера или

```
GET / HTTP/1.0
Host: ****.****.****.***
x-audiocast-udpport: 6000
Icy-MetaData: 0
Accept: */*
```

в случае Icecast-сервера.

4. Получаем из сокета сообщение вида:

```
HTTP/1.0 200 OK
Server: Icecast/VERSION
Content-Type: audio/mpeg
x-audiocast-name: Great Songs
x-audiocast-genre: Jazz
x-audiocast-url: http://icecast.serv.dom/
x-audiocast-streamid:
x-audiocast-public: 0
```

```
x-audiocast-bitrate: 24
x-audiocast-description: served by Icecast
```

для Icecast-сервера или, в случае, Shoutcast-сервера

```
ICY 200 OK
icy-noticel:<BR>This stream requires <a
href="http://www.winamp.com/">Winamp</a><BR>
icy-notice2:SHOUTcast Distributed Network Audio Server/posix
v1.0b<BR>
icy-name:whatever
icy-genre:whatever
icy-url:whatever
icy-pub:1
icy-br:128
```

5. И далее читаем из сокета в буфер данные: в случае icecast будет только mp3-поток - <data>; в случае shoutcast mp3-поток может прерываться метаданными - <data><songtitle><data>.

Мр3-данные передаются в виде так называемых фреймов (frame, или кадр), в которых хранятся аудиоданные внутри mp3-файла.

Источники

Для отчета были использованы исходные коды пакетов libshout (поставщик mp3-данных серверу - источник), серверов ruby-shout, LifeRadio, клиентов icecast-client, mpg321, freeamp.