



Поиск (теги):

[НОВОСТИ \(+\)](#)[КОНТЕНТ](#)[WIKI](#)[МАН'Ы](#)[ФОРУМ](#)**ПРОЕКТ IP-AS.RU
100% РЕЗУЛЬТАТ**

Протоколы сетевых радиотрансляций Icast/Shoutcast (radio shoutcast sound multimedia mp3)

<< [Предыдущая](#) [ИНДЕКС](#) [Поиск в статьях](#) [src](#) [Установить закладку](#) [Перейти на закладку](#) [Следующая](#) >>**Ключевые слова:** radio, shoutcast, sound, multimedia, mp3, ([найти похожие документы](#))

From: Соколов Р.А. <romanso at rt tчk mпт tчk ru>
Newsgroups: email
Date: Mon, 14 Mar 2004 14:31:37 +0000 (UTC)
Subject: Протоколы сетевых радиотрансляций Icast/Shoutcast

Протоколы сетевых радиотрансляций Icast/Shoutcast

PDF версия: https://www.opennet.ru/soft/Network_radio_protocols.pdf

Соколов Р.А.
E-mail: <romanso at rt tчk mпт tчk ru>

Аннотация - Несмотря на то, что был создан ряд протоколов для трансляции медиаданных по сети (UDP/RTP, RTSP, MMS, SDP и др.), в настоящий момент для радиотрансляций по сети преимущественно используются протоколы TCP/HTTP с программными протоколами Shoutcast/Icast (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 Icast/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 Icast/Shoutcast.

Система трансляции (сервер) Shoutcast (и соответствующий протокол) разрабатывается компанией Nullsoft и является полукоммерческой, тогда как Icast 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 Icast.

URL типичного Icast- или Shoutcast-потока имеет вид:

[http://Server[/path\] \[/file\]:port](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 Icecast-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.

А. Основные инструменты при транслировании:

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

В. Источник-сервер

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=updinfo&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-noticel:<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 (hompage 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 прочитывать, 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:

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:

Получаем и разбиваем адрес трансляции на имя хоста и порт.

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 данных и метаданных.

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.

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
```

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.

<< [Предыдущая](#) [ИНДЕКС](#) [Поиск в статьях](#) [src](#) [Установить закладку](#) [Перейти на закладку](#) [Следующая](#) >>

Обсуждение

[[Линейный режим](#) | [Показать все](#) | [RSS](#)]

- 1.1, [bsd2002](#), 18:43, 16/03/2004 [[ответить](#)] [[смотреть все](#)]

[+/-](#)

Прикольно, а кто расскажет как использовать источником TV/FM тюнер?

- 2.2, [Maxim Chirkov](#), 10:03, 17/03/2004 [[^](#)] [[ответить](#)] [[смотреть все](#)]

[+/-](#)

>Прикольно, а кто расскажет как использовать источником TV/FM тюнер?



mixer rec 100

```
./liveice -F /usr/local/icecast/conf/liveice.cfg -@ 2 -m
/usr/local/icecast/bin/icecast
```

Настройка icecast трудностей вызвать не должна.

Пример liveice.cfg для проигрывания потока с линейного входа звуковой карты:

```
SERVER 192.168.1.1
PORT 8000
PASSWORD пароль
USE_LAME3 /usr/local/bin/lame
ENCODER_ARGS -r -x -S -b 24 -B 24 -s 48 --resample 22 -m m -f -V 9
SOUNDCARD
HALF_DUPLEX
```

```
SAMPLE_RATE 48000
MONO
ENCODING_QUALITY 70
X_AUDIOCAST_LOGIN
MOUNTPPOINT live
NAME Tyumen Live Stream
GENRE baracuda
PUBLIC 1
URL http://radio.test.ru
DESCRIPTION XXX FM Online
NO_MIXER
PLAYLIST playlist
DECODER_COMMAND mpg123
MIX_CONTROL_MANUAL
CONTROL_FILE ./mix_command
TRACK_LOGFILE track.log
VERBOSE 0
```

- 1.3, [lvigala](#), 00:33, 13/06/2005 [[ответить](#)] [[смотреть все](#)]

+/-

Можно ли передавать WAV файлы ???

- 2.13, [Dj Forsage](#), 15:05, 11/05/2007 [^] [[ответить](#)] [[смотреть все](#)]

+/-

Можно! в DSP плагине к винампу ты указываешь encoder (MP3, ACC, None) вот какраз "None" и будет передавать на сервер поток без кодирования в WAV формате.

- 1.4, [Vadim](#), 16:46, 26/10/2005 [[ответить](#)] [[смотреть все](#)]

+/-

Подскажите, как организовать вещание "Наше радио" через инет в локальную сеть? Чтобы поток приходил на сервак, а уже от него - к клиентам сети.
Спасибо.

- 1.5, [Vanilla](#), 22:18, 26/11/2005 [[ответить](#)] [[смотреть все](#)]

+/-

Господа! Где нарыть скриптец, который бы выдирает из потока(?) название песни, битрейт... для выкладывания всего этого на сайте... Хелп, плиз..

- 2.12, [J4ck](#), 14:44, 28/04/2007 [^] [[ответить](#)] [[смотреть все](#)]

+/-

Тут был код выдающий тайтл
http://www.skillz.ru/docs/docsext/misc/article-Blok_monitoringa_radio_icecast
И о настройке icecast+darkice+fm tuner на фре
http://www.skillz.ru/dev/freebsd/article-Setevoje_radio_ICECAST_FM_tjuner.htm

- 1.6, [SoundGod](#), 09:26, 07/12/2005 [[ответить](#)] [[смотреть все](#)]

+/-

Люди, помогите нам с Vanilla! Я тож ишу возможность отображения названия песни на сайте! Откликнитесь!

- 1.7, [begemotik](#), 00:37, 07/02/2006 [[ответить](#)] [[смотреть все](#)]

+/-

А как быть с теми слушателями, кто заходит в сеть через прокси?
Пример. Корпоративная сеть, У всех на рабочих компах браузер настроен на прокси, а из плееров стоит только медиаплеер (напомню, его сетевые настройки по дефолту - ИЕшные). ведь не заставишь админов всех таких сетей разрешить на серваке нужные порты (с вещанием).
Еще вопрос. Может, кто знает более промышленное решение радиовещания под линухой?
Спасибо.

- 1.9, [Artomen](#), 19:46, 06/04/2006 [[ответить](#)] [[смотреть все](#)]

+/-

А для установки appserv ставить нужно?
А то что, не вгоняю!

- 1.10, [345](#), 05:34, 26/07/2006 [[ответить](#)] [[смотреть все](#)]

+/-

одни вопросы без ответов)

- 2.11, [Дмитрий](#), 01:52, 12/03/2007 [^] [[ответить](#)] [[смотреть все](#)]

+/-

Какие вопросы - такие и ответы.

- 1.14, [4igavskiy](#), 07:06, 25/06/2007 [[ответить](#)] [[смотреть все](#)]

+/-

классная статья, но не описано, как меняется заголовок песни в icecast...



- 1.15, [crazy](#), 13:57, 18/09/2007 [[ответить](#)] [[смотреть все](#)]

+/-

Скажите, нигде не могу найти инфу, если мне нужно чтоб могли подключиться одновременно допустим 100 клиентов и поток вещания 32 кбпс то в итоге полоса в интернет нужна 3200 кбпс? или только 32кбпс?

- 1.16, [j4ck](#), 11:21, 22/11/2007 [[ответить](#)] [[смотреть все](#)]

+/-

С мультикастом - 32кбп, иначе по полной

- **2.18**, [rrv](#), 21:37, 28/10/2008 [[^](#)] [[ответить](#)] [[смотреть все](#)]

[+/-](#)

Вот тут подробное описание как настроить вещание в локалке радиостанции http://rrv.nsk.ru/wiki/index.php/IceCast_%D0%BD%D0%B0

- **1.19**, [Серг](#), 00:30, 16/02/2013 [[ответить](#)] [[смотреть все](#)]

[+/-](#)

Хм если честно вообще не чего не понял! И даже не понял для чего эта статья написана! Я искал в яндексе как мне вывести мое радио в интернет то есть вывести общий прямой урл к моему потоку. Чтоб другие люди могли слушать.

Ваш комментарий

Имя:

E-Mail:

Заголовок:

Текст:



[Розыгрыш призов для подписчиков журнала Linux Format](#)

Все, кто приобрел подписку на 2016 год, могут участвовать и выиграть один из призов:

- Набор для сборки 3D-принтера;
- Intel Edison;
- Cubieboard;
- Raspberry Pi и др.

Оформить подписку, а также посмотреть полный перечень призов можно на [нашем сайте](#).

[Закладки на сайте](#)
[Проследить за страницей](#)

Created 1996-2016 by [Maxim Chirkov](#)
[Добавить](#), [Реклама](#), [Вебмастеру](#), [ГИД](#)

